Proceedings for ESSLLI 2015 Workshop 'Empirical Advances in Categorial Grammar' (CG 2015)

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

July 7, 2015

Preface

This volume contains the papers selected for presentation at the ESSLLI 2015 workshop 'Empirical Advances in Categorial Grammar' (CG 2015), to be held on August 10–14, 2015 in Barcelona.

There were 18 submissions to the workshop. Each submission was reviewed by at least two reviewers, either the program commttee members or outside reviewers. 10 papers were selected out of the 18 submissions.

We would like to thank the following individuals for reviewing the submissions: Nicholas Asher, Chris Barker, Daisuke Bekki, Frederick Hoyt, Pauline Jacobson, Mark Mcconville, Koji Mineshima, Michael Moortgat, Richard Moot, Glyn Morrill, Reinhard Muskens, Rick Nouwen, Richard Oehrle, Carl Pollard, Mark Steedman, Wataru Uegaki, Oriol Valentín, Michael White, Neal Whitman, Yoad Winter, and Shûichi Yatabe.

We would like to thank the steering committee and the local organizers of ESSLLI 2015 for making this event happen. We would also like to express our thanks to our sponsors, the Institute for Comparative Research at the University of Tsukuba and Japan Society for the Promotion of Science (KAKENHI; Grant number 15K16732) for their financial support.

July 7, 2015
Columbus

Yusuke Kubota
Robert Levine

Table of Contents

Two types of Japanese scrambling in combinatory categorial grammar 1
Daisuke Bekki

A Categorial Analysis of Chinese Alternative Questions 13
Manjuan Duan

Syntactic Features for Regular Constraints and an Approximation of Directional Slashes in Abstract Categorial Grammars 34
Makoto Kanazawa

Compositional semantics of *same*, *different*, *total* 71
Oleg Kiselyov

A Unidimensional Syntax-Semantics Interface for Supplements 82
Scott Martin

Combining grammar logics 107
Michael Moortgat

Comparing and evaluating extended Lambek calculi 108
Richard Moot

Computational Coverage of TLG: Displacement 132
Glyn Morrill and Oriol Valentín

Coordination in Linear Categorial Grammar with Phenogrammatical Subtyping .. 162
Carl Pollard and Chris Worth

Right-Node Wrapping: A combinatory account 183
Alexander Warstadt

Two types of Japanese scrambling in combinatory categorial grammar

Daisuke Bekki *

Ochanomizu University †
National Institute of Informatics ‡
CREST, Japan Science and Technology Agency §

Abstract

This paper presents an analysis of Japanese scrambling in the framework of combinatory categorial grammar (CCG) and dependent type semantics (DTS). Along the line of non-uniform analysis of Japanese scrambling by Ueyama (1998, 2003) in a minimalist framework, I claim that two combinatory rules, CB and Bx, are responsible for generating two types of scrambling in Japanese: surface and deep scrambling constructions. We claim that i) there is a language specific rule CB, that we call the permuted functional composition rule, which serves as the source of Ueyama's (unbounded) surface scrambling construction that accompanies a reconstruction effect; ii) the rule Bx, the crossed functional composition rule that Steedman (2000) adopts for the analysis of Japanese scrambling, derives Ueyama's (clause-bounded) deep scrambling constructions, in which the dislocated element is interpreted in its landing site. While this analysis successfully simulates Ueyama's analysis in a purely type-theoretical framework, it also yields different predictions in the long-distance scrambling across complex NP boundaries, which is marginally allowed when certain conditions are met. A construction that we call a *subject chunk* plays a crucial role here, in which a matrix subject and an embedded subject merge as a single constituent, mediated by the empty determiner and the general function composition rules. It further predicts that the coordination of subject chunks is allowed, which highlights an empirical difference between the predictions of our analysis, other categorial analyses and minimalist analyses.

Keywords: Japanese Scrambling; Combinatory Categorial Grammar

1 Introduction: Two types of scrambling constructions and operations in Japanese

It has been observed in literature (e.g., Saito (1992), Ueyama (1998, 2003)) that Japanese language has two forms of *scrambling* constructions: 1) *clause-internal scrambling*, which

*My sincere thanks to Ayumi Ueyama, Koji Mineshima, Ai Kawazoe and Yusuke Kubota for many helpful comments. I also thank the anonymous reviewers for their insightful comments. This research is partially supported by JST, CREST.

†2-1-1 Ohtsuka, Bunkyo-ku, Tokyo 112-8610, Japan.
‡2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan.
§4-1-8 Honcho, Kawaguchi, Saitama, 332-0012, Japan.

is a dislocation of an argument NP within a clause, and 2) *long-distance scrambling*, which is a dislocation of an argument NP over clause boundaries.

Scrambling has been a central topic in Japanese syntax, partly because it involves some puzzles, including the following two.

1) In clause-internal scrambling, the dislocated NP may be either interpreted in its original position (known as the *reconstruction effect*) or in its landing site, as discussed in Hoji (1985, chapter 3) and Ueyama (1998, 2003). For long-distance scrambling, in contrast, the reconstruction effect is obligatory. How can these different behaviors be explained?

2) Harada (1977) has pointed out that long-distance scrambling obeys a particular kind of subjacency condition, but syntactic studies since then have revealed that the judgments are not as robust as expected: for example, it may violate complex-NP constraints when certain conditions are met. Since this is the case, what kind of 'movement' can scrambling be explained as?

According to Ueyama (1998, 2003)'s non-uniform analysis[1], there are two distinct, underlying scrambling constructions — *surface* and *deep* scrambling constructions — that realize the clause-internal and long-distance scrambling constructions mentioned above. Surface scrambling constructions are not clause bound, and in them the dislocated NPs are interpreted in their original positions (hence reconstruction effects). However, deep scrambling constructions are clause bound, and the dislocated NPs are interpreted in their landing sites. Moreover, the long-distance scrambling construction is derived only as surface scrambling while the clause-internal distance scrambling construction can be derived either as surface scrambling and as deep scrambling.

This paper aims to reformulate the analysis in Ueyama (1998, 2003), placing it into the setting of combinatory categorial grammar (CCG; Steedman (1996, 2000)). This analysis not only expands the empirical boundary of CCG in Japanese but also brings a new perspective on what takes place in these exotic, controversial constructions.

2 Analysis

We adopt the framework of Japanese CCG (Bekki, 2010) for our description of Japanese syntax (parts of speech, verb conjugation, and other syntactic features), and dependent type semantics (DTS; Bekki (2014))[2] for our description of semantics. Within this setting, we make two claims.

Claim 1: There exists, in Universal Grammar, a pair of combinatory rules $<$**CB** and $>$**CB**, which we call *permuted functional composition rules*, defined as follows.[3]

[1] Saito (2003)'s uniform analysis claims that both types of constructions are derived by a single operation. Ueyama (2003) details a counter to that argument.

[2] DTS is a framework of natural language semantics based on dependent type theory (Martin-Löf (1984), Coquand and Huet (1988)). As a proof-theoretic semantic of natural language, in contrast to model-theoretic semantics, DTS provides a unified analysis of entailment, anaphora resolution and presupposition binding, inheriting the *presupposition as anaphora* paradigm of van der Sandt (1992) and the *anaphoar resolution as proof construction* paradigm of Krahmer and Piwek (1999) by means of underspecified terms and context-passing mechanism (see Bekki (2014) for details). The DTS-style representation $(x{:}A) \rightarrow B$ roughly corresponds to $\forall x(A \rightarrow B)$, and the representation $\left[\begin{array}{l} x{:}A \\ B \end{array}\right]$ to $\exists x(A \wedge B)$.

[3] Our analysis of Japanese surface scrambling through the rule $>$**CB** shares the same spirit with the work of Kang (1987, pp.126-135) on Korean. Kang's proposal is to *relax* the universal, the Principle of Directional Consistency (Steedman, 1990, p.225), namely, "all syntactic combinatory rules must be

(1) $\dfrac{Y/Z : g \quad X/Y : f}{X/Z : \lambda x.f(g(x))} >\mathbf{CB}$ $\qquad$ $\dfrac{X\backslash Y : f \quad Y\backslash Z : g}{X\backslash Z : \lambda x.f(g(x))} <\mathbf{CB}$

We assume that each language optionally employs one or both of the **CB** rules, with Japanese (and possibly other languages with scrambling constructions such as Korean and Turkish) employing only $>\mathbf{CB}$ as a rule for deriving surface scrambling.

The symbol **CB** indicates its semantic definition in combinatory logic: $\mathbf{CB}gfx = \mathbf{B}gfx = f(g(x))$; namely, it is a variation of the functional composition rule **B** with its arguments scrambled by the **C** combinator (recall that the reduction rule for the combinator **C** is $\mathbf{C}fxy = fyx$). As a result, the rule $>\mathbf{CB}$ (and $<\mathbf{CB}$) swaps the applicable order of two adjacent functions g and f so that the function g on the left (right) applies first to the rightward (leftward) argument, to which the function f on the right (left) applies latter. Linguistically, g corresponds a dislocated constituent that is yet interpreted in its original position, thus yielding a reconstruction effect.

Claim 2: Among *functional crossed composition rules* $>\mathbf{B}_\times$ and $<\mathbf{B}_\times$, as shown below, which are harmonic rules often used for extraction from non-right-peripheral cases in English, Japanese has (at least) the $>\mathbf{B}_\times$ rule as the source of deep scrambling constructions, along the lines of Steedman (2000).

(2) $\dfrac{X/Y : f \quad Y\backslash Z : g}{X\backslash Z : \lambda x.f(g(x))} >\mathbf{B}_\times$ $\qquad$ $\dfrac{Y/Z : g \quad X\backslash Y : f}{X/Z : \lambda x.f(g(x))} <\mathbf{B}_\times$

3 Clause-internal scrambling

Let us illustrate how the $>\mathbf{CB}$ and the $>\mathbf{B}_\times$ rules derive, respectively, surface and deep scrambling constructions in the clause-internal scrambling constructions. The sentence schema (3) is the simplest case, where the object is dislocated to the sentence-initial position.

(3) OBJ SBJ V

The structure of (3) is ambiguous, following Ueyama (1998, 2003), with both surface and deep scrambling constructions being possible. In our analysis, the former is derived by means of the $>\mathbf{CB}$ rule, while the latter is derived by means of the $>\mathbf{B}_\times$ rule, as shown in (4) and (5).

---

consistent with the directionality of the principal function" for the 'non-configulational language' like Korean, so that the functional composition rule can be used in both of the following forms:

$$\frac{X/Y : f \qquad Y/Z : g}{X/Z : \lambda x.f(gx)} >\mathbf{B} \qquad \frac{Y/Z : g \qquad X/Y : f}{X/Z : \lambda x.f(gx)} >\mathbf{B}$$

However, unlike Kang, we regard **CB** as different rules from **B**, which does obey the principles of CCG. The difference is that, in Kang's scrambled usage of the $>\mathbf{B}$ rule (the right one above), the primary functor is f (in the sense that the output is $\mathbf{B}fg$) and it comes to the right side, thus violates the Principle of Directional Consistency, whereas in the rule $>\mathbf{CB}$ is g (since g is the first argument in the output semantic formula $\mathbf{CB}gf$) that appears on the left side. Thus we claim that the Principle of Directional Consistency is a universal one, and the parameter to distinguish the languages like Japanese, Korean and Turkish is whether a language allows to use the combinator **C** in their combinatory rules. Along the line of Ozaki and Bekki (2012), we assume that the combinators that participate in this parameter are **C**, **W** and **K**, which correspond to the structural rules in the sense of Hilbert-style substructural logics.

(4)
$$\dfrac{\dfrac{\dfrac{\text{OBJ}}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_o)\ :\ \lambda p.\lambda\vec{x}.\mathbf{Q}_2(y)(py\vec{x})} \quad \dfrac{\text{SBJ}}{S/(S\backslash NP_{ga})\ :\ \lambda p.\mathbf{Q}_1(x)(px)}}{S/(S\backslash NP_{ga}\backslash NP_{ni})\ :\ \lambda p.\mathbf{Q}_1(x)(\mathbf{Q}_2(y)(pyx))}\,{>}\mathbf{CB} \quad \dfrac{\text{V}}{S_{term}\backslash NP_{ga}\backslash NP_o\ :\ \lambda y.\lambda x.\mathbf{V}(x,y)}}{S_{term}\ :\ \mathbf{Q}_1(x)(\mathbf{Q}_2(y)(\mathbf{V}(x,y)))}\,{>}$$

(5)
$$\dfrac{\dfrac{\text{OBJ}}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_o)\ :\ \lambda p.\lambda\vec{x}.\mathbf{Q}_2(y)(py\vec{x})} \quad \dfrac{\dfrac{\text{SBJ}}{S/(S\backslash NP_{ga})\ :\ \lambda p.\mathbf{Q}_1(x)(px)} \quad \dfrac{\text{V}}{S_{term}\backslash NP_{ga}\backslash NP_o\ :\ \lambda y.\lambda x.\mathbf{V}(x,y)}}{S_{term}\backslash NP_o\ :\ \lambda y.\mathbf{Q}_1(x)(\mathbf{V}(x,y))}\,{>}\mathbf{B}_\times}{S_{term}\ :\ \mathbf{Q}_2(y)(\mathbf{Q}_2(x)(\mathbf{V}(x,y)))}\,{>}$$

Notice that, in their semantic representations, the scope relation between $\mathbf{Q}_1(x)$ (which originates in SBJ) and $\mathbf{Q}_2(y)$ (which originates in OBJ) is reversed: in the surface scrambling construction on the left, $\mathbf{Q}_2(y)$ takes a narrower scope under $\mathbf{Q}_1(x)$ (exhibiting a reconstruction effect) while in the deep scrambling construction on the right, $\mathbf{Q}_2(y)$ takes a wider scope over $\mathbf{Q}_1(x)$, which means that the dislocated NP is interpreted at its landing site.

This analysis is supported by the fact that both readings are available for the scheme (3), even when we choose, for the subject, a quantifier that does not allow inverse scope reading. For instance, it is known to be difficult for most Japanese speakers to interpret a floating quantifier such as 'bengosi-o 5-nin' (=five lawyers) in the object position as in (6a) to outscope a quantifier in the subject position (i.e. the inverse scope reading), thus showing an asymmetry for the availability of canonical ($2 > 5$) and inverse scope ($5 > 2$) readings as in (6a). Nevertheless, its scrambled counterpart (6b) allows both readings.

(6) a. Hutatu-no kigyoo-ga bengosi-o 5-nin uttaeta.
two-GEN company-NOM lawyer-ACC five sue-PAST-term
'Two companies sued five lawyers.'
ok $2 > 5$, *$5 > 2$

b. Bengosi-o 5-nin hutatu-no kigyoo-ga uttaeta.
lawyer-ACC five two-GEN company-NOM sue-PAST-term
(lit.) 'Two companies sued five lawyers.'
ok $2 > 5$, $5 > 2$

This distribution of the available readings is correctly predicted in our analysis, in parallel with Ueyama (1998), in the way that the reconstructed interpretation ($2 > 5$) is derived as in (4), while the scrambled-order interpretation ($5 > 2$) is derived as in (5).

4 Long-distance scrambling: complex-NP violation case

Now, let us explain how and why long-distance scrambling in Japanese is allowed to violate the complex-NP constraint. The sentence (7a) exemplifies such a construction, schematized as (7b)[4], which is reported to be more or less acceptable, although marginal

[4]NP-nom$_1$ and [$\cdots$ V]-N-acc$_1$ are arguments for V_1, the matrix verb; NP-nom$_2$ and NP-acc$_2$ are arguments for V_2, the verb of an embedded relative clause.

to many native speakers of Japanese.[5] Note that this sentence is not derivable by the analysis of Japanese scrambling given in Steedman (1996, 2000) and also not by that given in Komagata (1999).

(7) a. ?? [*mori-no-kuma-san*-no-moderu-to-nat-ta-ozyoosan]-ni$_1$ [Taro-wa
the.lady.in.“the other day I met a bear”-DAT Taro-TOP
[kuma-ga t_1 todoke-ta iyaringu]-o hakkensi-ta]
the.bear-NOM t_1 deliver-PAST-attr earring-ACC find-PAST-term
(lit.) ‘Taro found an earring that the bear delivered to *the lady who figured in the song “The Other Day I Met a Bear”*’

b. ? NP-acc$_2$ NP-nom$_1$ NP-nom$_2$ t V$_2$-N-acc$_1$ V$_1$

4.1 Japanese relative clauses

Before getting into the details of the derivation of (7a), let us first examine the derivation of (8a), which is the non-scrambled, unmarked counterpart of (7a), schematized as (8b).

(8) a. Taro-wa [kuma-ga Mary-ni todoke-ta] iyaringu-o
Taro-TOP the.bear-NOM Mary-DAT deliver-PAST-attr earring-ACC
hakkensi-ta
find-PAST-term
(lit.) ‘Taro found an earring that the bear delivered to Mary.’

b. NP-nom$_1$ NP-nom$_2$ NP-acc$_2$ V$_2$-N-acc$_1$ V$_1$

The construction (8a) contains a relative clause *kuma-ga Mary-ni todoke-ta* (=“which the bear delivered to Mary”) which modifies the common noun *iyaringu* (=“earring”). We assume, following Bekki (2010), that the empty category left in a Japanese relative clause is *pro*, namely, an empty pronoun[6,7], which undergoes the standard derivation for a clause headed by a ditransitive verb, as in (9).

(9)

$$\dfrac{\text{kuma-ga}\quad \dfrac{\text{Mary-ni}\quad \boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ni})\quad : \lambda p.p(mary) \qquad \dfrac{\dfrac{pro_1\quad \boldsymbol{T}/(\boldsymbol{T}\backslash NP_o)\quad : \lambda p.\lambda y.\lambda x.\lambda c.p(\pi_2\pi_1 c)yxc}{\qquad} \qquad \dfrac{\text{todoke-ta}\quad S_{attr}\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o\quad : \lambda z.\lambda y.\lambda x.\lambda c.\mathbf{deliver}(x,y,z)}{\qquad}}{S_{attr}\backslash NP_{ga}\backslash NP_{ni}\quad : \lambda y.\lambda x.\lambda c.\mathbf{deliver}(x,y,\pi_2\pi_1 c)}>}{S_{attr}\backslash NP_{ga}\quad : \lambda x.\lambda c.\mathbf{deliver}(x,mary,\pi_2\pi_1 c)}>}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga})\quad : \lambda p.p(bear)}$$

$$S_{attr}\quad : \lambda c.\mathbf{deliver}(bear, mary, \pi_2\pi_1 c)\ >$$

We also assume, again following Bekki (2010), that the Japanese empty relativizer is of category $N/N\backslash NP_{attr}$, which attaches to the clause (9) of attributive form to form a nominal modifier as in (10). We further assume that Japanese has an empty determiner

[5]The complex-NP violation construction is known to be more acceptable if the dislocated NP is *ni*-marked (dative) rather than *o*-marked (accusative), and a “heavy” dislocated NP (as in (7a)) tends to make it more acceptable.

[6]Therefore, no movement is involved in deriving relative clauses in Japanese. This analysis is motivated by the widely acknowledged fact that Japanese relativization does not show any kind of subjacency, and the *pro* can be replaced by overt pronouns.

[7]In (12), the *pro* is semantically interpreted as anaphoric. For the sake of simplicity, we use the representation in which the anaphora is already resolved: $@_1 = \lambda c.\pi_2\pi_1(c)$. See the anaphora resolution process in Bekki (2014).

that is either existential, anaphoric or generic. Suppose that the existential one is chosen as below.

(10)

$$\dfrac{\dfrac{\textit{(determiner)}}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{nc})/N} \quad \dfrac{\dfrac{\dfrac{\textit{(relativizer)}}{N/N\backslash S_{attr}} \quad (9)}{N/N} < \quad \dfrac{\text{iyaringu}}{N}}{N} >}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{nc})} >$$

$$: \lambda n.\lambda p.\lambda x.\lambda c.\begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} v{:}ny(c,y) \\ pyx(c,(y,v)) \end{bmatrix} \end{bmatrix} \quad : \lambda p.\lambda n.\lambda x.\lambda c.\begin{bmatrix} u{:}nx(c,x) \\ p(c,(x,u)) \end{bmatrix} \quad : \lambda n.\lambda x.\lambda c.\begin{bmatrix} u{:}nx(c,x) \\ \mathbf{deliver}(bear, mary, \pi_2 c) \end{bmatrix} \quad : \lambda x.\lambda c.\mathbf{earring}(x)$$

$$: \lambda x.\lambda c.\begin{bmatrix} u{:}\mathbf{earring}(x) \\ \mathbf{deliver}(bear, mary, \pi_2 c) \end{bmatrix}$$

$$: \lambda p.\lambda x.\lambda c.\begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} v{:}\begin{bmatrix} u{:}\mathbf{earring}(y) \\ \mathbf{deliver}(bear, mary, y) \end{bmatrix} \\ pyx(c,(y,v)) \end{bmatrix} \end{bmatrix}$$

Then the accusative case marker *o* attaches to (10) to form an accusative argument. The derivation of the whole sentence (8a) is shown as (11).

(11)

$$\dfrac{\text{Taro-ga} \quad \dfrac{\dfrac{\dfrac{\text{-o}}{\boldsymbol{T}\backslash NP_{nc}/(\boldsymbol{T}\backslash NP_o)} \quad (10)}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_o)} >\mathbf{B} \quad \dfrac{\text{hakkensi-ta}}{S_{term|attr}\backslash NP_{ga}\backslash NP_o}}{S_{term|attr}\backslash NP_{ga}} >}{S_{term|attr}} >$$

$$\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga}) \quad : \lambda p.p(taro) \quad : \lambda x.\lambda c.\begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} v{:}\begin{bmatrix} u{:}\mathbf{earring}(y) \\ \mathbf{deliver}(bear, mary, y) \end{bmatrix} \\ \mathbf{find}(x,y) \end{bmatrix} \end{bmatrix} \quad : \lambda p.\lambda x.\lambda c.\begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} v{:}\begin{bmatrix} u{:}\mathbf{earring}(y) \\ \mathbf{deliver}(bear, mary, y) \end{bmatrix} \\ pyx(c,(y,v)) \end{bmatrix} \end{bmatrix} \quad : \lambda y.\lambda x.\lambda c.\mathbf{find}(x,y)$$

$$: \lambda c.\begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} v{:}\begin{bmatrix} u{:}\mathbf{earring}(y) \\ \mathbf{deliver}(bear, mary, y) \end{bmatrix} \\ \mathbf{find}(taro, y) \end{bmatrix} \end{bmatrix}$$

4.2 Extraction from complex-NP

Contrary to the standard derivation in (9)(10)(11), the sentence (7a), repeated below, is derived in three steps.

(7a) ?? [...ozyoosan]-ni$_1$ [Taro-wa [kuma-ga t_1 todoke-ta iyaringu]-o hakkensi-ta]
the.lady-DAT Taro-TOP the.bear-NOM t_1 deliver-PAST-attr earring-ACC find-PAST-term

In the first step, the embedded verb *todoke-ta*, after merging with its accusative argument *pro*, composes with the empty relativizer. By means of the functional composition rule $>\mathbf{B}^2$, it merges with a transitive verb and yields a relative clause that leaves both the nominative and dative arguments unsaturated.

$$\dfrac{\dfrac{\underline{(12)}\quad \dfrac{pro_1}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_o)}}{\begin{array}{l}: \lambda p.\lambda y.\lambda x.\lambda c.p(\pi_2\pi_1 c)yx\end{array}} \quad \dfrac{\text{todoke-ta}}{\begin{array}{l}S_{attr}\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o \\ : \lambda z.\lambda y.\lambda x.\lambda c.\mathbf{deliver}(x,y,z)\end{array}}}{\begin{array}{c}S_{attr}\backslash NP_{ga}\backslash NP_{ni} \\ : \lambda y.\lambda x.\lambda c.\mathbf{deliver}(x,y,\pi_2\pi_1 c)\end{array}}> \quad \dfrac{\dfrac{}{(relativizer)}}{\begin{array}{c}N/N\backslash S_{attr} \\ : \lambda p.\lambda n.\lambda x.\lambda c.\begin{bmatrix} u{:}nx(c,x) \\ p(c,(x,u)) \end{bmatrix}\end{array}}}{\begin{array}{c}N/N\backslash NP_{ga}\backslash NP_{ni} \\ : \lambda y.\lambda x.\lambda n.\lambda z.\lambda c.\begin{bmatrix} u{:}nz(c,z) \\ \mathbf{deliver}(x,y,\pi_2(c)) \end{bmatrix}\end{array}}>\mathbf{B}^2$$

In the second step, a constituent consisting of the matrix subject and the embedded subject (NP-nom$_1$ and NP-nom$_2$ in (7b)) is derived. This is a non-standard, distinctive constituent in our analysis, which we call a *subject chunk*.

(13)

$$\dfrac{\dfrac{\text{... ozyoosan-ni}}{\begin{array}{l}\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ni}) \\ : \lambda p.p(lady)\end{array}} \quad \dfrac{\dfrac{\text{Taro-wa}}{\begin{array}{l}\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga|ni|o}) \\ : \lambda p.p(taro)\end{array}} \quad \dfrac{\dfrac{}{(determiner)}}{\begin{array}{l}\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{nc})/N \\ : \lambda n.\lambda p.\lambda x.\lambda c.\begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} v{:}ny(c,y) \\ pyx(c,(y,v)) \end{bmatrix} \end{bmatrix}\end{array} \quad \dfrac{\text{kuma-ga}}{\begin{array}{l}\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga}) \\ : \lambda p.p(bear)\end{array}}}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{nc})/N/(N/N\backslash NP_{ga})}>\mathbf{B}^2\ (\boldsymbol{T}{=}N/N)}{\begin{array}{l}\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga}\backslash NP_{nc})/N/(N/N\backslash NP_{ga}) \\ : \lambda p.\lambda n.\lambda q.\lambda x.\lambda c.\begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} v{:}p(bear)ny(c,y) \\ qyx(c,(y,v)) \end{bmatrix} \end{bmatrix}\end{array}}>\mathbf{B}^3}{\begin{array}{l}\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga}\backslash NP_{nc})/N/(N/N\backslash NP_{ga}/NP_{ni}) \\ : \lambda p.\lambda n.\lambda q.\lambda c.\begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} v{:}p(lady)(bear)ny(c,y) \\ qytaro(c,(y,v)) \end{bmatrix} \end{bmatrix}\end{array}}>\mathbf{CB}$$

There are two key aspects that enable the formation of subject chunks: 1) the existence of an empty determiner in a Japanese noun phrase. In the derivation (13), the empty determiner for the object NP *iyaringu* mediates the composition of the matrix subject and the embedded subject. 2) When the empty determiner combines with the embedded subject *kuma-ga* by application of the $>\mathbf{B}^2$ rule, the category variable $\boldsymbol{T}$ in the category of the embedded subject (i.e., $\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga})$) unifies with N/N, instead of with N.

The existence of subject chunks is supported by the fact that the following coordinated structure yields an acceptable sentence. (cf. Mukai (2003))

(14) a. ? Taroo-wa [kuma-ga, ∅ ziroo-ha [raion-ga, ozyoosan-ni
Taro-TOP bear-NOM and Jiro-TOP lion-NOM lady-DAT
todoke-ta iyaringu]-o hakkensi-ta
deliver-PAST-attr earring-ACC find-PAST-term
(lit.) 'Taro found an earring that the bear delivered to the lady, and Jiro found an earring that the lion delivered to the lady.'

b. ? NP-nom$_1$ NP-nom$_2$ and NP-nom$_1$ NP-nom$_2$ NP-acc$_2$ V$_2$-N-acc$_1$ V$_1$

It may be controversial whether Japanese grammar (or a parser) allows the category variable $\boldsymbol{T}$ to be unified with N/N, which itself does not appear in the category of the primary argument.

Subject chunks do not exist in SVO languages such as English, while they are derivable in SOV languages, such as Japanese, given that the $>\mathbf{B}^2$ and $>\mathbf{B}^3$ rules are available[8]. A subject chunk is not a normal form, and so they appear in only long-distance scrambling constructions (otherwise, it is excluded by the usual technique of normal-form parsing).

In the third step, in the last line in (13), the dislocated (heavy) NP *mori-no-kuma-san-no moderu-to nat-ta ozyoosan-ni* (abbreviated as *...ozyoosan-ni* in (13)) and the subject chunk *Taro-wa kuma-ga* are combined by the permuted functional composition rule $>\mathbf{CB}$. This operation *reconstructs* the semantic representation of the dislocated NP, sending it back to the trace position in the embedded clause, namely, the position on the right of the embedded subject.

The resulting constituent of (13) is of category $\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga}\backslash NP_{nc})/N/(N/N\backslash NP_{ga})$, which is a non-standard category, yet combines successively with the output of (12), the common noun *iyaringu*, the accusative case marker *-o*, and the embedded verb *mituke-ta* as in (15), yielding the derivation of the whole sentence (7a).

(15)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{...ozyoosan-ni Taro-wa kuma-ga}}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga}\backslash NP_{nc})/N/(N/N\backslash NP_{ga}/NP_{ni})}\,(13) \quad \dfrac{\text{todoke-ta}}{N/N\backslash NP_{ga}\backslash NP_{ni}}\,(12)}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga}\backslash NP_{nc})/N}>}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga}\backslash NP_{nc})}> \quad \dfrac{\text{iyaringu}_1}{N}}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga}\backslash NP_{o})}>_{\mathbf{B}} \quad \dfrac{\text{-o}}{\boldsymbol{T}\backslash NP_{nc}/(\boldsymbol{T}\backslash NP_{o})}}{S_{term}}> \quad \dfrac{\text{mituke-ta}}{S_{term}\backslash NP_{ga}NP_{o}}$$

This analysis can be adopted in most modern frameworks of categorial grammars, such as Hybrid TLCG (Kubota, 2010), displacement calculus (Morrill, 2010), and abstract categorial grammar (de Groote, 2001), so long as a rule corresponding to $>\mathbf{CB}$ can be added.

5 Predictions on non-standard constituents

Our analysis yields at least two empirical predictions. First, since a dislocated NP and a subject chunk form a constituent as (13) shows, a pair of such constituents can be

[8]The use of $<\mathbf{B}^2$ and $<\mathbf{B}^3$ rules are considered to be common in Japanese, notably when an auxiliary verb attaches to a transitive and a ditransitive verb: for example, *syookaisi* (a transitive verb of category $S\backslash NP\backslash NP\backslash NP$) and *ta* (a past suffix of category $S\backslash S$) are combined by the $<\mathbf{B}^3$ rule.

coordinated. This prediction is supported by the fact that (16a), schematized as (16b), is acceptable under appropriate contexts.

(16) a. ?? ozyoosan-ni$_1$ [Taro-wa [kuma-ga t_1 ∅
the.lady-DAT Taro-TOP the.bear-NOM t_1 and
obaasan-ni$_2$ [Ziroo-wa [raion-ga t_2 todoke-ta
the.old.lady-DAT Jiro-TOP the.lion-NOM t_2 deliver-PAST-attr
iyaringu]-o hakkensi-ta]
earring-ACC find-PAST-term
(lit.) 'Taro found an earring that the bear delivered to the lady, and Jiro found an earring that the lion delivered to the old lady.'

b. ?? NP-dat$_2$ NP-nom$_1$ NP-nom$_2$ and NP-dat$_2$ NP-nom$_1$ NP-nom$_2$ V$_2$-rel-N-CM$_1$ V$_1$

The sentence (16a) is derivable since the subject chunks are constituents as shown in (13), thus can be conjoined as (17). The rest of the sentence should be derived in the same manner as (15).

$$\dfrac{\dfrac{\text{(17) ozyoosan-ni Taroo-wa kuma-ga}}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga}\backslash NP_{nc})/N/(N/N\backslash NP_{ga}/NP_{ni})}\ (13)\quad \dfrac{\text{obaasan-ni Ziroo-wa raion-ga}}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga}\backslash NP_{nc})/N/(N/N\backslash NP_{ga}/NP_{ni})}\ \approx (13)}{\boldsymbol{T}/(\boldsymbol{T}\backslash NP_{ga}\backslash NP_{nc})/N/(N/N\backslash NP_{ga}/NP_{ni})}\ \langle\Phi\rangle$$

$$: \lambda p.\lambda n.\lambda q.\lambda c.\begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} v{:}p(younglady)(bear)ny(c,y) \\ qytaroo(c,(y,v)) \end{bmatrix} \end{bmatrix} \quad : \lambda p.\lambda n.\lambda q.\lambda c.\begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} v{:}p(oldlady)(rion)ny(c,y) \\ qyziroo(c,(y,v)) \end{bmatrix} \end{bmatrix}$$

$$: \lambda p.\lambda n.\lambda q.\lambda c.\begin{bmatrix} w{:}\begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} v{:}p(younglady)(bear)ny(c,y) \\ qytaroo(c,(y,v)) \end{bmatrix} \end{bmatrix} \\ \begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} v{:}p(oldlady)(rion)ny((c,w),v) \\ qyziroo((c,w),(y,v)) \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

However, the presence of this construction is fatal to most minimalist analyses while the construction is derivable in most modern categorial grammars. In this sense, the acceptability of the scheme (16b) is a strong support for categorial grammars over minimalist frameworks.

Second, in the derivation of (7a), long-distance scrambling construction forces a formation of a subject chunk, which is a non-normal form. It is predicted that subsequent elements—namely, an embedded predicate *todoke-ta* (of category $N/N\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o$), and a common noun *iyaringu* (of category N)—do not form a constituent in these non-normal form derivations, even though they do in normal form derivations.

(18)

$$\dfrac{\dfrac{\dfrac{\text{todoke-ta}}{S_{attr}\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o}}{: \lambda z.\lambda y.\lambda x.\lambda c.\mathbf{deliver}(x,y,z)} \quad \dfrac{\textit{(relativizer)}}{N/N\backslash S_{attr}}}{: \lambda p.\lambda n.\lambda x.\lambda c.\begin{bmatrix} u{:}nx(c,x) \\ p(c,(x,u)) \end{bmatrix}}\ >\mathbf{B}^3 \qquad \dfrac{\text{iyaringu}}{N}$$

$$N/N\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o \qquad : \lambda x.\lambda c.\mathbf{earring}(x)$$

$$: \lambda z.\lambda y.\lambda x.\lambda n.\lambda x.\lambda c.\begin{bmatrix} u{:}nx(c,x) \\ \mathbf{deliver}(x,y,z) \end{bmatrix}$$

$$*\qquad\qquad *$$

This prediction is also supported by the fact that the sentence (19a), schematized as (19b), is not acceptable, in which the subsequent elements are coordinated.

(19) a. * the.lady-DAT Taro-TOP the.bear-NOM t_1 deliver-PAST-attr earring-ACC and give.as.a.gift-PAST-attr necklace-ACC find-PAST-term

(lit.) 'Taro found an earring that the bear delivered to the lady and a necklace that the bear gave the lady as a gift.'

b. * NP-acc$_2$ NP-nom$_1$ NP-nom$_2$ $\underline{\text{V}_2\text{-rel-N-CM}_1}$ and $\underline{\text{V}_2\text{-rel-N-CM}_1}$ V$_1$

Unlike the first prediction, the unacceptability of the scheme (19b) would be problematic for most modern categorial grammars with a full-fledged deduction theorem. In other words, it is difficult to both derive (16b) and block (19b). In the full paper, we show how the sentence (19a) is not derivable in CCG while it is derivable in other frameworks that are substructurally stronger than CCG.

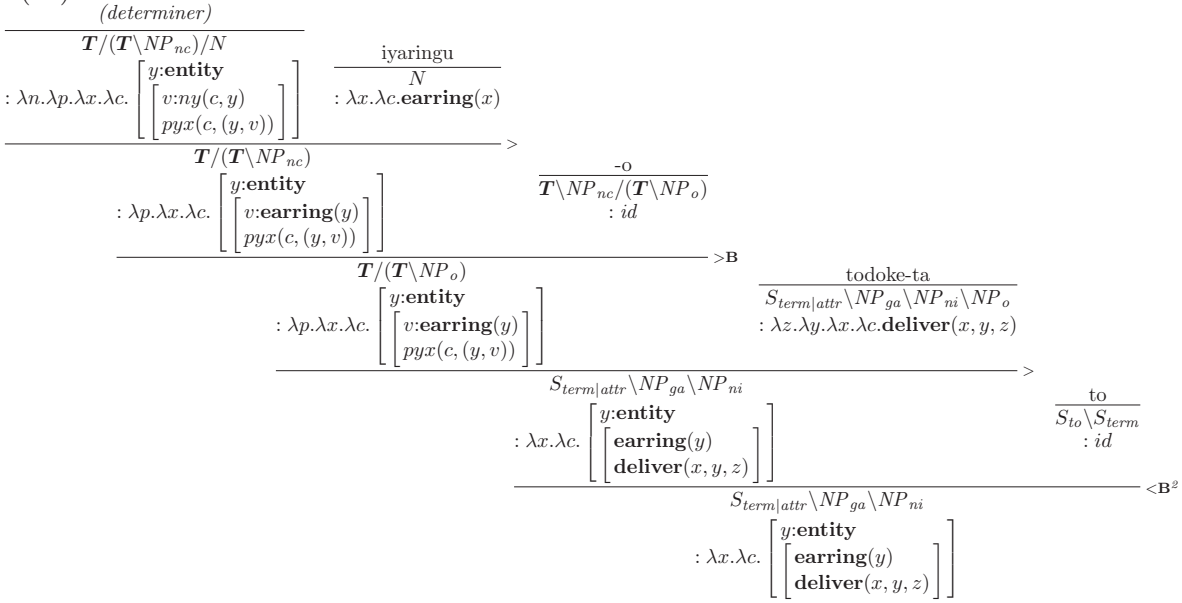
It might be tempting to attribute the unacceptability of (19b) to its syntactic complexity. However, this is falsified by the clear acceptability of the sentence (20a), schematized as (20b); this scheme is as complex as (19a). The difference between (19a) and (20a) is that the embedded clause in the former is a relative clause, while it is a that-clause in the latter.

(20) a. ? ozyoosan-ni$_1$ [Taro-wa [kuma-ga t_1
the.lady-DAT Taro-TOP the.bear-NOM t_1
$\underline{\text{iyaringu-o todoke-ta-to}}$ sosite
earring-ACC deliver-PAST-term-that and
$\underline{\text{kubikazari-o purezentosi-ta-to}}$ omoikonde-i-ta.]]
necklace-ACC give.as.a.gift-PAST-term-that believe-PROG-PAST-term

(lit.) 'Taro wrongly believed that the bear delivered an earring and gave a necklace as a gift to the lady.'

b. ? NP-acc$_2$ NP-nom$_1$ NP-nom$_2$ [NP-acc$_2$ V$_2$-that] and [NP-acc$_2$ V$_2$-that] V$_1$

(21)



References

Bekki, Daisuke. 2010. *Nihongo-Bunpoo-no Keisiki-Riron - Katuyootaikei, Toogohantyuu, Imigoosei - (trans. 'Formal Japanese Grammar: the conjugation system, categorial syntax, and compositional semantics')*. Tokyo: Kuroshio Publisher.

Bekki, Daisuke. 2014. Representing anaphora with dependent types. In N. Asher and S. V. Soloviev, eds., *Logical Aspects of Computational Linguistics (8th international conference, LACL2014, Toulouse, France, June 2014 Proceedings), LNCS 8535*, 14–29. Toulouse: Springer, Heiderburg.

Coquand, Thierry and Gérard Huet. 1988. The calculus of constructions. *Information and Computation* 76(2-3):95–120.

de Groote, Philippe. 2001. Towards abstract categorial grammers. In *ACL 2001*. Toulouse, France.

Harada, S.-I. 1977. Nihongo-ni 'henkei'-wa hituyoo-da ('transformation' is necessary for japanese). *Gengo* 6(11-12).

Hoji, Hajime. 1985. *Logical Form Constraints and Configurational Structures in Japanese*. Doctoral dissertation, University of Washington.

Kang, Beom-mo. 1987. *Functional Inheritance, Anaphora, and Semantic Interpretation in a Generalized Categorial Grammar*. Ph.d. thesis, Brown University.

Komagata, Nobo. 1999. *A Computational Analysis of Information Structure Using Parallel Expository Texts in English and Japanese*. Ph.D. thesis, University of Pennsylvania.

Krahmer, Emiel and Paul Piwek. 1999. Presupposition projection as proof construction. In H. Bunt and R. Muskens, eds., *Computing Meanings: Current Issues in Computational Semantics*, Studies in Linguistics Philosophy Series. Dordrecht: Kluwer Academic Publishers.

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

Martin-Löf, Per. 1984. *Intuitionistic Type Theory*, vol. 17. Naples: Italy: Bibliopolis. Sambin, Giovanni (ed.).

Morrill, Glyn V. 2010. *Categorial Grammar: Logical Syntax, Semantics, and Processing*. Oxford: Oxford University Press.

Mukai, Emi. 2003. On verbless conjunction in japanese. In *NELS33*.

Ozaki, Hiroko and Daisuke Bekki. 2012. Extractability as deduction theorem in subdirectional combinatory logic. In *Logical Aspect of Computational Linguistics (LACL2012)*. Nantes, France.

Saito, Mamoru. 1992. Long distance scrambling in japanese. *Journal of East Asian Linguistics* 1(1):69–118.

Saito, Mamoru. 2003. A derivational approach to the interpretation of scrambling chains. *Lingua* 113(4-6):481–518.

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

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

Steedman, Mark J. 2000. *The Syntactic Process (Language, Speech, and Communication)*. Cambridge: The MIT Press.

Ueyama, Ayumi. 1998. *Two Types of Dependency*. Doctoral dissertation, University of Southern California. distributed by GSIL publications.

Ueyama, Ayumi. 2003. Two types of scrambling constructions in japanese. In A. Barss, ed., *Anaphora: A Reference Guide*. Cambridge: Blackwell.

van der Sandt, Rob. 1992. Presupposition projection as anaphora resolution. *Journal of Semantics* 9:333–377.

A Categorial Analysis of Chinese Alternative Questions

Manjuan Duan
Ohio State University

Abstract

This paper discusses the syntax and semantics of Chinese alternative questions in Linear Categorial Grammar (LCG), a linear-logic based form of categorial grammar (Pollard, In Press; Mihaliček and Pollard, 2012; Pollard and Smith, 2012). We propose a lexical entry for *háishì*, which conjoins alternatives to form alternative questions in Chinese. The lexical entry proposed provides a straightforward composition of the syntax and semantics of alternative questions in Mandarin Chinese. Our analysis based on this lexical entry shows that alternative questions are different from *wh*-questions in that a *háishì* disjunction in an alternative question does not scope over a question as a *wh*-phrase does. Our analysis also shows that an A-not-A verb phrase, unlike a *háishì* disjunction, cannot scope over its continuation, which provides a straightforward explanation for the difference observed by Huang (1982) between A-not-A questions and alternative questions in terms of their (in)sensitivity of island constraints.

Keywords: alternative questions; categorial grammar, Mandarin Chinese

1 Introduction

Alternative questions are questions which present multiple possible answers for the hearer to choose from. They are like *wh*-questions in that they cannot be answered by *yes* or *no*. At the same time, they are like polar questions in that the hearer is expected to choose an answer from a set of mutually exclusive propositions.

In Mandarin Chinese, alternative questions are formed by conjoining disjuncts with *háishì*, as shown in (1). For the question presented in (1), the speaker expects the hearer to choose one of the alternatives as the answer. So for (1), an ideal answer would be *chá* 'tea' or *kāfēi* 'coffee'. It is unacceptable to respond it with *shì de* 'yes' or *bú shì* 'no'.

(1) yūehàn yào hē kāfēi háishì chá?
John want drink coffee HAISHI tea
'Does John want to drink coffee ↑ or tea↓?'[1].

Syntactically, *háishì* can conjoin disjuncts of various syntactic categories. They can be NPs as in (1), VPs as in (2), verbs as in (3) or sentences, as in (4).

I am indebted to Carl Pollard, Judith Tonhauser and Robert Levine for their valuable comments. I am also grateful to Murat Yasavul and Chris Worth for many helpful discussions. I would also like to thank the participants of SWAMP 2012 and the Synners discussion group for their comments and suggestions. This research was supported by a TIE Fellowship of the Department of Linguistics at OSU in the Summer of 2013.

[1] ↑ and ↓ are used to label alternative questions in English translation, because alternative questions and disjunctive polar questions in English are only prosodically different(Pruitt and Roelofsen, 2013; Heidenreich et al., 2014)

(2) zhāngsān xǐhuān wángwǔ háishì bùxǐhuān wángwǔ?
Zhangsan like Wangwu HAISHI not-like Wangwu?
'Does Zhangsan like Wangwu or not?'

(3) zhāngsān xǐhuān háishì bùxǐhuān wángwǔ?
Zhangsan like HAISHI not-like Wangwu?
'Does Zhangsan like Wangwu or not?'

(4) zhāngsān xǐhuān wángwǔ háishì wángwǔ xǐhuān zhāngsān?
Zhangsan like Wangwu HAISHI Wangwu like Zhangsan?
'Does Zhangsan like Wangwu or does Wangwu like Zhangsan?'

In Mandarin Chinese, the question particle *ma* is used at the end of the main clause to convert a declarative sentence into a polar question. (5) shows that *ma* cannot occur at the end of a *háishì* sentence, which indicates that *háishì* comes with an interrogative mood.

(5) *yūehàn yào hē kāfēi háishì chá ma?
John want drink coffee HAISHI tea Q
Intended: 'Does John want to drink coffee ↑ or tea↓?'

Semantically, *háishì* supposes that the disjuncts are mutually exclusive. (6) sounds unnatural because the two alternatives are not mutually exclusive in terms of our common knowledge: a math book can be a textbook.

(6) #zhāngsān zài kàn shùxúeshū háishì jìaokēshū?
Zhangsan in read math book HAISHI textbook?
Intended: 'Is Zhangsan reading a math book ↑ or a textbook ↓?'

More specifically, *háishì*-sentences present a set of alternatives together with the persistent entailment that exactly one of the alternatives is true; ideally, a felicitous answer identifies the true alternative. When (1) is asked, the speaker *expects* the hearer to choose one of the alternatives as the true answer.

Both Huang (1982) and Erlewine (2012) treat the *háishì* disjunction in Chinese as a *wh*-phrase. According to Huang (1982), the *háishì* disjunction undergoes covert *wh*-movement like a *wh*-phrase in Chinese; both occur in situ, but they move covertly to [Spec, CP] at LF to yield the desired interpretation. Erlewine (2012), based on Beck 2006, proposes that the *háishì* disjunction stay in-situ but the focus semantic value of the *háishì* disjunction propagates to TP, where it is elevated into ordinary semantic value by the Q operator.

However both analyses fail to explain why (7) is grammatical in Chinese but (8) is not if the *háishì* disjunction is treated as a wh-phrase .

(7) shéi xǐhuān shéi?
who like who
'Who likes who?'

(8) *shéi xǐhuān Lǐsì háishì Wángwǔ?
who like Lisi HAISHI Wangwu
Not interpretable.

Huang (1991) hypothesizes alternative questions are different from A-not-A questions[2] in terms of their sensitivity of island constraints, when he observes that an alternative question can be embedded within a relative clause or a sentential subject clause while an A-not-A question cannot.

[2]According to Huang et al. (2009), A-not-A questions are a special type of alternative questions in Chinese in addition to polar questions and constituent questions. We will introduce this type of questions formally later.

(9) a. wǒ qù měigúo háishì bú qù měigúo bǐjìaohǎo?
I go America HAISHI not go America better
'Is it better for me to go to America↑ or not↓?'

b. *wǒ qù-bú-qù měigúo bǐjìaohǎo?
I go-not-go America better?
Intended meaning: 'Is it better for me to go to America↑ or not↓?'

(10) a. nǐ xǐhūan rènshí nǐ háishì bú rènshí nǐ de rén?
you like know you HAISHI not know you DE people
'Do you like the people who know you↑ or people who do not know you↓?'

b. *nǐ xǐhūan rèn-bú-rènshí nǐ de rén?
you like know-not-know you DE people
Intended meaning: 'Do you like the people who know you↑ or people who do not know you↓?'

In (9a), the *háishì* construction occurs within a subordinate subject clause and in (10a), it occurs within a relative clause. However, replacing the *háishì* construction with an A-not-A verb in (9b) and (10b) make the two sentences ungrammatical. Huang (1991), thus, concludes that the *háishì* construction and A-not-A questions are different in terms of their sensitivity to island constraints.

However, Huang (1991) does not provide an explanation for this difference, despite the fact that A-not-A questions and alternative questions are sometimes semantically interchangeable.

In this paper, we analyze Chinese alternative questions in a categorial grammar formalism. We will show that a properly defined lexical entry for *háishì* can not only yield the expected semantics for alternative questions, but also account for the difference between *háishì* disjunctions and *wh*-phrases illustrated in (7) and (8). We will also show that the seemingly different island sensitivity between alternative questions and A-not-A questions can be accounted for within the same framework without resorting to the notions like movement or island.

2 Background on Chinese Alternative Questions

2.1 *wh*-movement of the interrogative disjunction

J.Huang (1982, p.273) treats *háishì* disjunction the same as a *wh*-phrase. In his analysis, the *háishì* disjunction undergoes covert *wh*-movement like a *wh*-phrase in Chinese. In (11a), the interrogative disjunction, 'Zhangsan haishi Lisi', is moved to the [Spec, CP] at LF.

(11) *háishì* disjunction as *wh* movement (Huang, 1982, p.273)

a. $[_{\text{CP}}$ [Zhangsan haishi Lisi]$_i$ [t$_i$ hui lai]]?
Zhangsan HAISHI Lisi will come
'Will Zhangsan ↑ or Lisi ↓ come?'

b. $[_{\text{CP}}$[neige x, x ∈ {Zhangsan, Lisi}], [x hui lai]]
Which x will come

Here, J.Huang argues that the meaning of (11a) can be represented as (11b), 'which x, $x \in$ {Zhangsan, Lisi}, will come?'. By interpreting (11a) as (11b), J.Huang argues that

the *háishì* construction in (11a) works like a *wh*-phrase. Both occur in situ, but they have to move covertly to the Spec CP position at LF to yield the desired interpretation.

However, according to Huang (1991), the covert movement of*háishì* disjunction, different from the movement in A-not-A questions, is not subject to sentential subject and relative clause island constraints, based on the observation of the sentences in (9) and (10).

2.2 Alternative Questions in Alternative Semantics

Erlewine (2012) analyzes alternative question in Mandarin Chinese within the framework of alternative semantics.[3] He based his analysis of alternative questions in Mandarin Chinese on Beck (2006), who provides a semantic analysis of intervention effects in wh-questions. According to Beck (2006), both *wh*-phrases and focused phrases introduce alternatives into the computation of compositional interpretation, but unlike focused phrases, *wh*-phrases make no ordinary semantic contribution. Therefore, the ordinary semantics of *wh*-phrases are undefined. However, "since wh-phrases occur in expressions that have a perfectly well-defined ordinary semantic value, something must rescue the structure as a whole from undefinedness. This is the role of the question operator, Q." (Beck, 2006, p.12). She proposes that "semantics of Q elevate its sister's focus semantic value to the ordinary semantics."(Beck, 2006, p.12). However, questions containing a focus whose contribution is evaluated within the scope of the Q operator are not well-defined, therefore ungrammatical. This situation is schematized in (12).

(12) [Q ...[OP [$_\phi$ ...XP$_F$ ...wh ...]]]

For the focus on XP to be evaluated within the scope of the Q operator means that there is a focus-sensitive operator, OP in (12), which uses both the ordinary and the focus semantic value of ϕ. For example, in order to derive the semantics of "Only John left", we need to consider both the proposition that John left, and the alternatives propositions 'that x left' for alternatives x to John. However, since the *wh*-phrase does not have an ordinary semantic value, the ordinary semantic value of ϕ is consequently undefined. In general, a *wh*-phrase not immediately c-commanded by a coindexed Q operator will be uninterpretable, since the expression it is contained in can never have a well-defined ordinary interpretation. A wh-phrase c-commanded by an intervening focus-sensitive operator (the $\sim$ operator) will lead to uninterpretability despite a c-commanding Q operator, because the $\sim$ operator makes use of both the ordinary and focus semantic interpretation of its sister, and it resets the focus semantics to the ordinary semantics. The Q-operator is the only binder for distinguished variables that uses just the focus semantic interpretation. Therefore the structure (12) is excluded by Beck (2006)'s analysis.

Beck (2006) also mentioned that there is no intervention effect in much English data because at the relevant level, Logical Form, there is no intervention configuration. The presence of the intervention configuration, according to the paper, crucially, depends on what movement the *wh*-phrase undergoes. For example, the English sentence, "Who did only John see?", is not subject to intervention effects because the *wh*-phrase 'who' undergoes overt movement, which does not trigger intervention effects, while the covert *wh*-movements in those languages where *wh*-phrases occur in-situ trigger intervention effects.

[3]Shan (2004) points out some technical difficulties associated with alternative semantics framework in general. He points out that the compositional, bottom-up computation of alternative sets is incompatible with major theories of quantification which use variables for binding. A detailed explanation of his point is out of the scope of this paper, but readers are encouraged to read Shan (2004) for a detailed explanation.

Following Beck (2006), Erlewine (2012) assumes that *háishì* disjunction introduces focus alternatives as does a wh-phrase. Also the *háishì* disjunction does not have ordinary semantic values as a *wh*-phrase. Therefore, in (1) the focus semantic value of the *háishì* disjunction, "coffee HAISHI tea", is the set of ordinary semantic values of its disjuncts, 'coffee' and 'tea'. The ordinary semantic value of "coffee HAISHI tea" is undefined, as illustrated in (13).

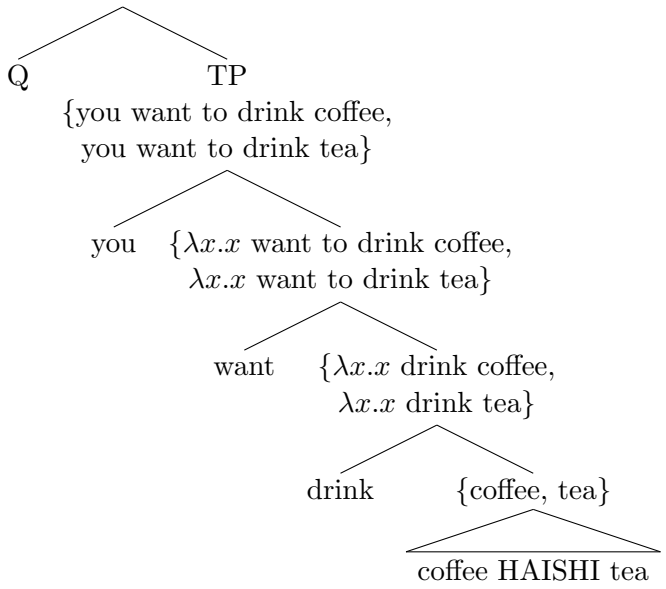
(13) Erlewine (2012, p.225)
$[\![\text{coffee HAISHI tea}]\!]^{f} = \{\text{coffee, tea}\}$, $[\![\text{coffee HAISHI tea}]\!]^{o}$ undefined

The focus semantic value of the *háishì* disjunction propagates upward along the tree until it is elevated to an ordinary semantic value by the Q operator at TP to yield an interpretable sentence, as shown in (16).

(14) Erlewine (2012, p.226)
$[\![\text{TP}]\!]^{f} = \{\text{you want to drink coffee, you want to drink tea}\}$
$[\![\text{TP}]\!]^{o}$ undefined

(15) Erlewine (2012, p.226)
$[\![\text{Q TP}]\!]^{o} = [\![\text{TP}]\!]^{f} = \{\text{you want to drink coffee, you want to drink tea}\}$

(16) Computation of focus-semantic values (Erlewine, 2012, p.2)



Beck and Kim (2006) proposes that the disjunction in the derivation of English alternative questions does not undergo any syntactic movement. Following Beck and Kim (2006), Erlewine (2012) assumes that the disjunction 'coffee HAISHI tea' stays *in situ* throughout the derivation. Since the *háishì* disjunction does not move, *háishì* alternative questions are not sensitive to sentential subject and relative clause island constraint.

However, the empirical evidence for adopting the assumption that *háishì* disjunction does not move in Mandarin alternative questions is not thoroughly pursued here. First, this assumption cannot explain the presence of a Q-operator at [Spec, CP]. Erlewine (2012) seems to suggest that the *háishì* disjunction does not move to the complementizer because association with focus between a focus-operator and a focus-marked constituent does not obey syntactic island (Erlewine, 2012, p.226). If we adopt this assumption, the

Q operator works as the $\sim$ operator defined in Rooth (1985) and the *háishì* disjunction is a focus-marked constituent, which should have both ordinary and focus semantic values, as claimed by Beck (2006). Then we should not be able to observe any intervention effects according to Beck (2006). Analyzing *háishì* disjunction as a focused phrase contradicts his own claim that Mandarin alternative questions are sensitive to intervention effects. If Erlewine (2012) argues that Mandarine alternative questions are subject to intervention effects, the *háishì* disjunction has to be analyzed as a *wh*-phrase, and then the analysis needs to explain how the Q-operator moves to [Spec, CP] at LF.

According to Erlewine (2012), although Mandarin alternative questions are not subject to sentential subject or relative island, they *are* sensitive to *wh* -islands. (17) and (18) are used to show that the *háishì* disjunction can be embedded as in (17), but cannot occur within a *wh*-island as in (18).

(17) *háishì* disjunction in an embedded clause (Erlewine, 2012, p.227)

ni juede [Zhangsan xihuan Lisi haishi WangWu]?
you think Zhangsan like Lisi HAISHI Wangwu

'Do you think Zhangsan likes Lisi ↑ or Wangwu ↓?'

(18) *háishì* in a *wh*-island:

ni xiangzhidao shei xihuan Lisi haishi Wangwu?
you wonder who like Lisi HAISHI Wangwu

Intended: 'Is it Lisi or Wangwu that you wonder who likes?'

(19) [$_{\text{CP}_1}$Q$_1$ you wonder[$_{\text{CP}_2}$Q$_2$ who like[Lisi HAISHI Wangwu]]]

The underlying syntactic structure of (18) is shown in (19). Erlewine (2012) argues that (18) is not grammatical because "question-embedding verbs such as *xiangzhidao* 'wonder' requires that their complements be headed by Q, which will convert CP_2's focus semantic value into an ordinary semantic value. The complement of Q_1 will no longer contain multiple focus alternatives, so CP_1 cannot be interpreted as a question."(Erlewine, 2012, p.227).

This explanation is confusing and problematic in several aspects. There are two Q-operators in (19). However, Erlewine (2012) does not indicate in (19) which Q-operator associates with which wh-phrase. I venture to assume Q_1 is associated with the *háishì* disjunction and Q_2 is associated with 'who', because it seems the alternative question interpretation takes wide scope over 'who' in the intended interpretation provided in (18). Assuming Beck (2006), '[Lisi HAISHI Wangwu]' does not have an ordinary semantic value. Erlewine (2012) seems to suggest Q_2 elevates the focus semantic value of '[Lisi HAISHI Wangwu]' into ordinary semantic values so CP_2 now has an ordinary semantic value and consequently 'you wonder who like Lisi HAISHI Wangwu' also has an ordinary semantic value. However, 'you wonder who like Lisi HAISHI Wangwu' cannot be the complement of Q_1 because the Q-operator is looking for a complement containing multiple focus alternatives. According to Erlewine (2012), this is the reason why (18) is not grammatical.

The first problem of this analysis has to do with whether it is possible for the focus semantic values introduced by a *wh*-phrase to be evaluated by a Q-operator not associated with that *wh*-phrase. In the case of (19), the focus semantic values of the *háishì* disjunction is evaluated by the Q_2, which is supposed to be associated with 'who'.

Secondly, if we assume the analysis is correct, CP_2 has an ordinary semantic value. According to Beck (2006), if the LF of a linguistic expression has ordinary semantic values, the linguistic expression is interpretable. Supposedly, the linguistic expression which expresses

CP_2 should be interpretable now. However, when CP_2 is realized in Mandarin Chinese, the resulting sentence, (21), is not interpretable. Also, we fail to see how intervention effects or the analysis proposed here can satisfactorily account for the ungrammaticality of (21).

(20) shéi xǐhuān shéi?
who like who
'Who likes who?'

(21) *shéi xǐhuān Lǐsì háishì Wángwǔ?
who like Lisi HAISHI Wangwu
Not interpretable.

More specifically, if we adopt Erlewine (2012)'s assumption that *háishì* disjunction functions like *wh*-phrases, we fail to explain why (20) is grammatical while (21) is not.

Finally, alternative questions like (18) are wrongly predicted to be ungrammatical in Erlewine (2012). (18) appears to be unacceptable because it is hard to find a context in which we can situate this question naturally. We can suppose a context in which a graduate student is conducting research on the dialects spoken in one of the two major Chinese cities, Beijing and Shanghai. The speaker does not remember which city it was that she is looking for subjects from. So the speaker can ask her a question like (22).

(22) nǐ xiǎng zhīdào shéi cóng Běijīng lái háishì cóng Shànghǎi lái?
you want-to know who from Beijing come HAISHI from Shanghai come
'Do you want to know who is from Beijing or who is from Shanghai?'

(23) [$_{CP_1}$Q$_1$ you want-to know[$_{CP_2}$Q$_2$ who[come from Beijing HAISHI come from Shanghai]]]

(23) shows the *háishì* disjunction 'come from Beijing HAISHI come from Shanghai' is embedded in a *wh*-island headed by 'who', as in (19). However, (22) is a perfectly acceptable sentence in Mandarin Chinese.

It is hard to see any movement-based analysis can correctly predict the grammaticality of (18) and at the same time, the ungrammaticality of (8). If the *háishì* disjunction can move to the [Spec, CP] position in (18), it should be able to undergo the same movement in (8).

In the following section, we start to analyze Mandarin alternative questions in a categorial grammar formalism. The categorial framework we assumed is completely lexicalized and compositional. Therefore we do not need to resort to notions like movement or island for the computation of the semantics. We will show that a properly defined lexical entry for *háishì* can not only yield expected semantics for alternative questions, but also account for differences between *háishì* disjunctions and *wh*-phrases which the analysis based on movement fails to. We will also show that this categorial analysis of of alternative questions can be easily extended to other types of question in Mandarin Chinese without extra technicalities.

3 Theoretical Framework

3.1 Agnostic Hyperintensional Semantics

Agnostic Hyperintensional Semantics (AHS, Plummer and Pollard, 2012; Pollard, In Press) is assumed to be the underlying semantic theory for our categorial analysis of Mandarin alternative questions. We assume AHS, instead of Montague's semantics, because AHS, adopting proposition as one of the basic types instead of as the set of worlds in which it is true, avoids the granularity problem that Montague's semantics has. This is why it is"agnostic" because it remains agnostic about the question whether propositions are sets

of worlds or worlds are sets of propositions. At the same time, AHS achieves the same rigor and simplicity as Montague's semantics. AHS employs the basic types e (entities), t (truth value), w (worlds) and p (propositions). The remaining types are formed from these basic types by using the type constructor $\rightarrow$. Entities, propositions, and functional types formed by these two are *sense* type. Truth values and worlds are not senses.

For each type of sense, there is an extensional type for the corresponding reference. In AHS the correspondence is given by the function Ext from sense types to extensional types defined recursively as follows (here A and B are sense types):

(24) **Types for Extensions of Senses**
a. $\mathrm{Ext(e)} = \mathrm{e}$
b. $\mathrm{Ext(p)} = \mathrm{t}$
c. $\mathrm{Ext(T)} = \mathrm{T}$
d. $\mathrm{Ext(A} \rightarrow \mathrm{B)} = \mathrm{A} \rightarrow \mathrm{Ext(B)}$

The familiar set of truth-value symbols, connectives and quantifiers is observed here: T, F, $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\forall$, $\exists$. In addition to this set, AHS has a set of *propositional* connectives and quantifiers. For example,

(25) **AHS Propositional Constants**

$\vdash \mathsf{truth} : \mathrm{p}$	$\vdash \mathsf{implies} : \mathrm{p} \rightarrow \mathrm{p} \rightarrow \mathrm{p}$
$\vdash \mathsf{falsity} : \mathrm{p}$	$\vdash \mathsf{forall}_A : (A \rightarrow \mathrm{p}) \rightarrow \mathrm{p}$
$\vdash \mathsf{not} : \mathrm{p} \rightarrow \mathrm{p}$	$\vdash \mathsf{exists}_A : (A \rightarrow \mathrm{p}) \rightarrow \mathrm{p}$
$\vdash \mathsf{and} : \mathrm{p} \rightarrow \mathrm{p} \rightarrow \mathrm{p}$	$\vdash \mathsf{exist!}_A : (A \rightarrow \mathrm{p}) \rightarrow \mathrm{p}$
$\vdash \mathsf{or} : \mathrm{p} \rightarrow \mathrm{p} \rightarrow \mathrm{p}$	$\vdash \mathsf{equals}_A : A \rightarrow A \rightarrow \mathrm{p}$

For example, not is a propositional connective, which is of type $\mathrm{p} \rightarrow \mathrm{p}$.

For example, the sense of the relativizer *that*, or property conjunction, are defined as follows:

(26) $\mathsf{that}_A =_{\mathrm{def}} \lambda_{PQx}.(P\ x)\ \mathsf{and}\ (Q\ x)$

Here that takes three arguments: two properties, P and Q of type $\mathrm{A} \rightarrow \mathrm{p}$, and an entity, x of type A. Note the propositional connective and is used between $(P\ x)$ and $(Q\ x)$, instead of the true-value connective $\wedge$, because $(P\ x)$ is a proposition, not an extension. In the same way, the sense of determiners, such as *some*, can be defined accordingly:

(27) $\mathsf{some}_A =_{\mathrm{def}} \lambda_{PQ}.\mathsf{exist}(P\ \mathsf{that}\ Q)$

Here some_A is taking two A-type properties, P and Q, as its argument to return a proposition. $(P\ \mathsf{that}\ Q)$ is the conjunction of the properties P and Q. By the definition of that defined above, some_A means for all x of type A, exists $P\ x$ and $Q\ x$.

In our following analysis of Mandarin alternative questions, when we talk about the semantics/meaning of a linguistic expression, we refer to its sense, not its extension.

3.2 Linear Categorial Grammar

Our analysis of Chinese alternative questions is couched in Linear Categorial Grammar (LCG), a linear-logic based form of categorial grammar (Pollard, In Press; Mihaliček and Pollard, 2012; Pollard and Smith, 2012). Following Curry (1961) and Oehrle (1994), LCG systematically distinguishes *phenogrammatical structure* (*pheno* for short) from *tectogrammatical structure* (*tecto* for short). Pheno represents the surface form, while tecto

is semantically relevant combinatorics. The semantic component of LCG composes the meaning of the linguistic expressions. The meaning of a complex linguistic expression is recursively composed from the meanings of its immediate syntactic constituents.

LCG is a sequent-style natural deduction system whose sequents are of the form

(28) $\Gamma \vdash a : A; B; c : C,$

where the context Γ is a set of *hypothetical signs*, i.e. signs whose pheno and sense terms are variables, and the statement of the sequent is a sign, an ordered triple with three components: pheno, tecto and sense. Pheno terms are expressed in a higher-order logic, with basic type s (strings), constant e:s (null string) and $\cdot$: s $\rightarrow$ s $\rightarrow$ s (concatenation written infix). The tecto component is expressed in linear logic, with basic types commonly employed by other categorial grammars. Tecto types used in this paper are S (sentences), N (common nouns), NP (noun phrases), Q (questions). Q has three tecto subtypes: Q_{al} (alternative questions), Q_{wh} (constituent questions) and Q_{AnA} (A-not-A questions). Other tecto types are formed by the type constructor $\multimap$ (linear implication). The semantic component is a typed sense term in a higher-order logic with the basic types e (entities) and p (propositions). The remaining sense types are formed from these basic types by using the type constructor $\rightarrow$.

There are two kinds of axioms: logical, corresponding to hypotheses (analogous to traces in generative grammar), and nonlogical, correspond ing to lexical entries.

(29) **Logical Axioms (Hypothesize)**
$s : A; B; x : C \vdash s : A; B; x : C$

(30) **Nonlogical Axioms (Lexical Entries)**
$\vdash$ zhāngsān; NP; z
$\vdash$ chá ; NP; t
$\vdash \lambda_{st}.t \cdot$ xǐhuān $\cdot\, s$; NP $\multimap$ NP $\multimap$ S; like

For nonlogical axioms, the types are not included if they are easy to infer. For example, in (30), in the lexical entry for *xǐhuān* 'like', the pheno term is of type s $\rightarrow$ s $\rightarrow$ s. The tecto type NP $\multimap$ NP $\multimap$ S means *xǐhuān* takes two NPs to form a sentence. On the semantic side, *xǐhuān* takes two arguments of type e to form a proposition and of sense type e $\rightarrow$ e $\rightarrow$ p.

There are two inference rules: modus ponens (MP) and hypothetical proof (HP).

(31) **Modus Ponens (MP)**

$$\frac{\Gamma \vdash b : A \rightarrow B; C \multimap D; d : E \rightarrow F \qquad \Delta \vdash a : A; C; c : E}{\Gamma\Delta \vdash (b\ a) : B; D; (d\ c) : F}$$

(32) **Hypothetical Proof(HP)**

$$\frac{\Gamma,\ s : A;\ B;\ x : C \vdash a : D;\ E;\ b : F}{\Gamma \vdash \lambda_s.\ a : A \rightarrow D; B \multimap E; \lambda_x.\ b : C \rightarrow F}$$

In the tecto component, MP and HP are, respectively $\multimap$- elimination and $\multimap$- introduction; in pheno and sense components, they are, respectively, $\rightarrow$-elimination and $\rightarrow$-introduction at the level of types and combination and abstraction, respectively at the level of terms. (33) has the derivation as follows.

(33) zhāngsān xǐhūan chá.
zhangsan like tea
'Zhangsan likes tea.'

$$\dfrac{\vdash \text{zhāngsān};\ \text{NP};\ \mathsf{z} \qquad \dfrac{\vdash \lambda_{st}.t \cdot \text{xǐhuān} \cdot s;\ \text{NP}\multimap\text{NP}\multimap\text{S};\ \lambda_{xy}.\ \mathsf{like}\ x\ y \qquad \dfrac{\vdash \text{chá};\ \text{NP};\ \mathsf{tea}}{\vdash \lambda_t.t \cdot \text{xǐhuān} \cdot \text{chá};\ \text{NP}\multimap\text{S};\ \lambda_y.\ \mathsf{like\ tea}\ y}\text{MP}}{\vdash \text{zhāngsān} \cdot \text{xǐhuān} \cdot \text{chá};\ \text{S};\ \mathsf{like\ tea\ z}}\text{MP}$$

3.3 Questions in AHS

According to Hamblin (1957), interrogative sentences express propositional properties, and therefore they denote sets of propositions. In AHS, the type of A-**properties** is $A \to$ p. Therefore the type of propositional properties is abbreviated as follows:

$$\text{k} =_{\text{def}} \text{p} \to \text{p}$$

Let's take an English embedded question as an example. Since embedded and matrix questions are syntactically different in English, we use a tecto type $\bar{\text{Q}}$ for English embedded question and reserve Q for matrix questions.[4] We have *whether* defined in (34), which means *whether* takes a string on its right at the level of pheno; syntactically, it takes a sentence to form an embedded question, and it has the meaning whether which, defined in (35), takes a proposition p, which is expressed by a declarative sentence, and return a set of propositions: q equals p or q does not equal p.

(34) $\vdash \lambda s.\text{whether} \cdot s; \text{S} \multimap \bar{\text{Q}}; \mathsf{whether}$

(35) $\mathsf{whether} =_{\text{def}} \lambda_{pq}.(q\ \mathsf{equals}\ p)\ \mathsf{or}\ (q\ \mathsf{equals}\ (\mathsf{not}\ p))$

By the definitions above, the sense of *whether it rains* is represented as in (36):

(36) $\mathsf{whether\ rain} =_{\text{def}} \lambda_q.(q\ \mathsf{equals\ rain})\ \mathsf{or}\ (q\ \mathsf{equals}\ (\mathsf{not\ rain}))$

which denotes a set of propositions q, which denotes either *it rains* or *it does not rain.*

4 Analysis

4.1 The *háishì* construction

As discussed before, a *háishì* disjunction is formed with the lexical entry *háishì*. Syntactically, *háishì* conjoins two disjuncts [5], which could be of various syntactic categories, to form the *háishì* disjunction. These two disjuncts must be of the same syntactic and semantic type. If the disjunct is of the syntactic type A, then the third argument of *háishì*, called the *continuation* of the *háishì* disjunction, is of the type A $\multimap$ S. If the *haíshì* disjunction is embedded within some subordinate clause, there could be some ambiguity about what is the continuation of the *haíshì* disjunction, because it can be formed with respect to the subordinate clause or the matrix clause, as will be illustrated in detail later. Therefore,

[4] $\bar{\text{Q}}$ is not necessary in Mandarin Chinese because there is no difference between embedded and matrix questions.

[5] We confine our study to alternative questions of two disjuncts, but our analysis should be readily extended to the cases where more than two disjuncts are conjoined.

for a disjunct of tecto type A and the continuation of tecto type A $\multimap$ S, the tecto type of the lexical entry *háishì* will be A $\multimap$ A $\multimap$ (A $\multimap$ S) $\multimap$ Q_{al}. Correspondingly, the sense type of the lexicon *háishì* will be b $\to$ b $\to$ (b $\to$ p) $\to$ k, so that *háishì* takes two disjuncts of sense type b and the continuation of sense b $\to$ p and returns a question whose sense type is k.

Based on the discussion above, the lexical entry for *háishì* is:

(37) $\lambda_{stc}.c(s{\cdot}\text{háishì}{\cdot}t)$; A $\multimap$ A $\multimap$ (A $\multimap$ S) $\multimap$ Q_{al}; $\lambda_{xyPr}.(r \text{ equals } (P\ x))$ or $(r \text{ equals } (P\ y))$

In the pheno term of *háishì*, s and t are the pheno terms for the two disjuncts, which are of type s(tring)[6], then $s \cdot \text{háishì} \cdot t$ is of type s. c is the pheno term for the continuation which is of the pheno type s $\to$ s. The pheno term of the continuation will take $s \cdot \text{háishì} \cdot t$ as its argument and return a string. Therefore the lexicon *háishì* has pheno type s $\to$ s $\to$ (s $\to$ s) $\to$ s. A is the tecto type of the disjunct. In the semantic term of *háishì*, x and y are the semantic terms of the two disjuncts. P is the semantic term of the continuation. Therefore, $(P\ x)$ and $(P\ y)$ will be of type p, denoting the two alternative propositions that the alternative question presents. An alternative question denotes a set of propositions, whose only member, r, which is the true answer of the alternative question, is either $P\ x$ or $P\ y$.

If we define a constant haishi as in (38), we can simplify the semantic term of *háishì* as in (39).

(38) $\text{haishi} = \lambda_{pqr}.(r \text{ equals } p) \text{ or } (r \text{ equals } q) : \text{p} \to \text{p} \to \text{k}$

where p and q are the two alternative propositions presented by the *háishì* alternative question.

(39) $\lambda_{stc}.c(s \cdot \text{háishì} \cdot t)$; A $\multimap$ A $\multimap$ (A $\multimap$ S) $\multimap$ Q_{al}; $\lambda_{xyP}.\text{haishi}(P\ x)(P\ y)$

With the constant haishi defined in (38), the constant whether can be simplified as.

(40) $\text{whether} = \lambda_p.\text{haishi } p \text{ (not } p)$

With the lexicon entry of *haíshì* defined in (39), we can derive the meaning of an alternative question like (41).

(41) zhāngsān xǐhuān kāfēi háishì chá?
Zhangsan like coffee HAISHI tea
'Does Zhangsan like coffee ↑ or tea ↓?'

In (41), *háishì* first takes the two disjuncts: *kāfēi* 'coffee', and *chá*, 'tea', of the tecto type NP and the sense type e to form the *háishì* disjunction *kāfēi háishì chá*, which is of the tecto type (A $\multimap$ S) $\multimap$ Q_{al} and the sense type (e $\to$ p) $\to$ k. Then the *háishì* disjunction takes the continuation, *zhāngsān xǐhuān*, 'Zhangsan likes', which is of tecto type NP $\multimap$ S and sense type e $\to$ p, to return an alternative question of the techo type Q_{al} and the sense type k. The full-fledged derivation of (41) is provided in the following proof trees.

[6]In this paper, only the cases where the disjuncts of the string type are considered. The pheno term of *háishì* becomes complicated when disjuncts have functional types. It is beyond the scope of this paper to lay the groundwork for explaining this.

[1]

$$\dfrac{\dfrac{\vdash \text{zhāngsān};\ NP;\ \mathsf{z} \qquad \dfrac{\vdash \lambda_{st}.t \cdot \text{xǐhuān} \cdot s;\ \text{NP} \multimap \text{NP} \multimap \text{S};\ \lambda_{xy}.\ \mathsf{like}\ x\ y \qquad a; NP; b \vdash a; NP; b}{a; NP; b \vdash \lambda_t.t \cdot \text{xǐhuān} \cdot a;\ \text{NP} \multimap \text{S};\ \lambda_y.\ \mathsf{like}\ b\ y}\text{MP}}{a; NP; b \vdash \text{zhāngsān} \cdot \text{xǐhuān} \cdot a;\ \text{S};\ \mathsf{like}\ b\ \mathsf{z}}\text{MP}}{\vdash \lambda_a.\text{zhāngsān} \cdot \text{xǐhuān} \cdot a;\ \text{NP} \multimap \text{S};\ \lambda_b.\ \mathsf{like}\ b\ \mathsf{z}}\text{HP}$$

[2]

$$\dfrac{\dfrac{\dfrac{\vdash \text{kāfēi};\ \text{NP};\ \mathsf{coffee} \qquad \dfrac{\vdash \lambda_{stc}.c(s \cdot \text{háishì} \cdot t);\ \text{NP} \multimap \text{NP} \multimap (\text{NP} \multimap \text{S}) \multimap \text{Q}_{al};\ \lambda_{xyP}.\mathsf{haishi}(P\ x)(P\ y)}{\vdash \lambda_{tc}.c(\text{kāfēi} \cdot \text{háishì} \cdot t);\ \text{NP} \multimap (\text{NP} \multimap \text{S}) \multimap \text{Q}_{al};\ \lambda_{yP}.\mathsf{haishi}(P\ \mathsf{coffee})(P\ y)}\text{MP} \qquad \vdash \text{chá};\ \text{NP};\ \mathsf{tea}}{\vdash \lambda_c.c(\text{kāfēi} \cdot \text{háishì} \cdot \text{chá});\ (\text{NP} \multimap \text{S}) \multimap \text{Q}_{al};\ \lambda_P.\mathsf{haishi}(P\ \mathsf{coffee})(P\ \mathsf{tea})}\text{MP}}{\vdash \text{zhāngsān} \cdot \text{xǐhuān} \cdot \text{kāfēi} \cdot \text{háishì} \cdot \text{chá};\ \text{Q}_{al};\ \mathsf{haishi}(\mathsf{like\ coffee\ z})(\mathsf{like\ tea\ z})}\text{MP}}{}\ [1]$$

For the sake of space, the derivation of (41) has been broken up into two parts, [1] and [2]. In [1], we hypothesize the object of *xǐhuān* 'like' with a trace which later will be discharged by the *háishì* disjunction. [1] illustrates the derivation of the continuation of the alternative question. In [2], the *háishì* disjunction takes the continuation as an argument, resulting in the final derivation of (41). The derivation shows that (41) has the surface form shown in the pheno component, syntactic category Q_{al} (question), and the meaning: haishi (like coffee z) (like tea z). By the definition of haishi, (41) means:

(42) haishi (like coffee z) (like tea z) = λ_r.(*r* equals (like coffee z)) or (*r* equals (like tea z))

(42) denotes a set of propositions, whose only member *r*, the true answer of this alternative question is either like coffee z or like tea z.

4.2 *háishì* constructions vs. A-not-A questions

In this section, we are going to show how the issue of island-(in)sensitivity of alternative questions and A-not-A questions in Mandarin Chinese put forward by Huang (1991) can be satisfactorily addressed within the LCG framework.

According to Huang et al. (2009), A-not-A questions are a special type of alternative question in Mandarin Chinese in addition to polar questions and constituent questions,

because an A-not-A question requests the addressee to choose between a positive and a negative alternative provided in the question. Following that, we call this type of question A-not-A question rather than polar question in this study. A-not-A questions in Mandarin Chinese are formed by A-not-A verbs, where A is either a monosyllabic verb or the first syllable of a multisyllabic verb. For example, *qù* 'go' is a monosyllabic verb in Mandarin Chinese, *qù-bú-qù* 'go-not-go' is an A-not-A verb, which yields an A-not-A question.

(43) nǐ qù-bú-qù měigúo?
you go-not-go America
'Are you going to America?'

(44) nǐ qù háishì bú qù měigúo?
you go HAISHI not go America
'Are you going to America ↑ or not ↓?

For (43), the affirmative answer is *qù*, 'go', and the negative answer is *bú qù*, 'not go'. Semantically, A-not-A questions are interchangeable with *háishì* alternative questions in some situations. For example, we can express (43) with an alternative question as shown in (44).

However, Huang (1991) shows that they are different in terms of island sensitivity. (9) and (10) are repeated here as (45) and (46).

(45) a. wǒ qù měigúo háishì bú qù měigúo bǐjìaohǎo?
I go America HAISHI not go America better
'Is it better for me to go to America ↑ or not ↓?'

b. *wǒ qù-bú-qù měigúo bǐjìaohǎo?
I go-not-go America better?
Intended meaning: 'Is it better for me to go to America ↑ or not ↓?'

(46) a. nǐ xǐhūan rènshí nǐ háishì bú rènshí nǐ de rén?
you like know you HAISHI not know you DE people
'What people do you like better? People who know you or people who do not know you?'

b. *nǐ xǐhūan rèn-bú-rènshí nǐ de rén?
you like know-not-know you DE people
Intended meaning: 'What people do you like better? People who know you or people who do not know you?'

(45) is an example of the sentential subject constraint and (46) the complex NP constraint. In (45), the *háishì* construction occurs within a subordinate subject clause and in (46), it occurs within a relative clause. However, replacing the *háishì* construction with an A-not-A verb in (45b) and (46b) make the two sentences ungrammatical. Huang (1991) thus concludes that the *háishì* construction and A-not-A questions are different in terms of their sensitivity to those island constraints. If we change the main verb in (45), however, we can see that an A-not-A question like (47) becomes grammatical.

(47) wǒ qù-bú-qù měigúo hái méi dìng.
I go-not-go America yet not decide
'Whether I am going to America is not decided yet.'

(47) shows that when we change the main verb from *bǐjìaohǎo* 'better' to *hái méi dìng* 'not decided yet', the A-not-A question is not subject to the sentential subject constraint. This suggests that the ungrammaticality of (45b) results from different selectional restrictions of the subject that the verbs have. It seems *hái méi dìng* can take an A-not-A

question as its subject while *bǐjìaohǎo* cannot. A careful reader might ask why *bǐjìaohǎo* 'better' can take an alternative question as its subject but not an A-not-A question. The answer is that the *háishì* disjunction takes the continuation as its argument to form an alternative question, as defined in the lexical entry for *háishì*. More specifically in (45a), it is not that the continuation *wo ... bǐjìaohǎo* takes the *háishì* construction as its subject, but rather that the *háishì* disjunction takes the continuation *wo ... bǐjìaohǎo* as its argument. Unlike a *háishì* disjunction, an A-not-A verb takes the same kind of argument as the A-verb, and does not scope over its continuation. This is the crucial difference between the *haíshì* disjunctions and and A-not-A verbs in Mandarin Chinese, which accounts for the data in (45) and (46).

In order to clarify this, we need to give a lexical entry for an A-not-A verb. An A-not-A verb requires the same number and type of arguments as does the verb A or the verb which has A as its initial syllable. The difference is that an A-not-A verb yields an A-not-A question. Based on this observation, a lexical entry for an A-not-A verb such as *qù-bú-qù*, 'go-not-go', is defined as follows. Q_{AnA} is the tecto type for A-not-A questions.

(48) **qù-bú-qù** $=\vdash \lambda_{st}.t \cdot \text{qu-bu-qu} \cdot s;\ \mathrm{NP} \multimap \mathrm{NP} \multimap \mathrm{Q}_{AnA};\ \lambda_{xy}.\mathsf{whether}\ (\mathsf{go\text{-}to}\ x\ y)$

qù 'go' in Mandarin Chinese can be either a transitive or an intransitive verb. Here the lexical entry is for the transitive verb *qù* 'go'. An A-not-A question as in (49) can be derived as follows:

(49) zhāngsān qù-bú-qù měigúo?
Zhangsan go-not-go America
'Is Zhangsan going to America?'

$$\dfrac{\dfrac{\vdash \text{zhāngsān};\quad \mathrm{NP};\quad \mathsf{z}}{\vdash \lambda_t.t \cdot \text{qu-bu-qu} \cdot \text{měgúo};\quad \mathrm{NP} \multimap \mathrm{Q}_{AnA};\quad \lambda_y.\mathsf{whether}\ (\mathsf{go\text{-}to}\ \mathsf{america}\ y)} \qquad \dfrac{\vdash \lambda_{st}.t \cdot \text{qù-bú-qù} \cdot s;\quad \mathrm{NP} \multimap \mathrm{NP} \multimap \mathrm{Q}_{AnA};\quad \lambda_{xy}.\mathsf{whether}\ (\mathsf{go\text{-}to}\ xy) \qquad \vdash \text{měigúo};\quad \mathrm{NP};\quad \mathsf{america}}{\vdash \lambda_t.t \cdot \text{qu-bu-qu} \cdot \text{měgúo};\quad \mathrm{NP} \multimap \mathrm{Q}_{AnA};\quad \lambda_y.\mathsf{whether}\ (\mathsf{go\text{-}to}\ \mathsf{america}\ y)}}{\vdash \text{zhāngsān} \cdot \text{qù-bú-qù} \cdot \text{měigúo};\quad \mathrm{Q}_{AnA};\quad \mathsf{whether}\ (\mathsf{go\text{-}to}\ \mathsf{america}\ \mathsf{z})}$$

háishì takes two disjuncts and the continuation as arguments and returns an alternative question, while the A-not-A verb takes the arguments required by the A-verb and return an A-not-A question. Comparing lexical entries for *háishì* and the A-not-A verb, we can see that the *háishì* construction scopes over the continuation but the A-not-A verb does not. Therefore *háishì* disjunctions can take a wider scope than A-not-A verbs unless the continuation is null. This difference cannot be seen when they are not embedded, as illustrated by (43) and (44). But when they occur in subordinate clauses, this difference accounts for the acceptability of (45a) and (46a) and the unacceptability of (45b) and (46b).

When a *háishì* disjunction occurs in an embedded clause, it can scope over either the embedded clause or over the matrix sentence. If a *háishì* disjunction scopes over the embedded clause, the resulting sentence is a declarative sentence with an embedded alternative question. If the *háishì* disjunction scopes over the matrix sentence, the resulting interpretation will be an alternative question. Depending on whether the matrix verb can take an interrogative complement, the interpretation of the whole sentence could be either an alternative question or a declarative sentence with an embedded alternative question. However, when an A-not-A verb is embedded in a subordinate clause, the only possible interpretation is a declarative sentence with an embedded A-not-A question. When the

matrix verb cannot take an interrogative complement, the only reading is blocked and the sentence becomes unacceptable.

These predictions are exemplified by the following data.

(50) a. Zhāngsān zhīdào wǒ qù měigúo háishì yīnggúo?
zhangsan know I go America HAISHI England
'Does Zhangsan know I'm going to America↑ or England ↓?'

b. Zhāngsān zhīdào wǒ qù měigúo háishì yīnggúo.
zhangsan know I go-not-go America HAISHI England
'Zhangsan knows whether I am going to America or England.'

(51) a. *zhāngsān zhīdào wǒ qù-bú-qù měigúo?
Zhangsan know I go-not-to the-USA
Intended: 'Does Zhangsan know whether I am going to the USA?'

b. zhāngsān zhīdào wǒ qù-bú-qù měigúo.
Zhangsan know I go-not-to the-USA
'Zhangsan knows whether I am going to the USA.'

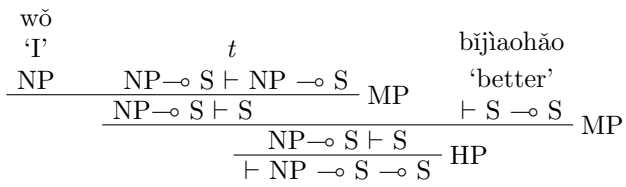
Since *zhīdào* 'know' can take either a declarative sentence or a question as a complement. Correspondingly, for a *haíshì* disjunction embedded in the complement of 'know', there are two readings: an alternative question reading (50a) and a declarative sentence reading with an embedded alternative question (50b). Since an A-not-A phrase cannot scope over the matrix sentence, the possibility that the root sentence is a question is ruled out. Only the reading of a declarative sentence with an embedded question survives, as shown in (51b). Along the same line of argument, we predict that verbs like *xiāngxìn* 'believe', which can only take declarative sentences as complements, will filter out the reading where the embedded clause is a question. For *háishì*, only the reading where the root sentence is a question is available. By contrast, for an A-not-A phrase, no reading is available because the possibility to interpret it as an embedded question is ruled out by the matrix verb. These predictions are exemplified by the following.

(52) a. zhāngsān xiāngxìn wǒ qù měigúo háishì yīnggúo?
Zhangsan believe I go America HAISHI England
'Does Zhangsan believe I will go to America ↑ or England ↓?'

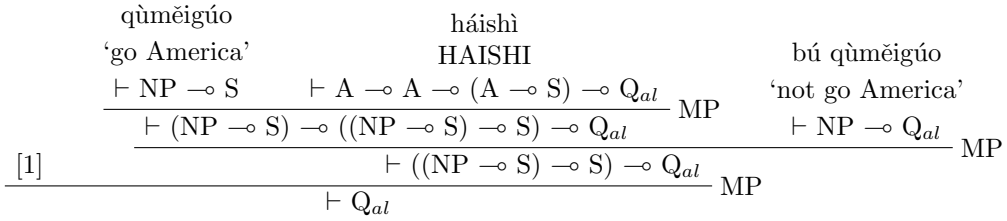
b. *zhāngsān xiāngxìn wǒ qù-bu-qù měigúo.
Zhangsan believe I go-not-go the-USA
Uninterpretable

Based on the above discussion, we can account for the unacceptability of (45b) accordingly. The intransitive verb *bǐjìaohǎo* 'is-better', like *xiāngxìn*, 'believe', cannot take an interrogative clause as its sentential argument. Therefore, in (45a), the only way to derive this sentence is for the *háishì* phrase to scope over the matrix sentence, and the whole sentence has to be interpreted as an alternative question. A simplified derivation showing only the tecto components is given below to show that (45a) is an acceptable sentence in Mandarin Chinese.

[1]

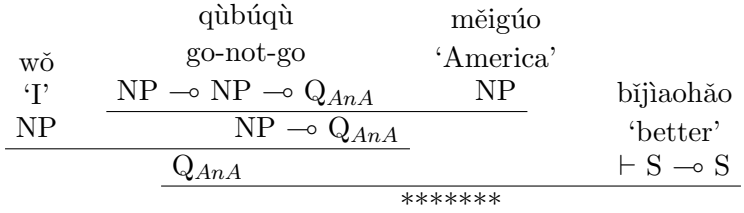


[2]



For the sake of space, the simplified derivation of (45a) is split into two parts. In [1], the continuation *wǒ...bǐjìaohǎo* is derived. In [2], the *háishì* disjunction is derived and then it takes the continuation derived in [1] to form an alternative question in the end.

However, in (45b), the possibility for A-not-A phrase to be interpreted as an embedded polar question is ruled out by the fact that the matrix verb, *bǐjìaohǎo*, only selects a propositional argument.



As shown above, the derivation of (45b) fails because *bǐjìaohǎo* is looking for a constituent of tecto type S as its complement, while the only complement available *wǒ qùbúqù měigúo* is an A-not-A question.

In this section, we accounted for the difference between Mandarin alternative questions and A-not-A questions within the LCG framework. Our analysis revealed that a *háishì* disjunction takes its continuation as argument while a A-not-A verb does not. Without resorting to notions like 'movement' or 'island constraint', our analysis successfully accounts for the data which used to be explained by 'island-sensitivity'.

4.3 *háishì* constructions vs. *wh* phrases

Both Huang (1982) and Erlewine (2012) treat the *haíshì* disjunction as *wh*-phrase occurring in-situ. Erlewine (2012) argues that *háishì* triggers an alternative set whose members are the alternatives or disjuncts introduced by *háishì*. Constituent question words such as, *who*, or *what*, are also widely considered to trigger alternative sets.

(53) zhāngsān xǐhuān shénme?
Zhangsan like what
'What does Zhangsan like?'

(54) Zhangsan xǐhuān chá háishì kāfēi?
Zhangsan like tea HAISHI coffee
'Does Zhangsan like tea↑ or coffee ↓?'

If the domain of the *wh*-word *shenme* 'what' in (53), is a set of entities, {tea, coffee}, then these two questions,(53) and (54), by Hamblin alternative semantics, denote the same set of propositions, {Zhangsan likes tea, Zhangsan likes coffee }. This parallel can also be observed when these two types of questions are embedded. Notice here that the scope ambiguity does not arise here for either of them, since *xǐangzhīdào* 'wonder', can only take a question as its complement.

(55) a. lǐsì xǐangzhīdào zhāngsān xǐhuān shénme.
Lisi wonder Zhangsan like what
'Lisi wonders what Zhangsan likes.'

b. lǐsì xǐangzhīdào Zhāngsān xǐhuān chá háishì kāfēi.
Lisi wonder Zhangsan like tea HAISHI coffee
'Lisi wonders whether Zhangsan likes tea or coffee.'

However, while it is possible to have multiple *wh*-phrases within one question, it is impossible to have a *wh* phrase and *háishì* disjunction occur in the same question, as shown in (7) and (8), repeated here as (56 and (57).

(56) shúi xǐhuān shénme.
who like what
'Who likes what?'

(57) *shúi xǐhuān chá háishì kāfēi.
who like tea HAISHI coffee
Uninterpretable

The analysis which proposes that *háishì* disjunction be treated as a *wh*-phrase cannot account for the ungrammaticality of (57). However, the ungrammaticality is predicted by the lexical entry of *háishì* defined in (39). Before we show how it works, we need to first illustrate how multiple constituent questions, i.e., constituent questions with multiple *wh*-words, are derived in LCG framework.

According to Mihaliček and Pollard (2012), when multiple *wh* phrases occur in one sentence, one of the them needs to scope over a continuation of type e $\rightarrow$ p to return a question of type k, and others have to scope over a continuation of type e $\rightarrow$ k. Based on this distinction, for each *wh*-word, there are two lexical entries. Following Mihaliček and Pollard (2012), I have the following two lexical entries for *shéi* 'who'.

(58) a. shéi $\vdash \lambda_f.f(\text{shéi}); (\text{NP} \multimap \text{S}) \multimap \text{Q}_{wh}; \lambda_{Qp}.\textsf{some person}\ (\lambda_x.p\ \textsf{equals}\ (Q\ x))$
b. shéi $\vdash \lambda_f.f(\text{shéi}); (\text{NP} \multimap \text{Q}_{wh}) \multimap \text{Q}_{wh}; \lambda_{Kp}.\textsf{some person}(\lambda_x.K\ x\ p)$

(58a) is used to derive a constituent question like, *shéi xǐhuān chá?* 'who likes tea?', in which *shéi* will take *xǐhuān chá* (e $\rightarrow$ p) 'likes tea' as an argument to return a constituent question. In a multiple constituent question like *shéi xǐhuān shéi?* 'who likes who?', one of the two *shéi*s has to be defined as in (58b), which takes as argument either *shéi xǐhuān* (e $\rightarrow$ k), 'who likes', or *xǐhuān shéi* (e $\rightarrow$ k) 'like who', to return a multiple constituent question. The lexical entry of *wh*-word defined in (58b) shows that (58b) is not used unless there is another *wh*-word is present.

Unlike *wh*-phrases, the *háishì* disjunction cannot scope over questions, as indicated in its lexical entry (39). This rules out the possibility to interpret (57) as an alternative question. Since the *wh*-word can only scope over constituent questions, as indicated in (57), the possibility for it to be a constituent question is also ruled out. Simplified derivations are given below to show no matter what lexical entry *shèi* takes, (57) is not grammatical.

(59) *shéi* takes the lexical entry in (58a).

shúi xǐhuān;	chá háishì kāfēi
'who like'	'tea HAISHI coffee'
$\vdash$NP $\multimap$ Q_{wh};	$\vdash$ (NP$\multimap$ S) $\multimap$ Q_{al};
* * * * * * *	

(60) *shúi* takes the lexical entry in (58b).

shúi;	xǐhuān chá háishì kāfēi
'who'	' like tea HAISHI coffee'
$\vdash$ (NP $\multimap$ Q_{wh}) $\multimap$ Q_{wh};	$\vdash$ NP $\multimap$ Q_{al};
* * * * * * *	

However, with the same lexical entries, we can account for the acceptability of (22), which Erlewine (2012) mistakenly predicts to be unacceptable. (22) is repeated as (61).

(61) nǐ xiǎng-zhīdào shúi cóng běijīng lài háishì cóng shánghǎi lái?
you want-to-know who from Beijing come HAISHI from Shanghai come
'Do you want to know who is from Beijing or who is from Shanghai?'

A simplified derivation which only consists of the tecto part is given below to show why (22) is acceptable by the lexical entries proposed. Since pheno and semantic parts are ignored here, the *háishì* disjunction is labelled as 'háishì disjunction' with tecto type '$\vdash$ ((NP $\multimap$ S) $\multimap$ S) $\multimap$ Q_{al}' in the derivation to make the derivation easier to read.

nǐ	xiǎng-zhīdào	shéi;				
'you'	'wonder'	'who'		t		
$\vdash$ NP	$\vdash$ NP $\multimap$ Q $\multimap$ S	$\vdash$ (NP $\multimap$ S) $\multimap$ Q_{wh}		NP $\multimap$ S	$\vdash$ NP $\multimap$ S	
			MP			MP
	$\vdash$ Q $\multimap$ S			NP $\multimap$ S $\vdash$ Q_{wh}		
					MP	
		NP $\multimap$ S $\vdash$ S				háishì disjunction
		$\vdash$ (NP $\multimap$ S) $\multimap$ S	HP			$\vdash$ ((NP $\multimap$ S) $\multimap$ S) $\multimap$ Q_{al}
						MP
				$\vdash$ Q_{al}		

In the derivation above, *xiǎng-zhīdào* 'wonder' has tecto type: $\vdash$ NP $\multimap$ Q $\multimap$ S, which means *xiǎng-zhīdào* takes an NP, the subject, and Q, its interrogative complement, to form a sentence. Here Q is an umbrella type for all question types, which means its complement can be a constituent question, an A-not-A question or an alternative question. The disjuncts *cóng běijīng lái* 'come from Beijing' and *cóng shánghǎi lái* 'come from Shanghai' are of the tecto type NP $\multimap$ S. Then the *háishì* disjunction is of the tecto type ((NP $\multimap$ S) $\multimap$ S)$\multimap$ Q_{al} and the continuation is of the tecto type (NP $\multimap$ S) $\multimap$ S. The derivation above shows that (61) is acceptable in Mandarin Chinese and it can be interpreted as an alternative question.

With the same lexical entries, we can also predict the ungrammaticality of sentences (62) and (63) in Mandarin Chinese.

(62) *zhāngsān xǐ-bù-xǐhuān chá háishì kāfēi?
Zhangsan like-not-like tea HAISHI coffee
Uninterpretable

zhāngsān xǐ-bù-xǐhuān;	chá háishì kāfēi
'Zhangsan like-not-like'	'tea HAISHI coffee'
$\vdash$NP $\multimap$ Q_{AnA};	$\vdash$(NP$\multimap$ S) $\multimap$ Q_{al};

* * * * * * *

This sentence is ungrammatical because *xǐbùxǐhuān*, 'like-not-like', forms an A-not-A question in Chinese, and the *háishì* construction cannot scope over it to return an alternative question. The following example is meant to show that A-not-A questions in Mandarin Chinese, like alternative questions, cannot scope over or cannot be scoped over by *wh*-question phrases. With the lexical entries defined in (58), we can account for the ungrammaticality of (63).

(63) *zhāngsān xǐbùxǐhuān shénme?
Zhangsan like-not-like what
Uninterpretable

zhāngsān xǐ-bù-xǐhuān;	shénme
'Zhangsan like-not-like'	'what'
$\vdash$NP $\multimap$ Q_{AnA};	$\vdash$(NP$\multimap$ S) $\multimap$ Q_{wh};

* * * * * * *

zhāngsān xǐ-bù-xǐhuān;	shénme
'Zhangsan like-not-like'	'what'
$\vdash$NP $\multimap$ Q_{AnA};	$\vdash$ (NP$\multimap$ Q_{wh}) $\multimap$ Q_{wh};

* * * * * * *

These facts confirm that our observation that *wh*-questions in Mandarin have different syntactic properties from alternative questions is empirically well-grounded.

5 Conclusion

This study proposes an analysis of Mandarin alternative questions within the LCG and AHS framework, which derives the Mandarin alternative questions in a completely compositional way. The surface string, syntactic type and semantics of Mandarin alternative questions can be derived in parallel based on the lexical entry proposed for *háishì*.

The main departure of the current analysis from those previous accounts is that this analysis is couched within a categorial grammar framework which is completely lexicalized. The derivation of the meaning of the alternative questions and other related questions in Mandarin is fully compositional given the inference rules.

This study reveals that an A-not-A phrase cannot scope over the continuation as the *háishì* disjunction does, which explains why, when embedded, an A-not-A question can only have the interpretation of being an embedded question within a declarative sentence if the matrix verb can take an interrogative complement. However, for an embedded alternative question, the *háishì* disjunction, as it was made clear in the lexical entry for *háishì*, can scope over the continuation, which could be either at level of the embedded clause or the matrix sentence. The resulting sentence could be either an alternative question or an indicative sentence with an embedded alternative question, depending on whether the main verb takes an interrogative or indicative complement.

With the same lexical entry for *haíshì*, we account for how a *háishì* disjunction differs from a *wh* phrase and correctly predict the grammaticality of the sentences like (22) and the ungrammaticality of the sentences like (8), which a movement-based analysis fails to do.

References

Beck, Sigrid. 2006. Intervention effects follow from focus interpretation. *Natural Language Semantics* 14:1–56.

Beck, Sigrid and Shin-Sook Kim. 2006. Intervention effects in alternative questions. *Journal of Comparative German Linguistics* 9:165–208.

Curry, Haskell. 1961. Some logical aspects of grammatical structure. In J. R, ed., *Structure of Language and its Mathematical Aspects*, 56 – 68. American Mathematical Society.

Erlewine, Michael Yoshitaka. 2012. Alternative questions through focus alternative in Mandarin Chinese. In *Proceedings of Chicago Linguistics Society 48*, 221–234.

Hamblin, C.L. 1957. *Language and the theory of information: Logic and Scientific Method Programme*. Ph.D. thesis, University of London.

Heidenreich, Samantha, Shari Speer, and Carl Pollard. 2014. Do the pitch accents or the phrasal accents determine an alternative question. In *the 27th Annual Cuny Conference on Human Sentence Processing*, 90.

Huang, Cheng-Teh James. 1982. *Logical relations in Chinese and The Theory of Grammar*. Ph.D. thesis, Massachusetts Institute of Techonology.

Huang, Cheng-Teh James. 1991. Modularity and Chinese A-not-A questions. In *Interdisciplinary approaches to linguistics: essays in honor of Yuki Kuroda.*. Kluwer Academic Publishers.

Huang, Cheng-Teh James, Yen hui Audrey Li, and Yafei Li. 2009. *The Syntax of Chinese*. Cambridge University Press.

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*, 130–145. LNCS 7395. Berlin: Springer-Verlag.

Oehrle, Richard. 1994. Term-labelled categorial type systems. *Linguistics and Philosophy* 17:633–678.

Plummer, Andrew and Carl Pollard. 2012. Agnostic possible worlds semantics. In *Logical Aspects of Computational Linguistics, Seventh International Conference*, 201–212. Nantes: Springer Lecture Notes in Computer Science 7351.

Pollard, Carl. In Press. Agnostic hyperintensional semantics. *a special issue of Synthese on hyperintentionality* 1–28.

Pollard, Carl and E.Allyn Smith. 2012. A unified analysis of the same, phrasal comparatives, and superlatives. In *Proceedings of SALT 22*, 307–325.

Pruitt, Kathryn and Floris Roelofsen. 2013. Interpretation of prosody in disjunctive questions. *Linguistic Inquiry* .

Rooth, Mats. 1985. *Association with Focus*. Ph.D. thesis, University of Massachusetts, Amherst.

Shan, C. 2004. Binding alongside hamblin alternative calls for variable-free semantics. In *Proceedings of SALT 14*, 289–304.

Syntactic Features for Regular Constraints and an Approximation of Directional Slashes in Abstract Categorial Grammars

Makoto Kanazawa*
National Institute of Informatics, Tokyo, Japan, and
SOKENDAI (Graduate University for Advanced Studies)

Abstract

I demonstrate the usefulness of syntactic features expressing regular constraints in a linguistic theory based on the formalism of abstract categorial grammars, using a fragment involving *wh*-extraction islands as an example. I then use the same technique to give a method of approximating in ACGs the behavior of the directional slashes of the Lambek calculus.

Keywords: abstract categorial grammar; Lambek calculus; intersection with regular sets; syntactic features

1 Introduction

This paper was motivated by the question of to what extent the formalism of *abstract categorial grammars* (de Groote, 2001) is capable of replicating Lambek grammars, where both formalisms are taken as systems for defining relations between strings and typed λ-terms (understood as representations of meanings). This is a purely technical question about the expressive power of two grammar formalisms, but I believe its resolution to be of interest to linguists since the apparent difficulty of the ACG formalism to simulate Lambek calculus analyses of non-constituent coordination and related phenomena seems to have led some people to conclude that the ACG is fundamentally inadequate as a vehicle to express a satisfactory theory of natural language syntax and compositional semantics.[1]

As a partial answer to this question, this paper presents an automatic procedure to translate an arbitrary Lambek grammar into an ACG that simulates it up to a certain point. This is done by adding two syntactic features to each atomic type, which express

*I am grateful to Chris Tancredi and Greg Kobele for giving me native speakers' judgments.

[1] This view was expressed by Kubota and Levine (2014b, p. 30):

> "...keeping track of the right word order becomes a virtually intractable problem in non-directional variants of CG such as Abstract Categorial Grammar (de Groote 2001) and Lambda Grammar (Muskens 2003)"

and by Moot (2014, p. 2):

> "The abstract categorial grammar treatment suffers from problems of overgeneration and problems at the syntax-semantics interface unlike any other categorial grammar."

regular constraints on the surface positions of "extraction sites". The way the syntactic features are percolated is automatically determined by the regular constraints; this is an application of a general method I developed in Kanazawa (2006). Since each feature has a finite number of possible values, employing it is formally equivalent to splitting each atomic type into a finite number of distinct variants.

Before turning to the simulation of Lambek grammars, I illustrate my general method by showing how λ-abstraction in derivations can be controlled through a finite-valued feature, using a fragment containing *wh*-extraction. (The fragment is vaguely reminiscent of GPSG (Gazdar et al., 1985).) The feature employed in the fragment is similar to what Pogodalla and Pompigne (2012) arrived at by an ad hoc construction. I then proceed to show how one can use the same method to limit λ-abstraction in such a way that the surface position of the extraction site created by λ-abstraction is limited to left or right periphery, mimicking the behavior of the introduction rules for Lambek's directional slashes.

A systematic use of the latter technique results in a procedure to translate an arbitrary Lambek grammar into an ACG. This is not a perfect simulation; the output ACG undergenerates relative to the string-meaning relation of the input Lambek grammar.[2] For the purpose of the description of natural language, this imperfection does not necessarily bode ill for the proposed technique. The discrepancy does not seem to show up in concrete cases that have been discussed by linguists, and I will suggest that there may even be reason to favor the feature mechanism over the directional slashes of the Lambek calculus.

I will start by giving an informal introduction to the ACG formalism, illustrating some of its important properties through concrete examples. Before proceeding to do so in Section 2, however, I would like to make some terminological and conceptual clarifications.

By a *grammar formalism*, I mean a mathematically defined class of finite devices (called *grammars*) equipped with a definition of the *language* of each grammar, which is a set of discrete entities of some sort (strings, trees, λ-terms, pairs of strings and λ-terms, etc.). A grammar formalism is an abstract mathematical object that can and should be studied in isolation from any applications. Each grammar formalism usually distinguishes itself from others by referring to a grammar in its class by a unique compound noun ending in the word "grammar". Thus, a *context-free grammar* or a *CFG* is a grammar belonging to a certain formalism found in any textbook in formal language theory; an *abstract categorial grammar* or an *ACG* is a grammar belonging to the formalism defined by de Groote (2001).[3]

In linguistics, the word "grammar" has another, entirely different use, where it appears as part of a proper name that refers to a particular (usually fairly comprehensive) *linguistic theory* that aims to capture the syntax (and often compositional semantics) of natural language. These names are almost always capitalized. Examples are Lexical-Functional Grammar (LFG), Generalized Phrase Structure Grammar (GPSG), and Head-Driven Phrase Structure Grammar (HPSG). These linguistic theories are often expressed by means of some (usually custom-built) grammar formalism in the sense of the preceding paragraph, but more often than not, the definition of the grammar formalism is

[2] In the initial abstract I submitted to this workshop, I stated that the output ACG also overgenerates relative to the input Lambek grammar. That statement was made in error, and I now think it only undergenerates.

[3] What I call an ACG in this paper is technically a coupling of two ACGs that share the same *abstract signature*. This use of ACGs was called the *transductive paradigm* by de Groote (2001). Other authors have proposed definitions roughly equivalent to de Groote's, most notably Muskens (2003), but not all details are the same.

only implicit in the linguistic theory.[4] (To add to the confusion, these linguistic theories themselves are sometimes called "grammar formalisms" when they serve as frameworks in which to carry out further linguistic investigations.)

The formalism of abstract categorial grammars was introduced without a well-developed linguistic theory to go with it. I think it is fair to say that subsequent uses of the ACG formalism in theoretical and computational linguistics have been sporadic and no broad consensus has emerged as to what form linguistically adequate grammars of English, French, etc., should take within a framework offered by the ACG formalism.[5] In this sense, there is no linguistic theory called *Abstract Categorial Grammar*, at least not yet.[6]

There is another point I wish to make at this point about the relation between a grammar formalism and a linguistic theory. When a grammar formalism is used in a linguistic theory, it need not be the case that the formalism provides the entire metalanguage in which to express individual grammars within that theory. A trivial example is the use of parentheses and braces in phrase structure rules to indicate optional elements and alternatives. These are conventions used to abbreviate multiple phrase structure rules into a single rule and do not appear in the official definition of a phrase structure rule. Nevertheless, as noted by Chomsky (1965, pp. 42–43), these abbreviatory devices may play important roles in expressing linguistically significant generalizations. Another example might be Jacobson's (2014, pp. 94–95) use of "directional rules" in the lexicon of a categorial grammar of English to express the generalization that English is a predominantly *head-first* language. Here, a directional rule is a kind of lexical rule that turns a lexical entry underspecified as to the directions of some slashes into one where the directions of those slashes are specified. Another, particularly elaborate example of an additional mechanism that falls outside of the grammar formalism adopted by a linguistic theory is found in the TAG-based theory of Frank (2002), where the Merge and Move operations are used to generate the finite set of elementary trees of a tree-adjoining grammar.

Now when the workings of such formalism-external devices are specified fully precisely, they may be incorporated into the grammar formalism and the resulting enriched formalism may be studied as a mathematical object. But they need not be. Rather, these devices may simply be viewed as part of a system of notation—compression scheme if you will—that enables compact representations of grammars belonging to the grammar formalism. The choice between these two viewpoints of course does not affect the content of the linguistic theory in any way; it is just a matter of what level of abstraction is the most fruitful in the mathematical study of grammar formalisms. When certain representational devices in the linguistic theory are left out of the grammar formalism, algorithmic

[4] An unfortunate practice that greatly hampers understanding of the relevant linguistic theory, in my opinion. A notable exception in this respect is Tree-Adjoining Grammar (Joshi and Schabes, 1997; Abeillé and Rambow, 2000; Frank, 2002).

[5] Needless to say, examples ("toy grammars") that are used to illustrate properties of a grammar formalism need not constitute any serious attempt to develop a linguistic theory based on that formalism. Also, when a linguist employs a certain formalism in her account of a certain empirical phenomenon, the grammar fragment provided for that purpose may involve choices that are tangential to the points she wants to make and which she would be happy to change in the face of possible criticisms.

[6] In fact, de Groote and Pogodalla (2004, p. 421) go so far as to say

> "Abstract Categorial Grammars are not intended to be yet another grammatical formalism that would compete with other well-established formalisms. They should rather be seen as the kernel of a grammatical framework — in the spirit of (Ranta, 2004) — in which other existing grammatical models may be encoded."

I myself do not share this view. I think the ACG formalism has some unique strengths that make it attractive as a grammar formalism for natural language.

properties of the formalism will guide and inform our understanding of the corresponding properties of the linguistic theory, rather than reveal them completely. This will in no way render the formalism irrelevant to the linguistic theory; on the contrary, such a separation between a grammar formalism and formalism-external representational devices may offer a more effective way to gain important insights into human grammars.

2 Abstract Categorial Grammars

The ACG formalism is based on the *simply typed λ-calculus*, the kind of λ-calculus linguists are familiar with from Montague semantics. *Simple types* are formed from atomic types with the connective →; if A and B are simple types, then $A \to B$ is a simple type. A *λ-term* is either a variable, a constant, an application MN of λ-terms M and N, or a λ-abstract $\lambda x.M$, where M is a λ-term and x is a variable. Note that I write MN instead of $M(N)$ for applications; the former notation is the standard one in λ-calculus. (We of course use parentheses where they are necessary to remove ambiguity.) Application is assumed to be left-associative, so MNP means $(MN)P$, not $M(NP)$. A sequence of λs is abbreviated to one, as in $\lambda xyz.x(yz) = \lambda x.\lambda y.\lambda z.x(yz)$. Another standard abbreviation is employed in writing types: $\alpha \to \beta \to \gamma$ abbreviates $\alpha \to (\beta \to \gamma)$ (i.e., the connective → associates to the right). I write $\alpha^n \to \beta$ for $\alpha \to \cdots \to \alpha \to \beta$, with n occurrences of α. I assume that the reader is familiar with the notions of *free* and *bound* variables, with λ-conversion or *β-reduction*, and with how types are assigned to λ-terms given the types of variables and constants.

Roughly, an *abstract categorial grammar* (ACG)[7] consists of a finite set of triples of the form (α, M, N), where α is a simple type over some finite set $\mathcal{P}$ of atomic types, and M and N are closed λ-terms of types $\sigma(\alpha)$ and $\tau(\alpha)$, respectively, where σ and τ are the two *type substitutions* specified by the grammar. In this paper, I call the triples (α, M, N) comprising an ACG *grammar entries*. The first component α of a grammar entry is a *syntactic type*. The λ-terms M and N are the λ-terms in the *form dimension* and the *meaning dimension*, respectively. The former is usually assumed to be a *linear* λ-term, while the latter is not restricted in that way.[8] These λ-terms may contain constants as well as bound variables and λ. In this paper, the constants appearing in the form dimension are all of the same type *str*, which is the type of string, except the constant ∘, which is of type $str \to str \to str$ and stands for string concatenation. We write ∘ as an infix operator. The constants appearing in the meaning dimension have various types built from atomic types (e, t, etc.).

An ACG generates a pair consisting of a string and a λ-term by combining grammar entries through application and linear λ-abstraction in accordance with their syntactic types. A succinct formal definition of ACGs is found in the appendix (Section A).

2.1 ACG Encoding of a CFG

Let us see our first example of an ACG, which is based on the translation from CFGs to ACGs given by de Groote (2001).

Consider the context-free grammar in Figure 1, where each rule is associated with a λ-term that specifies how the meaning of an expression built with that rule is determined by

[7] See footnote 3.

[8] A λ-term is *linear* if each λ binds exactly one occurrence of a variable. The restriction to linear λ-terms can be relaxed to *almost linear* λ-terms (Kanazawa, 2007, 2011, 2014) without affecting the computational properties of ACGs.

$s \to np\ vp$	$[\![vp]\!]\,[\![np]\!]$	$np \to$ Leslie	$\mathbf{Leslie}^{e}$
$vp \to v1\ np$	$\lambda x^{e}.[\![v1]\!]\,[\![np]\!]\,x$	$np \to$ Robin	$\mathbf{Robin}^{e}$
$vp \to v2\ s\_bar$	$\lambda x^{e}.[\![v2]\!]\,[\![s\_bar]\!]\,x$	$np \to$ Terry	$\mathbf{Terry}^{e}$
$vp \to v3\ s\_int$	$\lambda x^{e}.[\![v3]\!]\,[\![s\_int]\!]\,x$	$v1 \to$ hates	$\lambda y^{e} x^{e}.\mathbf{hate}^{e\to e\to t}\,y\,x$
$s\_bar \to$ that s	$[\![s]\!]$	$v1 \to$ likes	$\lambda y^{e} x^{e}.\mathbf{like}^{e\to e\to t}\,y\,x$
$s\_int \to$ whether s	$\mathbf{yn}^{t\to q}\,[\![s]\!]$	$v2 \to$ thinks	$\lambda y^{t} x^{e}.\mathbf{think}^{t\to e\to t}\,y\,x$
		$v2 \to$ claims	$\lambda y^{t} x^{e}.\mathbf{claim}^{t\to e\to t}\,y\,x$
		$v3 \to$ wonders	$\lambda y^{q} x^{e}.\mathbf{wonder}^{q\to e\to t}\,y\,x$
		$v3 \to$ knows	$\lambda y^{q} x^{e}.\mathbf{know}^{q\to e\to t}\,y\,x$

Figure 1: A CFG with Montague semantics.

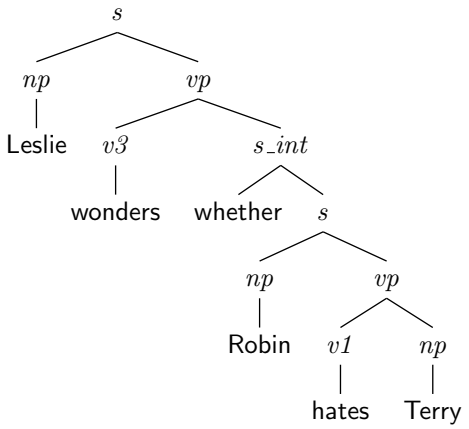


Figure 2: A CFG derivation tree.

the meanings of its immediate constituents. I use lower-case italic letters for nonterminals (i.e., syntactic categories) and sans serif for terminals (i.e., words). The type of a constant or variable is indicated by a superscript at its first occurrence. Here, e is the type of individual, t is the type of proposition, q is the type of question, and the constant $\mathbf{yn}$ denotes a function that turns a proposition into a yes-no (polar) question. According to this context-free grammar, the string Leslie wonders whether Robin hates Terry has the derivation tree in Figure 2 and its meaning is calculated to be

$$\begin{aligned}((\lambda y^{q} x^{e}.\mathbf{wonder}^{q\to e\to t}\,y\,x)(\mathbf{yn}^{t\to q}\,(((\lambda y^{e} x^{e}.\mathbf{hate}^{e\to e\to t}\,y\,x)\,\mathbf{Terry}^{e})\,\mathbf{Robin}^{e})))\,\mathbf{Leslie}^{e}\\ = \mathbf{wonder}^{q\to e\to t}\,(\mathbf{yn}^{t\to q}\,(\mathbf{hate}^{e\to e\to t}\,\mathbf{Terry}^{e}\,\mathbf{Robin}^{e}))\,\mathbf{Leslie}^{e}. \quad (1)\end{aligned}$$

Figure 3 lists the grammar entries of the ACG that encodes the CFG in Figure 1. Each rule of the CFG corresponds to an entry of the ACG. The correspondence should be fairly obvious. When a CFG has a rule $X \to w_0 X_1 w_1 \dots X_n w_n$, where $X, X_1, \dots, X_n$ are nonterminals and $w_0, w_1, \dots, w_n$ are strings of terminals, the corresponding ACG entry has the syntactic type $X_1 \to \dots \to X_n \to X$ and the λ-term $\lambda z_1^{str} \dots z_n^{str}.w_0 \circ z_1 \circ w_1 \circ \dots \circ z_n \circ w_n$ in the form dimension. The λ-term in the meaning dimension of the entry is obtained from the λ-term attached to the CFG rule by replacing $[\![X_1]\!], \dots, [\![X_n]\!]$ with appropriately typed variables $z_1, \dots, z_n$ and abstracting over them.

The derivation trees of the ACG are almost isomorphic to the derivation trees of the CFG. Each node of an ACG derivation tree is licensed by a grammar entry. Figure 4

$(\ np \to vp \to s,$	$\lambda z_1^{str} z_2^{str}.z_1 \circ z_2,$	$\lambda z_1^{e} z_2^{e\to t}.z_2 z_1$ $)$
$(\ v1 \to np \to vp,$	$\lambda z_1^{str} z_2^{str}.z_1 \circ z_2,$	$\lambda z_1^{e\to e\to t} z_2^{e} x^{e}.z_1 z_2 x$ $)$
$(\ v2 \to s\_bar \to vp,$	$\lambda z_1^{str} z_2^{str}.z_1 \circ z_2,$	$\lambda z_1^{t\to e\to t} z_2^{t} x^{e}.z_1 z_2 x$ $)$
$(\ v3 \to s\_int \to vp$	$\lambda z_1^{str} z_2^{str}.z_1 \circ z_2,$	$\lambda z_1^{q\to e\to t} z_2^{q} x^{e}.z_1 z_2 x$ $)$
$(\ s \to s\_bar,$	$\lambda z_1^{str}.\mathsf{that} \circ z_1,$	$\lambda z_1^{t}.z_1$ $)$
$(\ s \to s\_int,$	$\lambda z_1^{str}.\mathsf{whether} \circ z_1,$	$\lambda z_1^{t}.\mathbf{yn}^{t\to q}\ z_1$ $)$
$(\ np,$	Leslie,	$\mathbf{Leslie}^{e}$ $)$
$(\ np,$	Robin,	$\mathbf{Robin}^{e}$ $)$
$(\ np,$	Terry,	$\mathbf{Terry}^{e}$ $)$
$(\ v1,$	hates,	$\lambda y^{e} x^{e}.\mathbf{hate}^{e\to e\to t}\ y\ x$ $)$
$(\ v1,$	likes,	$\lambda y^{e} x^{e}.\mathbf{like}^{e\to e\to t}\ y\ x$ $)$
$(\ v2,$	thinks,	$\lambda y^{t} x^{e}.\mathbf{think}^{t\to e\to t}\ y\ x$ $)$
$(\ v2,$	claims,	$\lambda y^{t} x^{e}.\mathbf{claim}^{t\to e\to t}\ y\ x$ $)$
$(\ v3,$	wonders,	$\lambda y^{q} x^{e}.\mathbf{wonder}^{q\to e\to t}\ y\ x$ $)$
$(\ v3,$	knows,	$\lambda y^{q} x^{e}.\mathbf{know}^{q\to e\to t}\ y\ x$ $)$

Figure 3: The ACG encoding of the CFG in Figure 1.

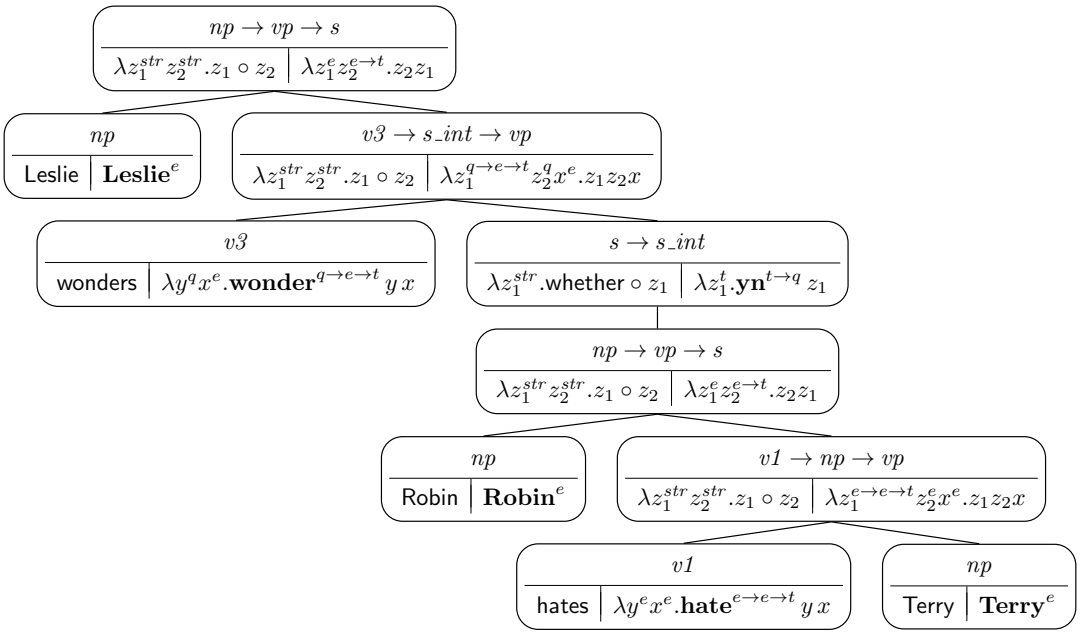


Figure 4: An ACG derivation tree representing a CFG derivation tree.

shows the ACG derivation tree corresponding to the CFG derivation tree in Figure 2. The well-formedness of an ACG derivation tree is checked by calculating the type of each subtree. Thus, a subtree T whose root node is labeled by a grammar entry of the form $(\alpha_1 \to \cdots \to \alpha_n \to p, M, N)$ is well-formed and is assigned type p if it has n immediate subtrees that are assigned types $\alpha_1, \ldots, \alpha_n$, respectively. The form/meaning associated with T is the result of applying the λ-term in the form/meaning dimension of the grammar entry $(\alpha_1 \to \cdots \to \alpha_n \to p, M, N)$ to the forms/meanings associated with the immediate subtrees of T. The λ-term in the form dimension associated with the entire derivation tree in Figure 4 is (after β-reduction) calculated to be

$$\mathsf{Leslie} \circ (\mathsf{wonders} \circ (\mathsf{whether} \circ (\mathsf{Robin} \circ (\mathsf{hates} \circ \mathsf{Terry})))),$$

and the corresponding λ-term in the meaning dimension is calculated to be (1) above. These λ-terms are guaranteed to be well-formed because of the existence of type substitutions σ and τ such that for every grammar entry (α, M, N), M is a well-formed λ-term of type $\sigma(\alpha)$ and N is a well-formed λ-term of type $\tau(\alpha)$. In the case of this ACG, σ maps each atomic syntactic type to *str*, and τ is the following type substitution:

$$\begin{aligned} s &\mapsto t, & v1 &\mapsto e \to e \to t, \\ np &\mapsto e, & v2 &\mapsto t \to e \to t, \\ vp &\mapsto e \to t, & v3 &\mapsto q \to e \to t. \\ s\_bar &\mapsto t, \\ s\_int &\mapsto q, \end{aligned}$$

The ACG in Figure 3 is particularly simple in two respects. First, the syntactic type of a grammar entry is always of the form $p_1 \to \cdots \to p_n \to p$ with $n \geq 0$, where $p_1, \ldots, p_n, p$ are all atomic types. ACGs with this property are (somewhat misleadingly) called *second-order* ACGs (see Kanazawa, 2009).[9] Second, in this ACG, the type substitution σ maps every atomic type to the type *str*. The next example is different in that σ maps some atomic syntactic types to complex types.

2.2 ACG encoding of a TAG

This example uses the method of encoding tree-adjoining grammars into ACGs due to de Groote (2002) and is loosely based on an example used by Pogodalla (2009, Chapter 2).

The tree-adjoining grammar in Figure 5 generates a fragment that is similar to the one generated by the CFG in Figure 1, but extended with *wh*-questions. For example, this TAG generates (2) through the derivation tree in Figure 6.

(2) Leslie wonders who Robin thinks that Terry hates

An interesting property of this TAG is that long distance *wh*-extraction must be mediated by bridge verbs that occur in auxiliary trees like the one for thinks.[10] Consequently, sentence (3) below is not generated. (Note that this is just an illustrative example.)

(3) Leslie wonders who Robin knows whether Terry hates

The ACG encoding of the TAG in Figure 5 coupled with a suitable semantics is given in Figure 7. We have a new constant $\mathbf{who}^{(e\to t)\to q}$ in the meaning dimension that turns a unary property into a *wh*-question. In this ACG, all atomic types are mapped to *str* by

[9] Second-order ACGs are roughly the same as what Kracht (2003) called *de Saussure grammars*.

[10] This treatment of *wh*-extraction in the TAG formalism is originally due to Kroch (1987). The account is slightly modified here for simplicity.

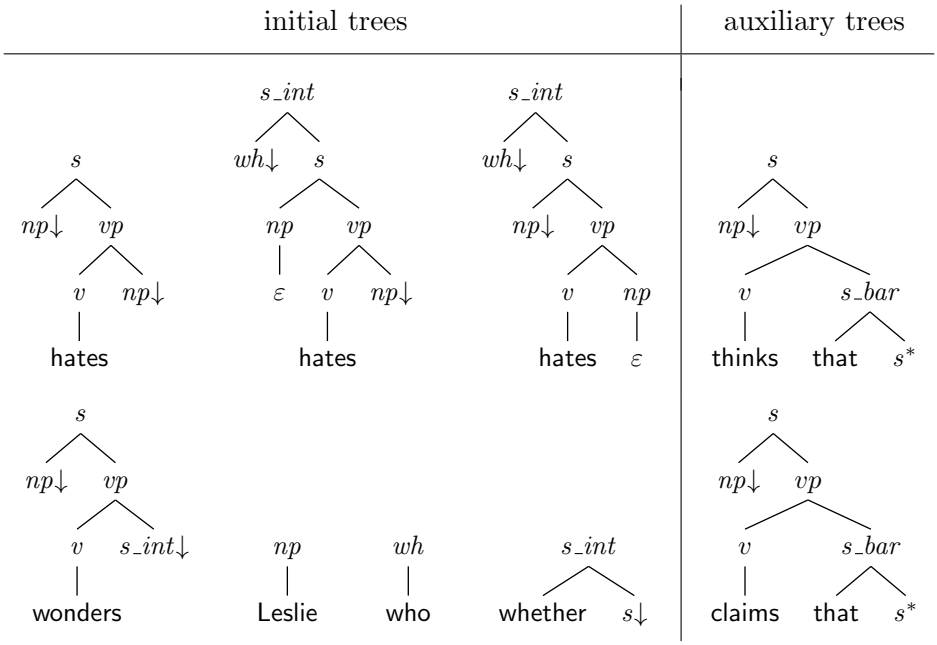


Figure 5: A tree-adjoining grammar. The arrow ↓ indicates substitution nodes and the asterisk ∗ indicates foot nodes of auxiliary trees. Each of the two auxiliary trees can *adjoin* into any node labeled with s (without ↓ or ∗). Three initial trees with likes (similar to those with hates), one with knows (similar to that with wonders), and one each with Robin and Terry (similar to that with Leslie) are not shown.

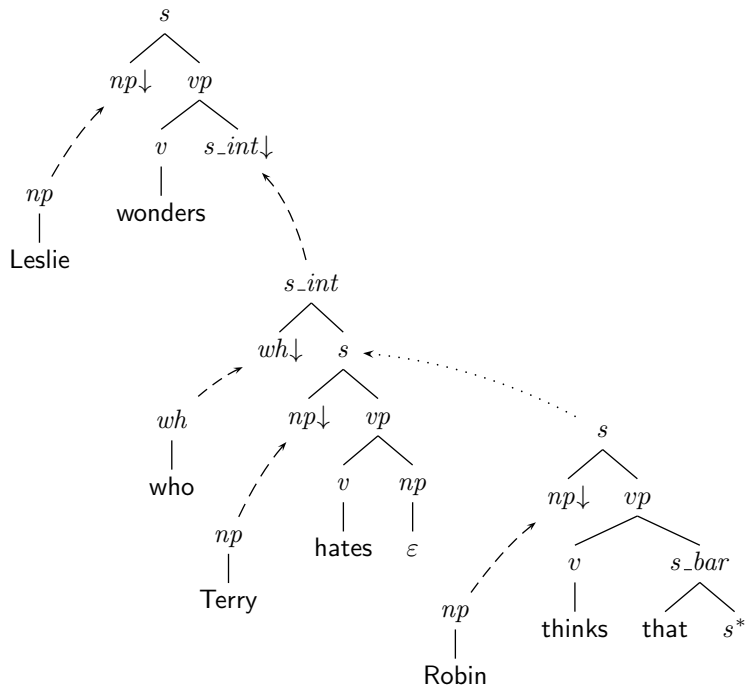


Figure 6: A TAG derivation tree. The dashed arrows indicate substitution and the dotted arrow adjunction.

$(\ np \to np \to s_A \to s,$	$\lambda z_1^{str} z_2^{str} z_3^{str \to str}.z_3(z_2 \circ (\mathsf{hates} \circ z_1)),$	$\lambda z_1^e z_2^e z_3^{t \to t}.\quad z_3(\mathbf{hate}^{e \to e \to t}\, z_1\, z_2)\ )$
$(\ np \to wh \to s_A \to s\_int,$	$\lambda z_1^{str} z_2^{str} z_3^{str \to str}.z_2 \circ (z_3(\varepsilon \circ (\mathsf{hates} \circ z_1))),$	$\lambda z_1^e z_2^{(e \to t) \to q} z_3^{t \to t}.$ $z_2(\lambda x^e.z_3(\mathbf{hate}^{e \to e \to t}\, z_1\, x))\ )$
$(\ np \to wh \to s_A \to s\_int,$	$\lambda z_1^{str} z_2^{str} z_3^{str \to str}.z_2 \circ (z_3(z_1 \circ (\mathsf{hates} \circ \varepsilon))),$	$\lambda z_1^e z_2^{(e \to t) \to q} z_3^{t \to t}.$ $z_2(\lambda x^e.z_3(\mathbf{hate}^{e \to e \to t}\, x\, z_1))\ )$
$(\ s\_int \to np \to s_A \to s,$	$\lambda z_1^{str} z_2^{str} z_3^{str \to str}.z_3(z_2 \circ (\mathsf{wonders} \circ z_1)),$	$\lambda z_1^q z_2^e z_3^{t \to t}.z_3(\mathbf{wonder}^{q \to e \to t}\, z_1\, z_2)\ )$
$(\ s\_int \to np \to s_A \to s,$	$\lambda z_1^{str} z_2^{str} z_3^{str \to str}.z_3(z_2 \circ (\mathsf{knows} \circ z_1)),$	$\lambda z_1^q z_2^e z_3^{t \to t}.z_3(\mathbf{know}^{q \to e \to t}\, z_1\, z_2)\ )$
$(\ np,$	$\mathsf{Leslie},$	$\mathbf{Leslie}^e\ )$
$(\ wh,$	$\mathsf{who},$	$\lambda y^{e \to t}\mathbf{who}^{(e \to t) \to q}(\lambda x^e.yx)\ )$
$(\ s \to s\_int,$	$\lambda z_1^{str}.\mathsf{whether} \circ z_1,$	$\lambda z_1^t.\mathbf{yn}^{t \to q}\, z_1\ )$
$(\ np \to s_A \to s_A,$	$\lambda z_1^{str} z_2^{str} x^{str}.z_2(z_1 \circ (\mathsf{thinks} \circ (\mathsf{that} \circ x))),$	$\lambda z_1^e z_2^{t \to t} x^t.z_2(\mathbf{think}\, x\, z_1)\ )$
$(\ np \to s_A \to s_A,$	$\lambda z_1^{str} z_2^{str} x^{str}.z_2(z_1 \circ (\mathsf{claims} \circ (\mathsf{that} \circ x))),$	$\lambda z_1^e z_2^{t \to t} x^t.z_2(\mathbf{claim}\, x\, z_1)\ )$
$(\ s_A,$	$\lambda x^{str}.x,$	$\lambda x^t.x\ )$

Figure 7: The ACG encoding of the TAG in Figure 5 coupled with a Montague semantics. There is one grammar entry in the ACG for each elementary tree of the TAG. The last entry in the ACG does not correspond to any elementary tree of the TAG and expresses the optionality of adjunction.

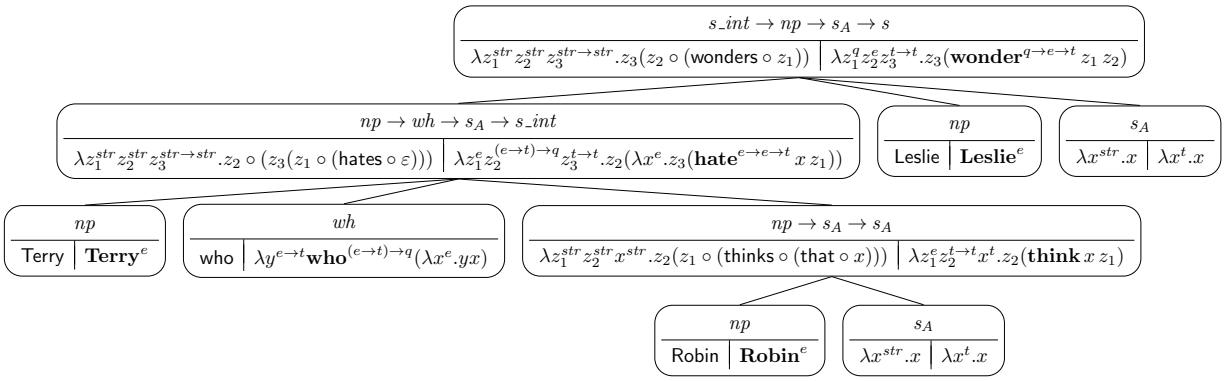


Figure 8: An ACG derivation tree representing a TAG derivation tree.

the type substitution σ except s_A, which is mapped to $str \to str$. The type substitution τ for the meaning dimension is as follows:

$$\begin{aligned} s &\mapsto t, \\ np &\mapsto e, \\ s\_int &\mapsto q, \\ wh &\mapsto (e \to t) \to q, \\ s_A &\mapsto t \to t. \end{aligned}$$

Figure 8 shows the ACG derivation tree for the pair

$$\Big(\mathsf{Leslie} \circ (\mathsf{wonders} \circ (\mathsf{who} \circ (\mathsf{Robin} \circ (\mathsf{thinks} \circ (\mathsf{that} \circ (\mathsf{Terry} \circ (\mathsf{hates} \circ \varepsilon))))))),$$
$$\mathbf{wonder}^{q \to e \to t}\, (\mathbf{who}^{(e \to t) \to q}\, (\lambda x^e.\mathbf{think}^{t \to e \to t}\, (\mathbf{hate}^{e \to e \to t}\, x\, \mathbf{Terry}^e)\, \mathbf{Robin}^e))\, \mathbf{Leslie}^e \Big).$$

Note that all entries of this ACG except the last contains at least one constant in the form dimension. An ACG all of whose entries have this property is said to be *lexicalized*.[11]

[11]This terminology comes from the TAG formalism, where a TAG is said to be *lexicalized* (Joshi and Schabes, 1997) if each of its elementary trees contains a terminal symbol.

$$\begin{array}{llll}
(\ wh \to (np \to s) \to q, & \lambda z_1^{str} z_2^{str \to str}.z_1 \circ (z_2\,\varepsilon), & \lambda z_1^{(e\to t)\to q} z_2^{e\to t}.z_1(\lambda x^e.z_2 x) &) \\
(\ wh & \mathsf{who} & \lambda y^{e\to t}.\mathbf{who}^{(e\to t)\to q}\,(\lambda x^e.yx) &) \\
(\ s \to s \to s, & \lambda z_1^{str} z_2^{str}.z_2 \circ (\mathsf{and} \circ z_1), & \lambda z_1^t z_2^t.\ \wedge^{t\to t\to t}\ z_1\, z_2 &) \\
(\ vp \to vp \to vp, & \lambda z_1^{str} z_2^{str}.z_2 \circ (\mathsf{and} \circ z_1), & \lambda z_1^{e\to t} z_2^{e\to t} x^e.\ \wedge^{t\to t\to t}\ (z_1 x)\,(z_2 x) &)
\end{array}$$

Figure 9: Additional entries for the ACG fragment handling *wh*-extraction and coordination.

It is easy to turn this ACG into a lexicalized one that generates the same set of string-meaning pairs.[12]

2.3 Encoding Freewheeling Extraction by Lambda Abstraction

The previous two examples of ACGs were both second-order ACGs. The derivations in a second-order ACG form a *local* set of trees,[13] and in this respect, second-order ACGs, together with well-known mildly context-sensitive grammars like TAGs, are "context-free" grammars in a generalized sense. Our next example is a higher-order ACG and extends the ACG in Section 2.1 by using λ-abstraction in derivations to handle extraction. We also add entries for coordination for good measure. The entries of this ACG are those in Figure 9 in addition to those in Figure 3. The new meaning constant $\wedge^{t\to t\to t}$ is the truth function for conjunction written in prefix notation. The type substitutions σ and τ for this ACG extend those for the ACG in Figure 3 by

$$\sigma(wh) = str, \quad \tau(wh) = (e \to t) \to q.$$

With a higher-order ACG, derivations are no longer trees (in general). They may contain nodes labeled by bound variables and λ-operators. (The binding relation between these nodes could be represented by arcs, so the structure of a derivation is a kind of graph.) We notate a bound variable node with an oval and a λ node with a diamond. The whole derivation will be a closed λ-term made up of grammar entries, bound variables, and λ. Each λ must bind exactly one occurrence of a bound variable; i.e., a derivation must be a linear λ-term. The form and meaning associated with a derivation are computed by interpreting λ-abstraction by λ-abstraction, replacing the type α of bound variable x^α with $\sigma(\alpha)$ and $\tau(\alpha)$, respectively.

Here are a few string-meaning pairs generated by this ACG:

$$\Big(\ \mathsf{Leslie} \circ (\mathsf{wonders} \circ (\mathsf{who} \circ (\mathsf{Terry} \circ (\mathsf{hates} \circ \varepsilon)))),\ \mathbf{wonder}^{q\to e\to t}\,(\mathbf{who}^{(e\to t)\to q}\,(\lambda x^e.\mathbf{hate}^{e\to e\to t}\,x\,\mathbf{Terry}^e))\,\mathbf{Leslie}^e\ \Big) \tag{4}$$

$$\Big(\ \mathsf{Leslie} \circ (\mathsf{wonders} \circ (\mathsf{who} \circ (\mathsf{Robin} \circ (\mathsf{thinks} \circ (\mathsf{that} \circ (\mathsf{Terry} \circ (\mathsf{hates} \circ \varepsilon))))))),\ \mathbf{wonder}^{q\to e\to t}\,(\mathbf{who}^{(e\to t)\to q}\,(\lambda x^e.\mathbf{think}^{t\to e\to t}\,(\mathbf{hate}^{e\to e\to t}\,x\,\mathbf{Terry}^e)\,\mathbf{Robin}^e))\,\mathbf{Leslie}^e\ \Big) \tag{5}$$

$$\Big(\ \mathsf{Leslie} \circ (\mathsf{wonders} \circ (\mathsf{who} \circ (\varepsilon \circ (\mathsf{likes} \circ \mathsf{Robin})))),\ \mathbf{wonder}^{q\to e\to t}\,(\mathbf{who}^{(e\to t)\to q}\,(\lambda x^e.\mathbf{like}^{e\to e\to t}\,\mathbf{Robin}^e\,x))\ \Big) \tag{6}$$

[12] It is known that every second-order ACG can be converted to an equivalent lexicalized second-order ACG (Kanazawa and Yoshinaka, 2005), but this generally makes σ a more complex type substitution.

[13] A set of trees is *local* if membership of a tree in the set depends only on the "local" fragments of the tree consisting of a node together with all its children (see Comon et al., 2008).

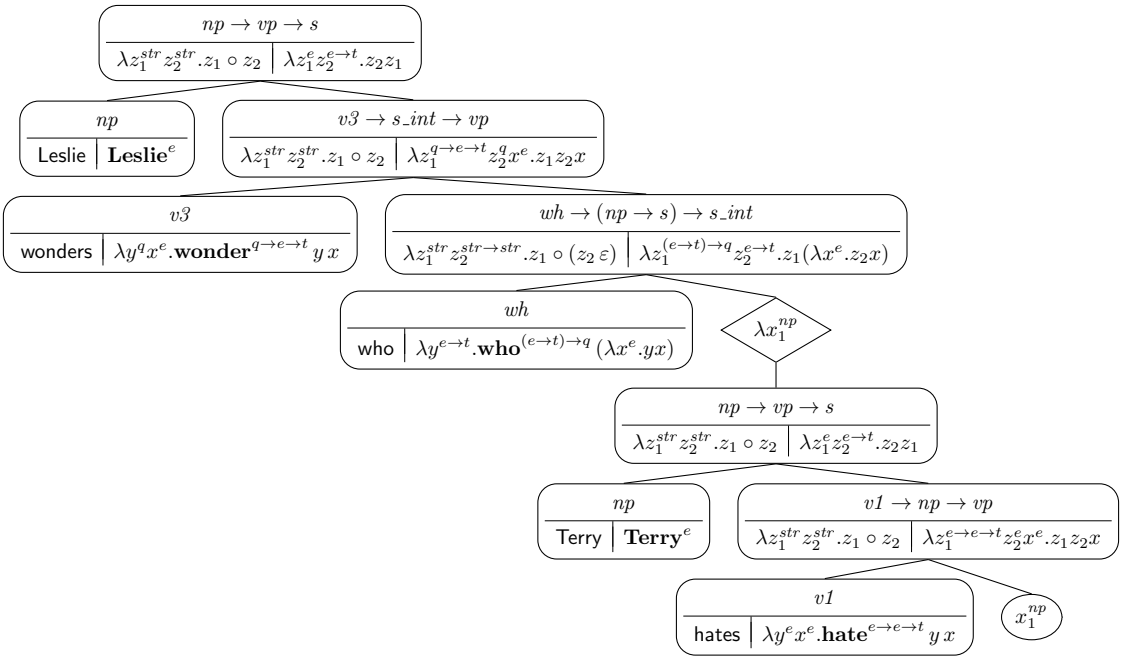


Figure 10: An ACG derivation λ-term that is not a tree.

The derivation for (4) is shown in Figure 10.

This ACG allows *wh*-extraction freely within the constraint imposed by linearity of derivations. In addition to pairs like (4)–(6), string-meaning pairs like the following are generated:

$$\Big(\mathsf{Leslie} \circ (\mathsf{wonders} \circ (\mathsf{who} \circ (\mathsf{Robin} \circ (\mathsf{thinks} \circ (\mathsf{that} \circ (\varepsilon \circ (\mathsf{hates} \circ \mathsf{Terry})))))))),$$
$$\mathbf{wonder}^{q \to e \to t}\, (\mathbf{who}^{(e \to t) \to q}\, (\lambda x^{e}.\mathbf{think}^{t \to e \to t}\, (\mathbf{hate}^{e \to e \to t}\, \mathbf{Terry}^{e}\, x)\, \mathbf{Robin}^{e}))\, \mathbf{Leslie}^{e} \Big) \tag{7}$$

$$\Big(\mathsf{Leslie} \circ (\mathsf{wonders} \circ (\mathsf{who} \circ (\mathsf{Robin} \circ (\mathsf{knows} \circ (\mathsf{whether} \circ (\mathsf{Terry} \circ (\mathsf{hates} \circ \varepsilon)))))))),$$
$$\mathbf{wonder}^{q \to e \to t}\ (\mathbf{who}^{(e \to t) \to q}\, (\lambda x^{e}.\mathbf{know}^{q \to e \to t}\, (\mathbf{yn}^{t \to q}\, (\mathbf{hate}^{e \to e \to t}\, x\, \mathbf{Terry}^{e}))\, \mathbf{Robin}^{e}))\ \mathbf{Leslie}^{e} \Big) \tag{8}$$

$$\Big(\mathsf{Leslie} \circ (\mathsf{wonders} \circ (\mathsf{who} \circ ((\mathsf{Robin} \circ (\mathsf{likes} \circ \varepsilon)) \circ (\mathsf{and} \circ (\mathsf{Terry} \circ (\mathsf{hates} \circ \mathsf{Robin})))))),$$
$$\mathbf{wonder}^{q \to e \to t}\ (\mathbf{who}^{(e \to t) \to q}\, (\lambda x^{e}.\, \wedge^{t \to t \to t}\, (\mathbf{like}^{e \to e \to t}\, x\, \mathbf{Robin}^{e})\, (\mathbf{hates}^{e \to e \to t}\, \mathbf{Robin}^{e}\, \mathbf{Terry}^{e})))\ \mathbf{Leslie}^{e} \Big) \tag{9}$$

In Section 3, I show how one can constrain λ-abstraction to block sentences like (8) and (9).

2.4 Some Important Properties of ACGs

We have seen three examples of ACGs. The first two (Sections 2.1 and 2.2) are second-order ACGs faithfully encoding a CFG and a TAG, respectively. No λ-abstraction occurs in

the derivations of these grammars. Neither is lexicalized, but the second ACG is nearly so and is easily converted to a lexicalized ACG.[14] The second ACG maps an atomic syntactic type s_A (corresponding to the auxiliary trees of the encoded TAG) to a nonatomic type $str \to str$. This is necessary to simulate the operation of adjunction in TAGs.

Formal properties of second-order ACGs are by now fairly well-understood (see Kanazawa, 2007, 2011, 2009). In particular, their string-generating power is the same as that of *multiple context-free grammars* (Seki et al., 1991) or *linear context-free rewriting systems* (Vijay-Shanker et al., 1987).[15]

Our third example (Section 2.3) is a higher-order ACG in that it contains a grammar entry with a syntactic type that is not of the form $p_1 \to \cdots \to p_n \to p$ (with $p_1, \ldots, p_n, p$ atomic). Compared to second-order ACGs, higher-order ACGs are of very high complexity[16] and their formal properties are not very well-understood. Although the ACG in Section 2.3 is not lexicalized, it can easily be converted to a *semilexicalized* ACG by combining the first two entries in Figure 9.[17]

It is important to remember that neither lexicalization nor semilexicalization is required of ACGs in general. This is one of several crucial differences that sets ACGs apart from categorial grammars like Lambek grammars, where every entry is attached to an overt word.

We have used constants $\circ$ and ε to represent the operation of string concatenation and the empty string, respectively. Alternatively, we can view them as uninterpreted symbols; the form component of the generated pairs will then be trees of a certain kind, rather than strings.[18] This shift in perspective will be useful in the next section.

3 Syntactic Features for Regular Constraints

Traditionally, various *island constraints* have been posited to account for unacceptable instances of *wh*-extraction. Thus, the whether-clause in (8) creates a *whether island*, and (9) involves a *coordinate structure island*.[19] In this section, I show how a general construction I gave in Kanazawa (2006) automatically supplies a finite-valued syntactic feature that captures island effects. A similar use of a syntactic feature to control *wh*-extraction is found in Pogodalla and Pompigne (2012). My point here is that whenever one can express a relevant constraint in terms of a regular set of trees (or strings), a syntactic feature that captures that constraint is automatically obtained.

[14] Since all second-order ACGs can be lexicalized, it is possible to lexicalize the ACG in Section 2.1 as well. In general, lexicalization of a second-order ACG requires a grammar transformation similar to conversion to Greibach normal form (Kanazawa and Yoshinaka, 2005).

[15] This was first proved by Salvati (2007).

[16] It has recently been shown that the emptiness and universal recognition problems for higher-order ACGs are of non-elementary complexity (Lazić and Schmitz, 2014); it is not known whether they are decidable. In comparison, the emptiness and universal recognition problems for second-order ACGs are P-complete and in EXPTIME (but at least PSPACE-hard), respectively (Kanazawa, 2011).

[17] An ACG is called *semilexicalized* (Salvati, 2005) if entries with higher-order syntactic types contain at least one constant in the form dimension. Yoshinaka (2006) showed that semilexicalized ACGs can be lexicalized. The universal recognition problem for semilexicalized ACGs is decidable (Salvati, 2005).

[18] We can even give a general definition of ACGs in such a way that they generate pairs of (not necessarily binary) trees and λ-terms. The string-meaning pairs of the grammar will then be obtained by taking the yields of the trees in the generated pairs.

[19] The grammatical status of these purported island violations has been in dispute in recent years (see, e.g., Phillips, 2006, 2013; Hofmeister and Sag, 2010; Chaves, 2012). To the extent that the apparent unacceptability of *wh*-extraction from these "islands" is due to extra-grammatical factors, my case for the usefulness of syntactic features to control extraction is weakened. Here, I assume that at least some purported islands are grammatical in nature.

Let us take the ACG in Section 2.3 (call it G) and treat $\circ$ and ε as uninterpreted symbols. The ACG generates pairs of binary trees and λ-term representations of meanings. The symbol ε occupies the positions of *wh*-extraction sites; they may be thought of as traces of *wh*-movement. In order to express the fact that certain constructions are islands, we introduce two unary function symbols I, O of type $str \to str$ and modify the form dimension of some of the entries as follows:

$$\begin{aligned}
&(\ s \to s_int,\ \lambda z_1^{str}.\mathsf{whether} \circ z_1,\ \dots\ ) \rightsquigarrow (\ s \to s_int,\ \lambda z_1^{str}.\,\mathsf{I}\,(\mathsf{whether} \circ z_1),\ \dots\ )\\
&(\ wh \to (np \to s) \to q,\ \lambda z_1^{str} z_2^{str\to str}.z_1 \circ (z_2\,\varepsilon),\ \dots\ )\\
&\qquad\qquad \rightsquigarrow (\ wh \to (np \to s) \to s_int,\ \lambda z_1^{str} z_2^{str\to str}.\,\mathsf{I}\,(z_1 \circ \mathsf{O}\,(z_2\,\varepsilon)),\ \dots\ )\\
&(\ s \to s \to s,\ \lambda z_1^{str} z_2^{str}.z_2 \circ (\mathsf{and} \circ z_1),\ \dots\ ) \rightsquigarrow (\ s \to s \to s,\ \lambda z_1^{str} z_2^{str}.\,\mathsf{I}\,(z_2 \circ (\mathsf{and} \circ z_1)),\ \dots\ )\\
&(\ vp \to vp \to vp,\ \lambda z_1^{str} z_2^{str}.z_2 \circ (\mathsf{and} \circ z_1),\ \dots\ )\\
&\qquad\qquad \rightsquigarrow (\ vp \to vp \to vp,\ \lambda z_1^{str} z_2^{str}.\,\mathsf{I}\,(z_2 \circ (\mathsf{and} \circ z_1)),\ \dots\ )
\end{aligned}$$

The symbol I serves to mark islands for *wh*-extraction, and the symbol O marks the "scope" of fronted *wh*-phrases. (We think of an occurrence of ε inside the scope of a *wh*-phrase as "bound" by that *wh*-phrase.) Call the modified ACG G'. The form components of some of the generated pairs now look as follows:

$$\mathsf{Leslie} \circ (\mathsf{wonders} \circ \mathsf{I}\left(\mathsf{who} \circ \mathsf{O}\left(\mathsf{Terry} \circ (\mathsf{hates} \circ \varepsilon)\right)\right)) \tag{4′}$$

$$\mathsf{Leslie} \circ (\mathsf{wonders} \circ \mathsf{I}\left(\mathsf{who} \circ \mathsf{O}\left(\mathsf{Robin} \circ (\mathsf{thinks} \circ (\mathsf{that} \circ (\mathsf{Terry} \circ (\mathsf{hates} \circ \varepsilon))))\right)\right)) \tag{5′}$$

$$\mathsf{Leslie} \circ (\mathsf{wonders} \circ \mathsf{I}\left(\mathsf{who} \circ \mathsf{O}\left(\varepsilon \circ (\mathsf{likes} \circ \mathsf{Robin})\right)\right)) \tag{6′}$$

$$\mathsf{Leslie} \circ (\mathsf{wonders} \circ \mathsf{I}\left(\mathsf{who} \circ \mathsf{O}\left(\mathsf{Robin} \circ (\mathsf{knows} \circ \mathsf{I}\left(\mathsf{whether} \circ (\mathsf{Terry} \circ (\mathsf{hates} \circ \varepsilon))\right)\right)\right)\right)) \tag{8′}$$

$$\mathsf{Leslie} \circ (\mathsf{wonders} \circ \mathsf{I}\left(\mathsf{who} \circ \mathsf{O}\left(\mathsf{I}\left((\mathsf{Robin} \circ (\mathsf{likes} \circ \varepsilon)) \circ (\mathsf{and} \circ (\mathsf{Terry} \circ (\mathsf{hates} \circ \mathsf{Robin})))\right)\right)\right)) \tag{9′}$$

The forms (4′)–(6′) obey the island constraints and (8′)–(9′) don't. Figure 11 shows (5′) and (8′) in graphical notation.

The trees that do and those that don't obey island constraints can be distinguished by the following deterministic *bottom-up finite tree automaton* (see, e.g., Comon et al., 2008), which has just two states, [GAP −] ("contains no unbound gap") and [GAP +] ("contains an unbound gap"):

$$\begin{aligned}
\varepsilon &\to [\textsc{gap}\,+],\\
a &\to [\textsc{gap}\,-] \quad \text{for each terminal symbol } a,\\
[\textsc{gap}\,-] \circ [\textsc{gap}\,-] &\to [\textsc{gap}\,-],\\
[\textsc{gap}\,-] \circ [\textsc{gap}\,+] &\to [\textsc{gap}\,+],\\
[\textsc{gap}\,+] \circ [\textsc{gap}\,-] &\to [\textsc{gap}\,+],\\
\mathsf{I}\,[\textsc{gap}\,-] &\to [\textsc{gap}\,-],\\
\mathsf{O}\,[\textsc{gap}\,+] &\to [\textsc{gap}\,-].
\end{aligned}$$

Note that there is no transition for $[\textsc{gap}\,+] \circ [\textsc{gap}\,+]$. The automaton accepts a tree if it can be rewritten to $[\textsc{gap}\,-]$ by applying these transitions bottom-up. The accepted trees are those such that

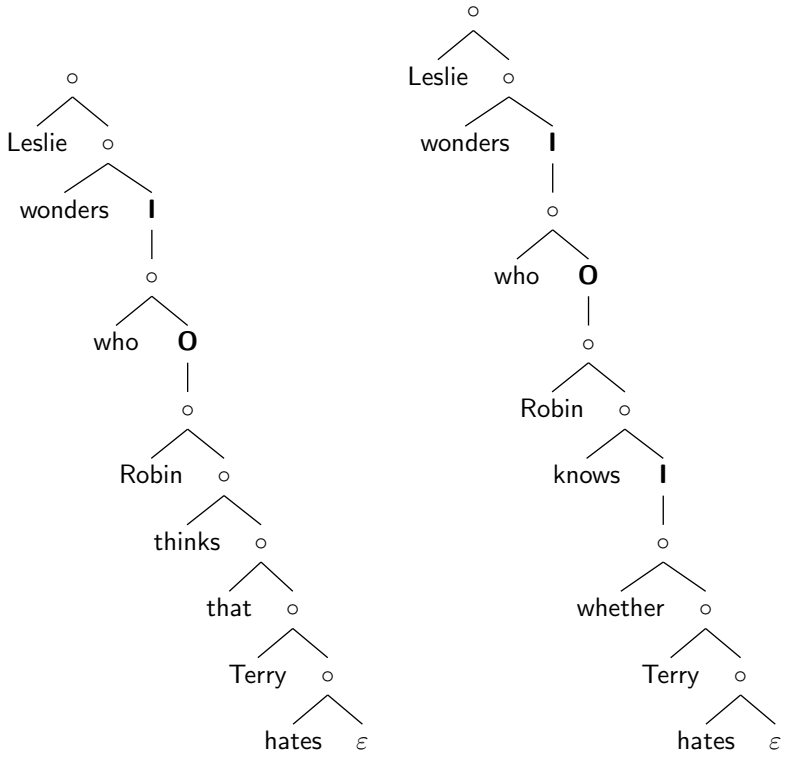


Figure 11: A tree that obeys island constraints (left) and one that does not (right).

- no subtree contains more than one unbound gap,
- no subtree that is an island (marked by **I**) contains any unbound gap, and
- the entire tree contains no unbound gap.

It is important to note that this characterization of island constraints in terms of a regular set of trees is possible no matter what construction creates an island, provided that a fronted *wh*-phrase (together with its scope) creates an island of its own.[20]

We modified the ACG G in Section 2.3 into an ACG G' which uses new constants **I** and **O** in the form dimension and which allows a simple characterization of the effects of islands in terms of a regular set R of trees (represented in the form of a finite tree automaton). By a construction of Kanazawa (2006, Lemma 3.3 and Theorem 5.1), we can form an ACG G'' such that

- G'' generates (M, N) if and only if G' generates (M, N) and $M \in R$,
- G' is the image of G'' under a type substitution that maps atomic syntactic types of G'' to atomic syntactic types of G'.

I do not give a formal definition of this construction here, but merely illustrate it with examples. An atomic type of G'' is of the form $p[\textsc{gap} \pm]$, where p is an atomic type of G'.[21]

[20]This is why Pogodalla and Pompigne (2012) needed a much more complex, infinite-valued feature mechanism to capture the fact that a tensed clause creates a scope island. The scope of a quantifier may contain a variable that is bound by another quantifier higher up, so quantification does not itself create a scope island.

[21]In the general case, this is a little more complex. If $\sigma(p)$ contained k occurrences of *str* instead of just one, then an atomic type of G'' that gets mapped to p would have k copies of the GAP feature attached to p.

Each grammar entry (α, M, N) of G' gives rise to zero, one, or more grammar entries of the form (α', M, N) such that α is the result of erasing all the feature specifications from α', and α' "induces" a typing of M consistent with the tree automaton for R. Let us use the entry

$$(\ wh \to (np \to s) \to s\_int,\ \lambda z_1^{str} z_2^{str \to str}.\, \mathsf{I}\,(z_1 \circ \mathsf{O}\,(z_2\,\varepsilon)),\ \lambda z_1^{(e\to t)\to q} z_2^{e\to t}.z_1(\lambda x^e.z_2 x)\ )$$

to illustrate this. Consider

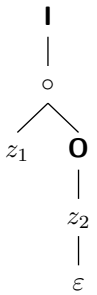
$$wh[\textsc{gap}\ g_1] \to (np[\textsc{gap}\ g_2] \to s[\textsc{gap}\ g_3]) \to s\_int[\textsc{gap}\ g_4],$$

where $g_1, g_2, g_3, g_4 \in \{-, +\}$, as a feature-specified version of the syntactic type of the above entry. Using g_1, g_2, g_3, we add new transitions

$$z_1 \to [\textsc{gap}\ g_1],$$
$$z_2\,[\textsc{gap}\ g_2] \to [\textsc{gap}\ g_3]$$

to the tree automaton and run it on the input tree

$\mathsf{I}\,(z_1 \circ \mathsf{O}\,(z_2\,\varepsilon)) =$



(the "body" of the λ-term in the form dimension of the entry) to see if it can be rewritten to $[\textsc{gap}\ g_4]$. Since ε gets rewritten to $[\textsc{gap}\ +]$, we must have $g_2 = +$. Since the only transition for O is $\mathsf{O}\,[\textsc{gap}\ +] \to [\textsc{gap}\ -]$, we must have $g_3 = +$. Likewise, since the only transition for I is $\mathsf{I}\,[\textsc{gap}\ -] \to [\textsc{gap}\ -]$, we must have $g_4 = -$ for the tree to be rewritten to $[\textsc{gap}\ g_4]$. Since we must have $[\textsc{gap}\ g_1] \circ [\textsc{gap}\ -] \to [\textsc{gap}\ -]$, it follows that g_1 must be $-$. So the only possibility is $(g_1, g_2, g_3, g_4) = (-, +, +, -)$.[22]

The entire grammar G'' is given in Figure 12. (We write $[-]$ and $[+]$ instead of $[\textsc{gap}\ -]$ and $[\textsc{gap}\ +]$ to save space.)

Finally, erasing I and O from G'' and restoring the original λ-terms in the form dimension gives a grammar G''' that generates the desired subset of the form-meaning pairs generated by G. The method is summarized in Figure 13.[23]

[22] This corresponds to the alternative typing

$$\lambda z_1^{[\textsc{gap}\ -]} z_2^{[\textsc{gap}\ +]\to[\textsc{gap}\ +]}.\, \mathsf{I}^{[\textsc{gap}\ -]\to[\textsc{gap}\ -]}\,(z_1 \circ^{[\textsc{gap}\ -]\to[\textsc{gap}\ -]\to[\textsc{gap}\ -]} \mathsf{O}^{[\textsc{gap}\ +]\to[\textsc{gap}\ -]}\,(z_2\,\varepsilon^{[\textsc{gap}\ +]}))$$

of the λ-term in the form dimension of the entry.

[23] There remains the problem of blocking (7) while allowing sentences like (6) and Leslie wonders who Robin thinks hates Terry. This could be done, for example, by letting the $np \to vp \to s$ entry specify subject as an island and positing special entries for *in-situ wh*-subjects and for bridge verbs combining with a vp, à la GPSG (Gazdar et al., 1985).

It is also possible to add *across-the-board wh*-extraction to the present fragment by using a three-valued feature $[\textsc{gap}\ -]$, $[\textsc{gap}\ +]$ ("properly contains a gap"), $[\textsc{gap}\ =]$ ("is a gap") instead and adding the following entry to G', where E is a new unary function symbol that requires its argument to be empty:

$$(\,(np \to s) \to (np \to s) \to np \to s,\quad \lambda z_1^{str\to str} z_2^{str\to str} x^{str}.\ \mathsf{I}\,(\mathsf{O}\,(z_2\,\varepsilon) \circ (\mathsf{and} \circ \mathsf{O}\,(z_1\,\varepsilon))) \circ (\mathsf{E}\,x),\quad \lambda z_1^{e\to t} z_2^{e\to t} x^e.\ \wedge^{t\to t\to t}\ (z_1 x)\,(z_2 x)\,)$$

(The position of $\mathsf{E}\,x$ may look unnatural, but what matters is the final product, a grammar with none of these markers.)

$(\ np[g_1] \to vp[g_2] \to s[g_3],$	$\lambda z_1^{str} z_2^{str}.z_1 \circ z_2,$	$\lambda z_1^{e} z_2^{e\to t}.z_2\, z_1$ $)$
$(\ v1[g_1] \to np[g_2] \to vp[g_3],$	$\lambda z_1^{str} z_2^{str}.z_1 \circ z_2,$	$\lambda z_1^{e\to e\to t} z_2^{e} x^{e}.z_1\, z_2 x$ $)$
$(\ v2[g_1] \to s\_bar[g_2] \to vp[g_3],$	$\lambda z_1^{str} z_2^{str}.z_1 \circ z_2,$	$\lambda z_1^{t\to e\to t} z_2^{t} x^{e}.z_1\, z_2 x$ $)$
$(\ v3[g_1] \to s\_int[g_2] \to vp[g_3]$	$\lambda z_1^{str} z_2^{str}.z_1 \circ z_2,$	$\lambda z_1^{q\to e\to t} z_2^{q} x^{e}.z_1\, z_2 x$ $)$
$(\ s[g] \to s\_bar[g],$	$\lambda z_1^{str}.\mathsf{that} \circ z_1,$	$\lambda z_1^{t}.z_1$ $)$
$(\ s[-] \to s\_int[-],$	$\lambda z_1^{str}.\mathsf{I}\,(\mathsf{whether} \circ z_1),$	$\lambda z_1^{t}.\mathbf{yn}^{t\to q}\, z_1$ $)$
$(\ np[-],$	Leslie,	$\mathbf{Leslie}^{e}$ $)$
$(\ np[-],$	Robin,	$\mathbf{Robin}^{e}$ $)$
$(\ np[-],$	Terry,	$\mathbf{Terry}^{e}$ $)$
$(\ v1[-],$	hates,	$\lambda y^{e} x^{e}.\mathbf{hate}^{e\to e\to t}\, y\, x$ $)$
$(\ v1[-],$	likes,	$\lambda y^{e} x^{e}.\mathbf{like}^{e\to e\to t}\, y\, x$ $)$
$(\ v2[-],$	thinks,	$\lambda y^{t} x^{e}.\mathbf{think}^{t\to e\to t}\, y\, x$ $)$
$(\ v2[-],$	claims,	$\lambda y^{t} x^{e}.\mathbf{claim}^{t\to e\to t}\, y\, x$ $)$
$(\ v3[-],$	wonders,	$\lambda y^{q} x^{e}.\mathbf{wonder}^{q\to e\to t}\, y\, x$ $)$
$(\ v3[-],$	knows,	$\lambda y^{q} x^{e}.\mathbf{know}^{q\to e\to t}\, y\, x$ $)$
$(\ wh[-] \to (np[+] \to s[+]) \to s\_[-],$	$\lambda z_1^{str} z_2^{str\to str}.\mathsf{I}\,(z_1 \circ \mathsf{O}\,(z_2\, \varepsilon)),$	$\lambda z_1^{(e\to t)\to q} z_2^{e\to t}.z_1(\lambda x^{e}.z_2 x)$ $)$
$(\ wh[-]$	who	$\lambda y^{e\to t}.\mathbf{who}^{(e\to t)\to q}\,(\lambda x^{e}.y x)$ $)$
$(\ s[-] \to s[-] \to s[-],$	$\lambda z_1^{str} z_2^{str}.\mathsf{I}\,(z_2 \circ (\mathsf{and} \circ z_1)),$	$\lambda z_1^{t} z_2^{t}.\ \wedge^{t\to t\to t}\ z_1\, z_2$ $)$
$(\ vp[-] \to vp[-] \to vp[-],$	$\lambda z_1^{str} z_2^{str}.\mathsf{I}\,(z_2 \circ (\mathsf{and} \circ z_1)),$	$\lambda z_1^{e\to t} z_2^{e\to t} x^{e}.\ \wedge^{t\to t\to t}\ (z_1 x)\,(z_2 x)$ $)$

Figure 12: An ACG with island and *wh*-scope markers that respects island constraints. Here, $(g_1, g_2, g_3) \in \{(-,-,-),(-,+,+),(+,-,+)\}$ and $g \in \{-,+\}$.

G		G'		G''		G'''
with unconstrained *wh*-extraction	$\rightsquigarrow$	with island and *wh*-scope markers	$\rightsquigarrow$	with markers respecting islands	$\rightsquigarrow$	without markers respecting islands

Figure 13: Deriving an ACG with a feature that respects *wh*-extraction islands.

Two remarks about the island and *wh*-scope markers I, O are in order. First, these markers are devices that we employed to *discover* an appropriate syntactic feature to use; they are *not* part of the final grammar that is obtained by this method. The "derivation" of the grammar G''' (Figure 13) has nothing to do with the derivations of individual sentences generated by G'''. Second, the present example is simple enough that it's easy to come up with a suitable feature mechanism without the help of these markers. The advantage of using them is that the grammar automatically comes with a correctness proof. The use of these markers reduces the linguist's job to the specification of the constraint in terms of a regular set of trees. As long as this specification is correct, the grammar is guaranteed to be correct.

4 Directional Slashes and Syntactic Features

The most popular style of linguistic analysis based on the formalism of ACGs has been the one exemplified by the fragment in Muskens (2003), which may be loosely described as "non-directionalization" of a Lambek grammar. Moot (2014) quite rightly pointed out that this approach does not scale well; once you add constructions like coordination, adverbial modification, etc., that are usually treated by Lambek grammars using complex functional types that take functional types as arguments, it seems that constraints on word order imposed by directional slashes of the Lambek calculus cannot be replicated by any amount of tweaking in the form dimension of the ACG written in this style. Needless to say, this does not make the formalism of ACGs "descriptively inadequate". Clearly, other styles of analysis, perhaps incorporating some ideas from TAGs, are possible within the

ACG formalism. Even the CFG-based style of analysis we have used in Sections 2.1 and 2.3 is immune to some of the criticisms of Moot (2014), namely, the ones based on adverbial modification and coordination of standard constituents. However, non-constituent coordination, a traditional stronghold of categorial grammars, does seem to present a difficulty for any ACG analysis that treats non-standard constituents as resulting from extraction (modeled by λ-abstraction).

I illustrate the problem in Section 4.1 using Right Node Raising, still following the same style of analysis from Sections 2.1 and 2.3. The solution I propose is to use a finite-valued syntactic feature again. A systematic use of this technique leads to a translation from an *arbitrary* Lambek grammar into an ACG with features (Section 4.2). In Section 4.3, I apply the technique to the treatment of Gapping by Kubota and Levine (2014a,b), which uses their *hybrid type-logical grammar* formalism. I end with my skepticism about treating Gapping with left- and right-peripheral extractions, suggesting that Lambek's directional slashes lack the flexibility one needs to control the extraction sites involved in Gapping (Section 4.4).

4.1 Right Node Raising

In Lambek calculus, Right Node Raising has been analyzed as coordination of two expressions of type s/X (for some type X). For example, sentence (10) would involve assigning the conjunction and the type $((s/np)\backslash(s/np))/(s/np)$:

(10) Terry hates and Leslie likes Robin.

A naive attempt to translate this entry for and into an ACG grammar entry would result in either of the following:

$$\begin{array}{l}(\,(np \to s) \to (np \to s) \to np \to s,\\ \quad \lambda z_1^{str\to str} z_2^{str\to str} x^{str}.((z_2\,\varepsilon) \circ (\mathsf{and} \circ (z_1\,\varepsilon))) \circ x,\ \lambda z_1^{e\to t} z_2^{e\to t} x^e.\ \wedge^{t\to t\to t}\ (z_1\,x)\,(z_2\,x)\,)\end{array} \tag{11}$$

$$\begin{array}{l}(\,(np \to s) \to (np \to s) \to np \to s,\\ \quad \lambda z_1^{str\to str} z_2^{str\to str} x^{str}.(z_2\,\varepsilon) \circ (\mathsf{and} \circ (z_1\,x)),\ \lambda z_1^{e\to t} z_2^{e\to t} x^e.\ \wedge^{t\to t\to t}\ (z_1\,x)\,(z_2\,x)\,)\end{array} \tag{12}$$

As pointed out by Moot (2014), adopting either of these entries in an ACG would result in overgeneration. If we add (11) to the fragment in Section 2.1, the following pair (with the meaning "Terry hates Robin and Robin likes Leslie") will be generated, with the derivation in Figure 14:

$$\begin{array}{l}(\,((\mathsf{Terry} \circ (\mathsf{hates} \circ \varepsilon)) \circ (\mathsf{and} \circ (\varepsilon \circ (\mathsf{likes} \circ \mathsf{Leslie})))) \circ \mathsf{Robin},\\ \quad \wedge^{t\to t\to t}\ (\mathbf{like}^{e\to e\to t}\,\mathbf{Leslie}^e\,\mathbf{Robin}^e)\,(\mathbf{hate}^{e\to e\to t}\,\mathbf{Robin}^e\,\mathbf{Terry}^e)\,).\end{array}$$

If we add (12) instead, then the following pair (with the same meaning) will be generated with a similar derivation:

$$\begin{array}{l}(\,(\mathsf{Terry} \circ (\mathsf{hates} \circ \varepsilon)) \circ (\mathsf{and} \circ (\mathsf{Robin} \circ (\mathsf{likes} \circ \mathsf{Leslie}))),\\ \quad \wedge^{t\to t\to t}\ (\mathbf{like}^{e\to e\to t}\,\mathbf{Leslie}^e\,\mathbf{Robin}^e)\,(\mathbf{hate}^{e\to e\to t}\,\mathbf{Robin}^e\,\mathbf{Terry}^e)\,).\end{array}$$

In the former approach, ungrammatical strings are generated, while in the latter approach, grammatical strings are paired with incorrect meanings.

Let us concentrate on the first approach (11) and see if we can remedy it. The Lambek entry for and requires each of the conjunct in Right Node Raising to contain an np gap

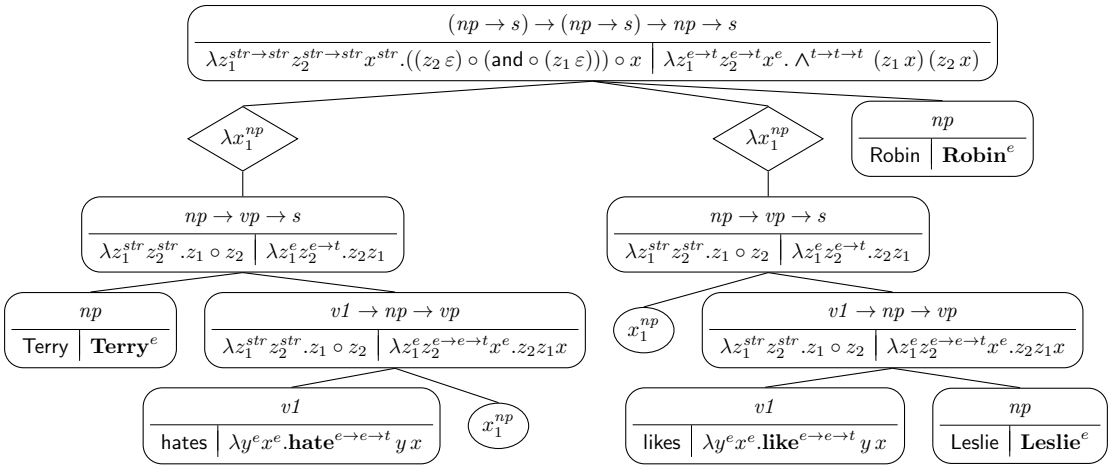


Figure 14: A derivation of Terry hates and likes Leslie Robin with the meaning "Terry hates Robin and Robin likes Leslie".

at its right periphery, but the ACG entry (11) cannot enforce that requirement. Since the requirement concerns the positions of the empty string symbol ε in the form component of generated pairs, we can try to use a syntactic feature to filter out unwanted sentences, like we did with violations of island constraints on *wh*-extraction.

We need to enforce that a gap occur at the right edge of each conjunct in a Right Node Raising construction. Since this is not a requirement on gaps in general, we have to distinguish gaps "bound" by the RNR entry for and from gaps bound by, e.g., *wh*-phrases. For this purpose, we introduce a special symbol ε^R for a right-peripheral gap, and mark the scope of its binder by $\mathbf{O}^R$. Note that these are markers we temporarily introduce to automatically discover a suitable feature mechanism; these are part of the specification, not the delivered product. The λ-term in the form dimension of the entry (11) is modified with these markers as follows:[24]

$$\begin{aligned}&(\,(np \to s) \to (np \to s) \to np \to s,\\ &\qquad \lambda z_1^{str \to str}\, z_2^{str \to str}\, x^{str}.(\mathbf{O}^R\,(z_2\,\varepsilon^R) \circ (\mathsf{and} \circ \mathbf{O}^R\,(z_1\,\varepsilon^R))) \circ x,\ \ldots\,)\end{aligned} \tag{13}$$

With this change, the derivation in Figure 14 now generates the following pair:

$$\begin{aligned}&(\,(\mathbf{O}^R\,(\mathsf{Terry} \circ (\mathsf{hates} \circ \varepsilon^R)) \circ (\mathsf{and} \circ \mathbf{O}^R\,(\varepsilon^R \circ (\mathsf{likes} \circ \mathsf{Leslie})))) \circ \mathsf{Robin},\\ &\qquad \wedge^{t \to t \to t}\ (\mathbf{like}^{e \to e \to t}\,\mathbf{Leslie}^e\,\mathbf{Robin}^e)\,(\mathbf{hate}^{e \to e \to t}\,\mathbf{Robin}^e\,\mathbf{Terry}^e)\,).\end{aligned}$$

We can express the constraint that needs to be satisfied as follows:

- Any occurrence of ε^R must be the rightmost leaf of a subtree marked by $\mathbf{O}^R$.

The deterministic bottom-up finite tree automaton that captures this constraint again has just two states, [RGAP 0] and [RGAP 1], which indicate the absence/presence of a right-peripheral gap. Its transitions are as follows:

$$\begin{aligned}\varepsilon^R &\to [\text{RGAP } 1],\\ a &\to [\text{RGAP } 0] \quad \text{for each terminal symbol } a,\\ [\text{RGAP } 0] \circ [\text{RGAP } 0] &\to [\text{RGAP } 0],\\ [\text{RGAP } 0] \circ [\text{RGAP } 1] &\to [\text{RGAP } 1],\\ \mathbf{O}^R\,[\text{RGAP } 1] &\to [\text{RGAP } 0].\end{aligned} \tag{14}$$

[24] We assume $\mathbf{O}^R$ binds stronger than $\circ$, so that $\mathbf{O}^R\,x \circ y$ means $(\mathbf{O}^R\,x) \circ y$.

$(\ np[0] \to vp[r] \to s[r],$	$\lambda z_1^{str}\, z_2^{str}.z_1 \circ z_2,$	$\lambda z_1^{e}\, z_2^{e\to t}.z_2 z_1$ $)$
$(\ v1[0] \to np[r] \to vp[r],$	$\lambda z_1^{str}\, z_2^{str}.z_1 \circ z_2,$	$\lambda z_1^{e\to e\to t}\, z_2^{e} x^{e}.z_1 z_2 x$ $)$
$(\ v2[0] \to s\_bar[r] \to vp[r],$	$\lambda z_1^{str}\, z_2^{str}.z_1 \circ z_2,$	$\lambda z_1^{t\to e\to t}\, z_2^{t} x^{e}.z_1 z_2 x$ $)$
$(\ v3[0] \to s\_int[r] \to vp[r]$	$\lambda z_1^{str}\, z_2^{str}.z_1 \circ z_2,$	$\lambda z_1^{q\to e\to t}\, z_2^{q} x^{e}.z_1 z_2 x$ $)$
$(\ s[r] \to s\_bar[r],$	$\lambda z_1^{str}.\mathsf{that} \circ z_1,$	$\lambda z_1^{t}.z_1$ $)$
$(\ s[r] \to s\_int[r],$	$\lambda z_1^{str}.\mathsf{whether} \circ z_1,$	$\lambda z_1^{t}.\mathbf{yn}^{t\to q}\, z_1$ $)$
$(\ np[0],$	$\mathsf{Leslie},$	$\mathbf{Leslie}^{e}$ $)$
$(\ np[0],$	$\mathsf{Robin},$	$\mathbf{Robin}^{e}$ $)$
$(\ np[0],$	$\mathsf{Terry},$	$\mathbf{Terry}^{e}$ $)$
$(\ v1[0],$	$\mathsf{hates},$	$\lambda y^{e} x^{e}.\mathbf{hate}^{e\to e\to t}\, y\, x$ $)$
$(\ v1[0],$	$\mathsf{likes},$	$\lambda y^{e} x^{e}.\mathbf{like}^{e\to e\to t}\, y\, x$ $)$
$(\ v2[0],$	$\mathsf{thinks},$	$\lambda y^{t} x^{e}.\mathbf{think}^{t\to e\to t}\, y\, x$ $)$
$(\ v2[0],$	$\mathsf{claims},$	$\lambda y^{t} x^{e}.\mathbf{claim}^{t\to e\to t}\, y\, x$ $)$
$(\ v3[0],$	$\mathsf{wonders},$	$\lambda y^{q} x^{e}.\mathbf{wonder}^{q\to e\to t}\, y\, x$ $)$
$(\ v3[0],$	$\mathsf{knows},$	$\lambda y^{q} x^{e}.\mathbf{know}^{q\to e\to t}\, y\, x$ $)$
$(\ (np[1] \to s[1]) \to$ $(np[1] \to s[1]) \to np[r] \to s[r],$	$\lambda z_1^{str\to str}\, z_2^{str\to str}\, x^{str}.$ $(\mathbf{O}^{R}\,(z_2\,\varepsilon^{R}) \circ (\mathsf{and} \circ \mathbf{O}^{R}\,(z_1\,\varepsilon^{R}))) \circ x,$	$\lambda z_1^{e\to t}\, z_2^{e\to t} x^{e}.\ \wedge^{t\to t\to t}\ (z_1\, x)\,(z_2\, x)$ $)$

Figure 15: An ACG with markers ε^R and $\mathbf{O}^R$ containing an entry for Right Node Raising. Here, $r \in \{0, 1\}$, and we write [0] and [1] for [RGAP 0] and [RGAP 1].

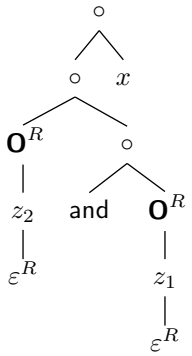
Using these transitions, we can calculate the features to assign to the occurrences of atomic types in the syntactic type of the entry (13). In order for a feature-specified syntactic type

$$(np[\textsc{rgap}\, r_1] \to s[\textsc{rgap}\, r_2]) \to (np[\textsc{rgap}\, r_3] \to s[\textsc{rgap}\, r_4]) \to np[\textsc{rgap}\, r_5] \to s[\textsc{rgap}\, r_6]$$

to be legitimate, the tree automaton (14) augmented with the transitions

$$\begin{aligned} z_1\,[\textsc{rgap}\, r_1] &\to [\textsc{rgap}\, r_2], \\ z_2\,[\textsc{rgap}\, r_3] &\to [\textsc{rgap}\, r_4], \\ x &\to [\textsc{rgap}\, r_5], \end{aligned}$$

must accept the tree



with final state [RGAP r_6]. We see that we must have $(r_1, r_2) = (r_3, r_4) = (1, 1)$, and $r_5 = r_6$, obtaining a new entry

$$(np[\textsc{rgap}\, 1] \to s[\textsc{rgap}\, 1]) \to (np[\textsc{rgap}\, 1] \to s[\textsc{rgap}\, 1]) \to np[\textsc{rgap}\, r] \to s[\textsc{rgap}\, r] \qquad \text{where } r \in \{0, 1\}.$$

Doing the same with the entries in Figure 3 results in the ACG in Figure 15. Note that the "final product" ACG is the result of removing $\mathbf{O}^R$ and changing ε^R to ε in these entries.

I leave it as an exercise for the reader to combine the fragment in Figure 15 with the one in Figure 12. This will require making a few decisions. You need to complete each of

$$
\begin{array}{ll}
(\, np, & \mathsf{Leslie} \quad) \\
(\, np \to np \to s, & \lambda z_1^{str} z_2^{str}. z_2 \circ (\mathsf{likes} \circ z_1) \quad) \\
(\, (np \to s) \to s, & \lambda z_1^{str \to str}.\mathsf{only\_she} \circ \mathbf{O}^L\,(z_1\, \varepsilon^L) \quad) \\
(\, (np \to s) \to s, & \lambda z_1^{str \to str}.\,\mathbf{O}^R\,(z_1\, \varepsilon^R) \circ \mathsf{only\_him} \quad)
\end{array}
$$

$$
\begin{array}{l}
(\, (np \to np \to s) \to (np \to np \to s) \to np \to np \to s, \\
\qquad \lambda z_1^{str \to str \to str} z_2^{str \to str \to str} z_3^{str} z_4^{str}. \\
\qquad z_4 \circ ((\mathbf{O}^R\,(\mathbf{O}^L\,(z_2\, \varepsilon^R\, \varepsilon^L)) \circ (\mathsf{and} \circ \mathbf{O}^R\,(\mathbf{O}^L\,(z_1\, \varepsilon^R\, \varepsilon^L)))) \circ z_3)\,)
\end{array}
$$

$$
\begin{array}{l}
(\, (((np \to s) \to s) \to s) \to (((np \to s) \to s) \to s) \to ((np \to s) \to s) \to s, \\
\quad \lambda z_1^{((str \to str) \to str) \to str} z_2^{((str \to str) \to str) \to str} z_3^{(str \to str) \to str}. \\
\quad (\mathbf{O}^R\,(z_2(\lambda y^{str \to str}.\,\mathbf{O}^R\,(y\, \varepsilon^R) \circ \varepsilon^R)) \circ (\mathsf{and} \circ \mathbf{O}^R\,(z_1(\lambda y^{str \to str}.\,\mathbf{O}^R\,(y\, \varepsilon^R) \circ \varepsilon^R)))) \circ \\
\qquad \mathbf{O}^L\,(z_3(\lambda x^{str}.\varepsilon^L \circ x))\,)
\end{array}
$$

Figure 16: An ACG translation of a Lambek grammar with markers. Not shown are entries for Robin and Terry similar to that for Leslie, and an entry for hates similar to that for likes.

the two tree automata with transitions for markers used in the other fragment and possibly add markings to the form components of some of the entries. Once these decisions are made, however, the rest of the grammar writing is automatic.

4.2 Translating Lambek grammars into ACGs

The use of the RGAP feature in the preceding section can be generalized to a method of translating an arbitrary Lambek grammar into an ACG (with finite-valued syntactic features). I only sketch the method here. The precise definition is found in the appendix (Section A).

Let us consider the following Lambek grammar as an example (we ignore the meaning dimension since it is simply preserved in the ACG):

$$
\begin{aligned}
np &: \mathsf{Leslie}, \mathsf{Robin}, \mathsf{Terry} \\
tv = (np\backslash s)/np &: \mathsf{likes}, \mathsf{hates} \\
subj = s/(np\backslash s) &: \mathsf{only\_she} \\
obj = (s/np)\backslash s &: \mathsf{only\_him} \\
(tv\backslash tv)/tv &: \mathsf{and} \\
((s/obj)\backslash(s/obj))/(s/obj) &: \mathsf{and}
\end{aligned} \tag{15}
$$

The last entry is used for sentences like Leslie likes and Robin hates only_him paired with the meaning "Leslie likes only him and Robin hates only him" (with "him" deictic). The other entry for and is for transitive verb coordination as in Leslie likes and hates Terry.

Translating this Lambek grammar into an ACG with markers in a way analogous to (13) gives the ACG in Figure 16 (the meaning dimension is omitted). Here, $\mathbf{O}^L$ and ε^L are the analogues of $\mathbf{O}^R$ and ε^R for left-peripheral gaps; in a well-formed sentence, any occurrence of ε^L must be the leftmost leaf of some subtree marked with $\mathbf{O}^L$. This constraint is expressed by the following tree automaton with two states, [LGAP 0] and [LGAP 1]:

$$
\begin{aligned}
\varepsilon^L &\to [\textsc{lgap}\,1], \\
\varepsilon^R &\to [\textsc{lgap}\,0], \\
a &\to [\textsc{lgap}\,0] \quad \text{for each terminal symbol } a,
\end{aligned}
$$

$$
\begin{array}{ll}
(\ np[0,0], & \mathsf{Leslie} \qquad\qquad) \\
(\ np[0,r] \to np[l,0] \to s[l,r], & \lambda z_1^{str} z_2^{str}.z_2 \circ (\mathsf{likes} \circ z_1) \qquad) \\
(\ (np[1,0] \to s[1,r]) \to s[0,r], & \lambda z_1^{str \to str}.\mathsf{only\_she} \circ (z_1\,\varepsilon) \qquad) \\
(\ (np[0,1] \to s[l,1]) \to s[l,0], & \lambda z_1^{str \to str}.(z_1\,\varepsilon) \circ \mathsf{only\_him} \qquad) \\
\multicolumn{2}{l}{(\ (np[0,1] \to np[1,0] \to s[1,1]) \to (np[0,1] \to np[1,0] \to s[1,1]) \to np[0,r] \to np[l,0] \to s[l,r],} \\
\multicolumn{2}{r}{\lambda z_1^{str \to str \to str} z_2^{str \to str \to str} z_3^{str} z_4^{str}.z_4 \circ (((z_2\,\varepsilon\,\varepsilon) \circ (\mathsf{and} \circ (z_1\,\varepsilon\,\varepsilon))) \circ z_3)\ ) } \\
\multicolumn{2}{l}{(\ (((np[0,1] \to s[l_1,1]) \to s[l_1,1]) \to s[0,1]) \to} \\
\multicolumn{2}{l}{\quad (((np[0,1] \to s[l_2,1]) \to s[l_2,1]) \to s[l_3,1]) \to ((np[0,r_1] \to s[1,r_1]) \to s[1,r_2]) \to s[l_3,r_2],} \\
\multicolumn{2}{l}{\qquad \lambda z_1^{((str \to str) \to str) \to str} z_2^{((str \to str) \to str) \to str} z_3^{(str \to str) \to str}.} \\
\multicolumn{2}{l}{\qquad\quad ((z_2(\lambda y^{str \to str}.(y\,\varepsilon) \circ \varepsilon)) \circ (\mathsf{and} \circ (z_1(\lambda y^{str \to str}.(y\,\varepsilon) \circ \varepsilon)))) \circ z_3(\lambda x^{str}.\varepsilon \circ x)\ )}
\end{array}
$$

Figure 17: An ACG translation of a Lambek grammar with features.

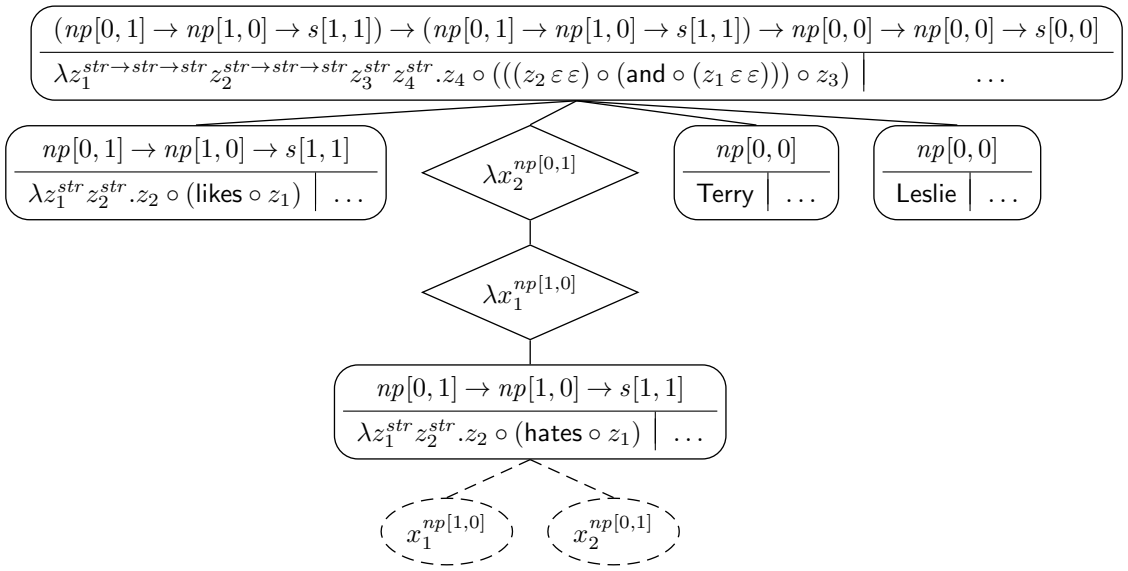


Figure 18: Failed derivation of Leslie likes and hates Terry with the meaning "Leslie likes Terry and Terry hates Leslie".

$$
\begin{aligned}
[\textsc{lgap}\ 0] \circ [\textsc{lgap}\ 0] &\to [\textsc{lgap}\ 0], \\
[\textsc{lgap}\ 1] \circ [\textsc{lgap}\ 0] &\to [\textsc{lgap}\ 1], \\
\mathbf{O}^L\,[\textsc{lgap}\ 1] &\to [\textsc{lgap}\ 0], \\
\mathbf{O}^R\,[\textsc{lgap}\ l] &\to [\textsc{lgap}\ l] \quad \text{for } l \in \{0,1\}.
\end{aligned}
$$

The tree automaton for the RGAP feature is similarly extended with transitions for ε^L and $\mathbf{O}^L$. These two automata independently determine how each feature is passed around in the syntactic types of the ACG entries. Writing $[l,r]$ for $[\textsc{lgap}\,l, \textsc{rgap}\,r]$, we get the final ACG in Figure 17. For example, this grammar blocks the derivation in Figure 18 because of feature mismatches between the entry for hates and its two arguments.

A further complication is necessary to deal with entries that introduce "binders" of two or more right-peripheral gaps (or left-peripheral gaps), like the following entry for and conjoining two ditransitive verbs:

$$((((np\backslash s)/np)/np)\backslash(((np\backslash s)/np)/np))/(((np\backslash s)/np)/np) : \mathsf{and}$$

This requires distinguishing two kinds of right-peripheral gaps, $\varepsilon_1^R, \varepsilon_2^R$, and the corresponding markers of scope, $\mathbf{O}_1^R, \mathbf{O}_2^R$. The full translation to ACGs with markers and the

associated tree automata are described in the appendix (Section A).

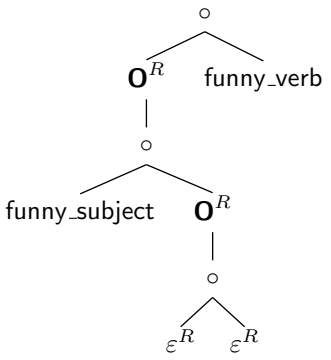
The ACG obtained by this method from an arbitrary Lambek grammar often undergenerates relative to the input Lambek grammar, since it does not allow nested uses of the / (or \) introduction rule. The following is the simplest kind of example that gives rise to this phenomenon:

$$s/(s/np) : \mathsf{funny\_subject}$$
$$(s/(s/np))\backslash s : \mathsf{funny\_verb}$$

The "η-expanded" form of the Lambek derivation for the sentence funny_subject funny_verb looks as follows:[25]

$$\dfrac{\mathsf{funny\_subject} \quad \dfrac{\dfrac{[s/np]^2 \quad [np]^1}{s}/E}{s/np}/I,1}{\dfrac{s}{s/(s/np)}/I,2} \qquad \mathsf{funny\_verb}}{s} \quad \begin{array}{l} /E \\ \\ /E \\ \\ \backslash E \end{array} \qquad \begin{array}{l} \\ \\ \\ \\ (s/(s/np))\backslash s \end{array}$$

The corresponding derivation of the ACG with markers gives the tree



which is not accepted by the tree automaton regulating the positions of ε^R. However, a situation like this does not seem to arise in practice in Lambek grammars for natural language, where two right-peripheral gaps "bound" by distinct binders (in this case $s/(s/np)$ and $(s/(s/np))\backslash s$) occur unbound in the same sub-derivation (in this case the derivation ending in the topmost occurrence of s).

Note that an analogous case involving both kinds of slashes poses no problem:

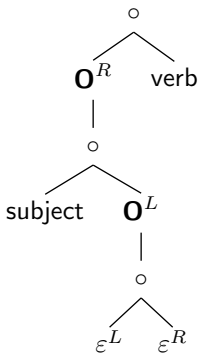
$$s/(np\backslash s) : \mathsf{subject}$$
$$(s/(np\backslash s))\backslash s : \mathsf{verb}$$

The Lambek derivation (in η-expanded form) for subject verb is

$$\dfrac{\mathsf{subject} \quad \dfrac{\dfrac{[np]^1 \quad [np\backslash s]^2}{s}\backslash E}{np\backslash s}\backslash I,1}{\dfrac{s}{s/(np\backslash s)}/I,2} \qquad \mathsf{verb}}{s} \quad \begin{array}{l} /E \\ \\ /E \\ \\ \backslash E \end{array} \qquad \begin{array}{l} \\ \\ \\ \\ (s/(np\backslash s))\backslash s \end{array}$$

[25] I'm displaying a Lambek derivation in η-expanded form because my method of translation effectively works on the η-expanded form of the one-line derivation consisting of the type assigned to a terminal. Of course, there is no need to restrict our attention to such derivations when discussing a Lambek grammar.

The corresponding ACG derivation will give the tree



which is accepted by the two automata (for ε^R and for ε^L).

Another hypothetical situation where the proposed translation will fail for the same reason is when a Lambek grammar has entries like

$$\begin{aligned}(np\backslash s)/(vp\_inf/np) &: \mathsf{is\_hard}\\ vp\_inf/(np\backslash s) &: \mathsf{to}\\ ((np\backslash s)/np)/np &: \mathsf{teach}\\ s\_int/(s/np) &: \mathsf{who}, \mathsf{which\_language}\\ (np\backslash s)/s\_int &: \mathsf{wonders}\\ np &: \mathsf{Leslie}, \mathsf{Robin}, \mathsf{Haskell}\end{aligned}$$

This grammar generates Leslie wonders who$_1$ Haskell$_2$ is_hard to teach t_1 t_2 (with crossed dependencies), but not Leslie wonders which_language$_2$ Robin$_1$ is_hard to teach t_1 t_2 (with nested dependencies) (Figure 19). (The ACG translation rejects both sentences.) Of course, a Lambek grammar like this does not even remotely resemble a correct description of a fragment of English; neither *wh*-movement nor *tough*-movement is restricted to right periphery, and if anything, the word order exhibiting nested dependencies often seems to be the preferred one.[26] In general, it seems to me that this peculiar ability of Lambek's directional slashes to enforce crossed, as opposed to nested, dependencies between binders and gaps is never utilized in descriptions of natural language.

The present automatic method of translating a Lambek grammar into an ACG seems to me to provide a good enough approximation of the input Lambek grammar for the purpose of linguistic description. I am not, however, advocating to build an ACG-based linguistic theory on the basis of this translation. When you are dealing with an expression whose meaning is a function, there is always a choice between analyzing its form as a λ-abstract or as a simple string. If you choose the latter, the syntactic type of the expression must be atomic. If you choose the former, there is a further choice between giving the expression a functional syntactic type or an atomic syntactic type. In the largely CFG-based analysis of Sections 2.1, 2.3, 3, and 4.1, all standard constituents except that, whether, and and were analyzed as simple strings in the form dimension and given atomic syntactic types, while non-standard constituents and constituents with gaps were analyzed as functions in the form dimension and given functional syntactic types.[27] Alternatively, you could give non-standard constituents atomic types while analyzing them as functions in the form

[26] Presumably, the particular sentences at hand are both ungrammatical in English because they involve extraction of the first object in a double object (dative shift) construction.

[27] Of course, it is also easy to treat that, whether, and and like all other standard constituents.

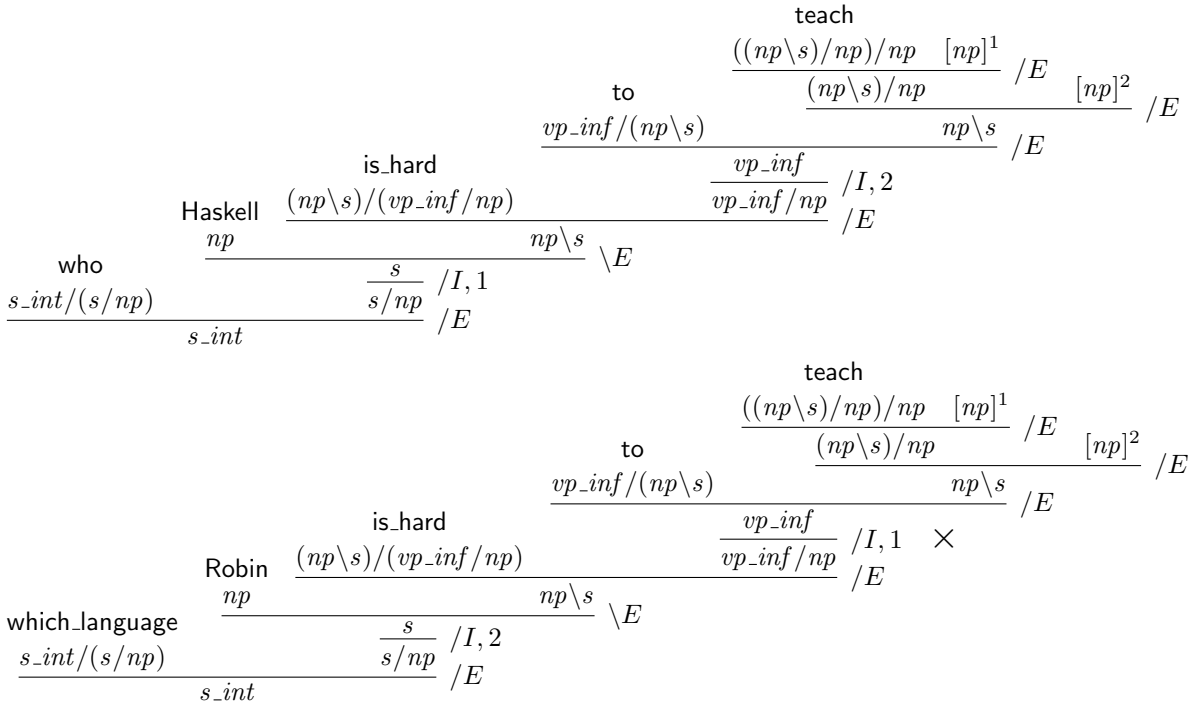


Figure 19: A Lambek derivation with crossed dependencies (above) and a failed derivation with nested dependencies (below).

dimension, on the model of the TAG-style treatment of *wh*-extraction in Section 2.2. Or you could analyze some standard constituents as functions in the form dimension and give them functional types, as in the translation of Lambek grammars. These choices (strings vs. functions, atomic vs. functional syntactic types) need not be made uniformly; the decision can be made on a case-by-case basis, taking into account properties of a particular construction at hand. Even when you decide to adopt a Lambek-style analysis of a certain construction, the output of the automatic translation should be regarded as a rough prototype which you can try to refine "manually" to suit the particular properties of the construction in question. The output of my translation method applied to an existing Lambek grammar (or a "hybrid" extension thereof, see below) should be viewed as a kind of baseline—the *worst* you could do in a linguistic theory based on the ACG formalism.

4.3 Gapping

The method of Section 4.2 is also applicable to Kubota and Levine's (2014a; 2014b) formalism of hybrid type-logical grammars, which is a kind of amalgam of Lambek grammars and ACGs. Hybrid type-logical grammars use both the directional slashes of the Lambek calculus and the "non-directional" implication from the simply typed lambda calculus. A *hybrid type* is either a directional type of the Lambek calculus or of the form $A \to B$, where A and B are hybrid types. For example, Kubota and Levine (2014b) postulated the following entry for the conjunction and in the Gapping construction, where $tv = (np\backslash s)/np$:[28]

$$\begin{aligned}&(\,(tv \to s) \to (tv \to s) \to tv \to s,\ \lambda z_1^{str\to str} z_2^{str\to str} z_3^{str}.(z_2 z_3) \circ (\mathsf{and} \circ (z_1\,\varepsilon)),\\ &\qquad \lambda z_1^{(e\to e\to t)\to t} z_2^{(e\to e\to t)\to t} z_3^{e\to e\to t}.\ \wedge^{t\to t\to t}\ (z_1(\lambda y^e x^e.z_3 y x))\,(z_2(\lambda y^e x^e.z_3 y x))\,).\end{aligned}$$

[28]This is actually an instance of a more general schema where, instead of tv, one has any Lambek type of the form $Y_0\backslash s/Y_1/\ldots/Y_n$ or $s/Y_0/Y_1/\ldots/Y_n$ with $n \geq 1$.

This entry together with the entries of the Lambek grammar in (15) license a hybrid derivation of the form-meaning pair

$$\begin{aligned}&(\,(\mathsf{Leslie} \circ (\mathsf{likes} \circ \mathsf{Robin})) \circ (\mathsf{and} \circ (\mathsf{Robin} \circ (\varepsilon \circ \mathsf{Terry}))),\\ &\qquad\qquad \wedge^{t\to t\to t}\ (\mathbf{likes\ Robin\ Leslie})\ (\mathbf{likes\ Terry\ Robin})\,).\end{aligned}$$

A naive translation of this fragment into an ACG, with the following entry for and, would make the sentence Leslie likes Robin and Robin Terry ambiguous between the actual reading "Leslie likes Robin and Robin likes Terry" and the reading "Robin likes Leslie and Terry likes Robin":[29]

$$\begin{aligned}&(\,((np \to np \to s) \to s) \to ((np \to np \to s) \to s) \to (np \to np \to s) \to s,\\ &\quad \lambda z_1^{(str\to str\to str)\to str} z_2^{(str\to str\to str)\to str} z_3^{str\to str\to str}.\\ &\qquad (z_2(\lambda y^{str} x^{str}.x \circ ((z_3\,\varepsilon\,\varepsilon) \circ y))) \circ (\mathsf{and} \circ (z_1(\lambda y^{str} x^{str}.x \circ (\varepsilon \circ y)))),\\ &\quad \lambda z_1^{(e\to e\to t)\to t} z_2^{(e\to e\to t)\to t} z_3^{e\to e\to t}.\ \wedge^{t\to t\to t}\ (z_1(\lambda y^e x^e.z_3 yx))\,(z_2(\lambda y^e x^e.z_3 yx))\,).\end{aligned}$$

Again, we can use the LGAP and RGAP features to deal with this problem, without abandoning the style of analysis employed by Kubota and Levine (2014b):[30]

$$\begin{aligned}&(\ ((np[0,r_1] \to np[l_1,0] \to s[l_1,r_1]) \to s[0,r]) \to\\ &\quad ((np[0,r_2] \to np[l_2,0] \to s[l_2,r_2]) \to s[l,0]) \to (np[0,1] \to np[1,0] \to s[1,1]) \to s[l,r],\\ &\quad \lambda z_1^{(str\to str\to str)\to str} z_2^{(str\to str\to str)\to str} z_3^{str\to str\to str}.\\ &\qquad (z_2(\lambda y^{str} x^{str}.x \circ (\mathbf{O}^R\,(\mathbf{O}^L\,(z_3\,\varepsilon^R\,\varepsilon^L)) \circ y))) \circ (\mathsf{and} \circ (z_1(\lambda y^{str} x^{str}.x \circ (\varepsilon \circ y)))),\\ &\qquad\quad \lambda z_1^{(e\to e\to t)\to t} z_2^{(e\to e\to t)\to t} z_3^{e\to e\to t}.\wedge^{t\to t\to t}\,(z_1(\lambda y^e x^e.z_3 yx))\,(z_2(\lambda y^e x^e.z_3 yx))\,).\end{aligned}$$

The way overgeneration is blocked here (Figure 20) is entirely analogous to the case of transitive verb conjunction (Figure 18).

4.4 A Closer Look at Gapping

Kubota and Levine's (2014b) decision to exclusively use Lambek types like $tv = (np\backslash s)/np$ as the possible categories of the gap (the elided material in the second conjunct of Gapping) is puzzling. It has been widely recognized that the gap can be discontiguous:

(16) Max seemed to be trying to force Ted to leave the room, and Walt [~~seemed to be trying to force~~] Ira [~~to leave the room~~] (Jackendoff, 1971, p. 25)

(17) Arizona elected Goldwater Senator, and Pennsylvania [~~elected~~] Schweiker [~~Senator~~] (Jackendoff, 1971, p. 24)

(18) Jack begged Elsie to get married, and Wilfred [~~begged~~] Phoebe [~~to get married~~] (Jackendoff, 1971, p. 24)

(19) Max wanted Ted to persuade Alex to get lost and [~~Max wanted~~] Walt [~~to persuade~~] Ira [~~to get lost~~] (Hankamer, 1973, pp. 26–27)

(20) John took Harry to the movies, and Bill [~~took~~] Mike [~~to the movies~~] (Sag, 1976, p. 218)

[29] There is an alternative translation considered by Moot (2014) and Kubota and Levine (2014b) which gives the sentence the reading "Leslie likes Robin and Terry likes Robin". This entry is shown in Figure 21.

[30] I'm showing the grammar with markers for the convenience of the reader; the markers are not actually present in the output ACG of the translation.

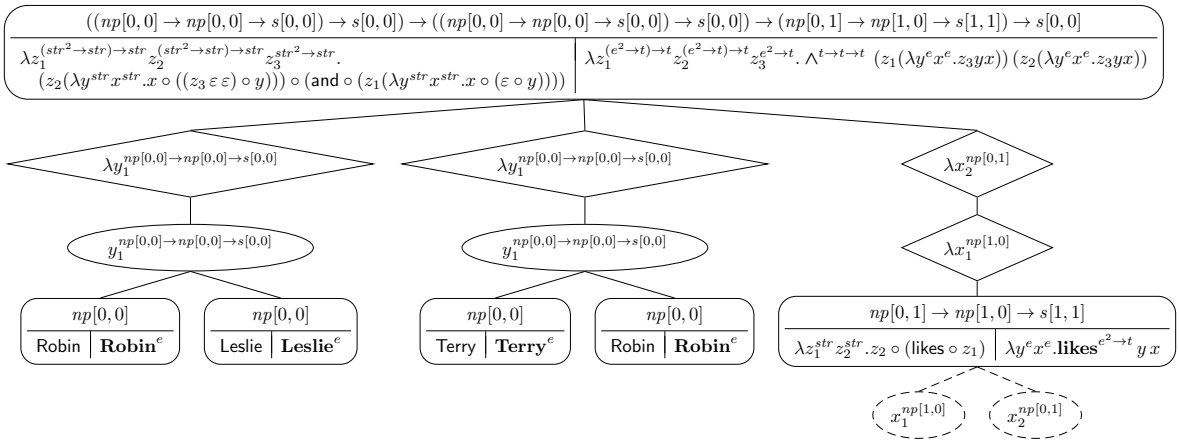


Figure 20: Failed derivation of Leslie likes Robin and Robin Terry with the meaning "Robin likes Leslie and Terry likes Robin".

(21) John persuaded Dr. Thomas to examine Mary, and Bill [~~persuaded~~] Dr. Jones [~~to examine Mary~~] (Sag, 1976, p. 225)

(22) Joe covered the floor with red paint, and Alice [~~covered~~] the walls [~~with red paint~~] (Neijt, 1980, p. 79)

(23) Joe painted his boat red, and Alice [~~painted~~] her car [~~red~~] (Neijt, 1980, p. 79)

(24) Some people want all doors to open to the left and others [~~want~~] all windows [~~to open to the left~~] (Neijt, 1980, p. 160)

In all of these examples, there is elided material to the right of the second remnant, so it seems that a hybrid type-logical grammar would need to assign the gap either a hybrid type $np \to (np \backslash s)$ or a simple type $np \to np \to s$.[31]

Examples like the following seem to require $np \to np \to s$ as the category of the gap:

(25) Max ordered Ted to persuade Alex to get lost and [~~Max ordered~~] Walt [~~to persuade~~] Ira [~~to get lost~~]

(26) I asked Peter to take Susan home, and [~~I asked~~] John [~~to take~~] Wendy [~~home~~]

(27) Rarely does John call Mary at home, and [~~rarely does~~] Mary [~~call~~] John [~~at home~~]

This puts hybrid type-logical grammars in the same quandary that plagued the naive ACG translation of a Lambek grammar (Section 4.3). Assuming hybrid entries in Figure 21, the sentence I asked Peter to take Susan home and John Wendy would come out as ambiguous between the reading indicated in (26) and the reading "I asked Peter to take Susan home, and I asked Wendy to take John home" (among other readings).

Sentences like (25) and (26) seem to be generally acceptable. If so, the true generalization about the word order in Gapping may be the following:

(28) In the first conjunct of Gapping, the correspondent of the first remnant must precede the correspondent of the second remnant.

It should be clear that (28) can be expressed as a regular constraint, if we mark the positions of the correspondents in the first conjunct of Gapping:

[31] In Hankamer's example (19), the hybrid type $np \to (np \backslash s)$ will work under the analysis where Gapping occurs in the infinitival clause [Ted to persuade Alex to get lost and Walt [~~to persuade~~] Ira [~~to get lost~~]].

np	: I	: $\mathbf{me}^{e}$
np	: Peter	: $\mathbf{Peter}^{e}$
np	: Susan	: $\mathbf{Susan}^{e}$
np	: John	: $\mathbf{John}^{e}$
np	: Wendy	: $\mathbf{Wendy}^{e}$
$vp\_inf/(np\backslash s)$	: to	: $\lambda y^{e\to t} x^{e}.yx$
$(np\backslash s)/vp\_inf/np$	: asked	: $\lambda y^{e} z^{e\to t} x^{e}.\mathbf{ask}^{e\to(e\to t)\to e\to t}\, y\,(zy)\,x$
$(np\backslash s)/pp/np$	: take	: $\lambda y^{e} z^{e} x^{e}.\mathbf{take}^{e\to e\to t}\, z\, y\, x$
pp	: home	: $\mathbf{home}^{e}$
$((np^2 \to s) \to s) \to$ $((np^2 \to s) \to s) \to$ $(np^2 \to s) \to s$	: $\lambda z_1^{(str^2\to str)\to str}$ $z_2^{(str^2\to str)\to str} z_3^{str^2\to str}.$ $(z_2(\lambda y^{str} x^{str}.z_3 yx)) \circ$ $(\mathsf{and} \circ (z_1(\lambda y^{str} x^{str}.x \circ (\varepsilon \circ y))))$	: $\lambda z_1^{(e^2\to t)\to t} z_2^{(e^2\to t)\to t} z_3^{e^2\to t}.$ $\wedge^{t\to t\to t}\,(z_1(\lambda y^{e} x^{e}.z_3 yx))$ $(z_2(\lambda y^{e} x^{e}.z_3 yx))$

Figure 21: A hybrid type-logical grammar for discontiguous Gapping.

$$\lambda z_1^{(str\to str\to str)\to str} z_2^{(str\to str\to str)\to str} z_3^{str\to str\to str}.$$
$$(z_2(\lambda y^{str} x^{str}.\,\mathbf{O}^{<}\,(z_3(\mathbf{C}_2\,y)(\mathbf{C}_1\,x)))) \circ (\mathsf{and} \circ (z_1(\lambda y^{str} x^{str}.x \circ (\varepsilon \circ y)))). \quad (29)$$

One can then capture this regular constraint with a syntactic feature, again using the general method of Kanazawa (2006). The resulting entry would work for all cases of Gapping in which the correspondents/remnants are two noun phrases. Note that the method is applicable no matter what regular constraint governs the positions of the correspondents/remnants. If, as has often been argued (Kuno, 1976; Neijt, 1980; Coppock, 2001; Johnson, 2014), the relative positions of the correspondents/remnants obey some (but perhaps not all) of the island constraints governing *wh*-extraction, those constraints can also be captured by the syntactic feature, as long as they are regular.[32]

Actually, even the status of the generalization (28) may be open to question. Sag et al. (1985) used the following example to question Hudson's (1982) claim that in Gapping "the order of constituents in the second conjunct ... parallel the order of the corresponding constituents in the first conjunct" (Hudson, 1982, p. 548):

(30) A policeman walked in at 11, and at 12, a fireman (Sag et al., 1985, p. 158)

The next examples are from Hankamer (1979, pp. 151–152):

(31) a. The beans, Harry cooked, and the potatoes, Henry
b. The beans, Harry cooked, and Henry, the potatoes

The following word order is also perfectly acceptable, given the right context:[33]

(32) Harry cooked the beans, and the potatoes, Henry

It seems likely that the reverse word order in (30), (31b), and (32) is the result of interaction between Gapping and a separate process of Topicalization. Hankamer (1979),

[32]In addition to constraints that hold of *wh*-extraction, it has been argued that Gapping obeys the Tensed S Condition (Neijt, 1980).

[33]For example, (32) is natural in the following dialogue:

(i) A. Gee, the beans and the potatoes are good! Did Harry cook them again?
B. No. Today, Harry cooked the beans, and the potatoes, Henry.

based on (31a) and (31b), concluded that the transformation of Topicalization follows Gapping. In terms of an analysis along the line of Kubota and Levine (2014b), this means that the first and the second arguments of the Gapping entry for and can independently undergo Topicalization.[34] If so, the implementation suggested above of the generalization (28) in terms of a syntactic feature can be maintained. Whether or not this is on the right track, data like (30), (31a), (31b), and (32) show that one needs an account of the effect of Topicalization on possible intonations and topic-focus structures of utterances in order to tell whether a given analysis of Gapping (and Topicalization) generates ungrammatical form-meaning pairs.[35]

To end this inconclusive discussion, no matter what constraint the grammar should ultimately impose on the order of the correspondents/remnants in Gapping, it seems fairly clear that Lambek's directional slashes are not the right tool for this purpose.

5 Conclusion

I would like to end by answering some questions that might be raised against this work.

What about the Pentus construction?

> *Pentus (1993; 1997) showed that a Lambek grammar can be translated into an equivalent context-free grammar, and Kanazawa and Salvati (2013) showed that the construction preserves the string-meaning relation. By de Groote's (2001) encoding, this means that every Lambek grammar has an equivalent ACG. Why do you need another translation which is only an approximation?*

Pentus's construction drastically changes the derivations of the input Lambek grammar. The ACG obtained this way is a particularly simple kind of second-order ACG and does not use λ-abstraction in the derivation. Along with Moot (2014) and Kubota and Levine (2014a,b), I am interested in the possibility of translating Lambek grammars into ACGs in such a way that structures of derivations are preserved.[36]

[34] In ACGs/hybrid type-logical grammars, Topicalization can be handled in a similar way to *wh*-extraction.

[35] Winkler (2005, p. 192) proposed the following principle:

> Contrastive Topic and Focus Principle:
> In gapping, the first remnant is a contrastive topic, the second remnant a contrastive focus.
> The gapped elements must be given.

This does not always seem to be the case, however:

(i) (Did Gwendolyn play poker and Alan canasta?)
No, Alan played poker, and Gwendolyn canasta. (Sag, 1976, p. 192)

[36] This is not to say that Pentus's conversion of Lambek grammars to CFGs is necessarily without linguistic interest. Moot (2014, p. 59) stresses that Pentus's construction brings about an exponential blowup in the size of the grammar. While this is literally true of the CFG defined by Pentus (1997, Theorem 2), this CFG contains a lot of useless nonterminals and superfluous rules, and should not be the focus of our attention when we are interested in grammar size. We can obtain a much more compact CFG by adhering to Pentus's proof closely (see Pentus, 1997, Lemma 7) and eliminating some obvious redundancies. It is also worth pointing out that even if it turns out that Pentus's construction must lead to an exponentially larger grammar in the worst case, this will not mean that exponential blowup is inevitable. A particular conversion method can only establish an upper bound on the increase in size when going from one formalism to another; establishing that such an upper bound is the best possible one would require an entirely different kind of proof. Note that the NP-completeness of derivability in the product-free Lambek calculus (Savateev, 2012) implies that any method of converting Lambek grammars to CFGs must have non-polynomial time complexity, but says nothing about the size of the output.

What are syntactic features?

> *Doesn't addition of syntactic features require extension of the type theory behind ACGs?*

I prefer to think of features as abbreviatory devices outside of the formalism of ACGs (see the clarification at the end of Section 1). An entry with variables as feature values abbreviates all of its instantiations. Since I only consider finite-valued features, this means that I do not go beyond the expressive power of ACGs as originally defined by de Groote (2001). An alternative is to embrace features as part of the formalism, using a type-theoretic extension of ACGs (de Groote and Maarek, 2007). In the particular case of finite-valued features, I see little advantage in doing so.

How many features are needed?

> *You have used one feature to express island constraints on wh-extraction, one feature to handle right-peripheral extraction in Right Node Raising, another feature to mimic Lambek's backslash, and yet another to constrain the relative order of the correspondents in the first conjunct of Gapping. If you need a new feature every time you want to express a regular constraint, don't you end up with an extremely large number of features?*

Beside these examples, I do not expect an ACG-based linguistic theory to need many more features like them. I consider the last three features to be natural ones for constraining the positions of the gaps (right- and left-peripherality and relative order between two gaps[37]). Note that these features are capable of capturing any Boolean combination of the relevant regular constraints; if, for instance, it turns out to be desirable to restrict the gap in a certain construction to a *non-right-peripheral* position, one can easily do so without introducing a new feature.[38]

Doesn't adding syntactic features result in an undesirable increase in the size of the grammar?

> *If you use variables as feature values to abbreviate grammar entries, the size of the grammar will be very large compared to the abbreviated notation. Isn't that a problem?*

I assume that a grammar is represented in the linguistic theory (or in the brain) in compressed form, using features and a host of other abbreviatory devices. The size of the uncompressed grammar may become an issue if it is assumed that the compressed representation needs to be fully decompressed in order to be put to use. This is a rather implausible assumption. Even if an individual entry needs to be "multiplied out" with concrete feature values in order to be used, it is reasonable to assume that only a limited number of entries are activated on any single occasion of language use (like sentence comprehension or production). It is hard to be concrete without a good model of parsing and generation (which we only have for second-order ACGs (Kanazawa, 2007, 2011)), but in terms of de Groote's (2015) parsing schema for higher-order ACGs, it is a trivial exercise to accommodate features like the ones we've been discussing, without using any decompression.

[37] In the marked ACG entry (29), the markers $\mathbf{C}_1$ and $\mathbf{C}_2$ effectively mark the positions of the gaps in the third argument of the entry.

[38] Depending on the style of analysis, you may not even need all of these features; if you adopt a CFG-based style of analysis of English, you may not ever need the LGAP feature, since, for example, the atomic type vp takes the place of $np\backslash s$.

Don't syntactic features increase the computational complexity of the linguistic theory?

> *Granted that the decompressed grammar need not be explicitly stored even temporarily, doesn't the presence of features still increase the computational complexity of parsing, relative to the complexity of parsing with ACGs given in uncompressed form?*

This question is a little hard to make sense of, since the time complexity of parsing with the decompressed grammar certainly places an upper bound on the time complexity of parsing with the compressed grammar, modulo the overhead incurred by decompression. The suggestion is that the dependence of the time complexity of parsing on the size of the grammar will be significantly higher in the presence of features than in their absence. In general, allowing an arbitrary number of finite-valued features changes a tractable parsing problem into an intractable one (Barton et al., 1987, Chapter 3), but here we are not dealing with a general case. What we have are a small number of features that express deterministic bottom-up finite tree automata. It is not hard to see that such features do not increase the degree of global ambiguity; the fact that they express deterministic bottom-up finite tree automata means that any derivation tree that is legitimate except for possible feature mismatches allows at most one assignment of feature values. How much *local ambiguity* increases, which is the real cause of increased time complexity, depends on the parsing algorithm. Again, since we do not have a good model of parsing for ACGs in general, it is hard to be concrete, but it is likely that determinism will help in reducing local ambiguity as well. In terms of second-order ACGs, whose computational complexity properties are well understood, the universal recognition problem is already PSPACE-hard, and one needs to place a bound on the size of $\sigma(\alpha)$ (for syntactic types α of grammar entries) to make it tractable (Kanazawa, 2011).[39] When the size of $\sigma(\alpha)$ and the number of features are both bounded, even explicitly decompressing the grammar only leads to a polynomial increase in grammar size, so the problem remains tractable in the presence of features.

Why not go GPSG and let the feature mechanism handle extraction itself?

> *The way the* GAP *feature behaves is similar to the SLASH feature in GPSG (Gazdar et al., 1985). If such a feature is needed, why don't you just use the SLASH feature instead of λ-abstraction? Likewise, left- and right-peripheral extraction may be handled by LSLASH and RSLASH features.*

I don't have a good answer to this question. Some people think λ-abstraction in derivations is problematic, and this feature of ACGs should be abandoned (Kiselyov, 2015). One point in favor of λ-abstraction in derivations may be that it allows an ACG to be written in a style very close to common practices of linguists. For example, it seems straightforward to express in ACGs not only Montague's (1973) fragment but also much of what's found in Heim and Kratzer's (1998) textbook, using the general style of CFG plus extraction. Features can be conveniently left aside when they are not important.

Why not go hybrid instead of adding syntactic features?

> *Even if ACGs can simulate Lambek's directional slashes with syntactic features, isn't using Lambek's slashes simpler and more elegant? Why are you trying to replace them with crummy features?*

[39] Compare a similar bound that Barton et al. (1987, Chapter 9) placed on the length of rules in R-GPSG.

I consider ACGs to possess a kind of a priori appeal that hybrid type-logical grammars lack. The syntactic features considered in this paper are the most direct expressions of the relevant constraints on the positions of gaps. In contrast, there are natural constraints on the positions of gaps that cannot be captured by Lambek's slashes (see Section 4.4), and I hinted in Section 4.2 that the power of Lambek's slashes is very much underutilized in descriptions of natural language. Lambek's slashes seem to me to be neither sufficient nor necessary.

Isn't the ACG formalism too powerful?

> *ACGs are capable of incorporating in the form of a syntactic feature an arbitrary stipulation that is expressible as a regular constraint. Doesn't that make the formalism of ACGs too powerful and devoid of explanatory power?*

That is exactly right, and the same can be said of any reasonable grammar formalism. The relevant property, closure under intersection with regular sets, is intimately tied to tabular parsing (Lang, 1994), and seems to be an essential property of any mathematically well-behaved grammar formalism. Any explanatory power that a linguistic theory may have must come from a combination of the grammar formalism employed by the theory—which is neutral with respect to different applications—and specifically *linguistic* principles espoused by the theory.

References

Abeillé, Anne and Owen Rambow, eds. 2000. *Tree Adjoining Grammars*. Stanford, California: CSLI Publications.

Barton, G. Edward, Robert Berwick, and Eric Sven Ristad. 1987. *Computational Complexity and Natural Language*. Cambridge, Mass.: MIT Press.

Chaves, Rui P. 2012. On the grammar of extraction and coordination. *Natural Language and Linguistic Theory* 30:465–512.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. Cambridge, Mass.: MIT Press.

Comon, Hubert, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. 2008. *Tree Automata Techniques and Applications*. Available at `http://tata.gforge.inria.fr`. November 18, 2008.

Coppock, Elizabeth. 2001. Gapping: In defense of deletion. In *CLS 37: The Main Session*, 133–148. Chicago: Chicago Linguistic Society.

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

de Groote, Philippe. 2002. Tree-adjoining grammars as abstract categorial grammars. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+6)*, 145–150. Universitá di Venezia.

de Groote, Philippe. 2015. Abstract categorial grammar as linear logic programming. In M. Kuhlmann, M. Kanazawa, and G. Kobele, eds., *Proceedings of the 14th Meeting on the Mathematics of Language*. The Association for Computational Linguistics.

de Groote, Philippe and Sarah Maarek. 2007. Type-theoretic extensions of abstract categorial grammars. In R. Muskens, ed., *Workshop on New Directions in Type-theoretic Grammars*, 19–30.

de Groote, Philippe and Sylvain Pogodalla. 2004. On the expressive power of abstract categorial grammars: Representing context-free formalisms. *Journal of Logic, Language and Information* 13:421–438.

Frank, Robert. 2002. *Phrase Structure Composition and Syntactic Dependencies*. Cambridge, Mass.: MIT Press.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum, and Ivan A. Sag. 1985. *Generalized Phrase Structure Grammar*. Cambridge, Mass.: Harvard University Press.

Hankamer, Jorge. 1973. Unacceptable ambiguity. *Linguistic Inquiry* 4:17–68.

Hankamer, Jorge. 1979. *Deletion in Coordinate Structures*. Outstanding Dissertations in Linguistics. New York: Garland.

Heim, Irene and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Oxford: Blackwell.

Hofmeister, Philip and Ivan Sag. 2010. Cognitive constraints and island effects. *Language* 86:365–415.

Hudson, Richard. 1982. Incomplete conjuncts. *Linguistic Inquiry* 13:547–550.

Jackendoff, Ray S. 1971. Gapping and related rules. *Linguistic Inquiry* 2:21–35.

Jacobson, Pauline. 2014. *Compositional Semantics*. Oxford: Oxford University Press.

Johnson, Kyle. 2014. Gapping. Manuscript available at `http://people.umass.edu/kbj/homepage/Content/gapping.pdf`.

Joshi, Aravind K. and Yves Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, eds., *Handbook of Formal Languages*, vol. 3, 69–123. Berlin: Springer.

Kanazawa, Makoto. 2006. Abstract families of abstract categorial languages. *Electronic Notes in Theoretical Computer Science* 165:65–80. Proceedings of the 13th Workshop on Logic, Language, Information and Computation (WoLLIC 2006).

Kanazawa, Makoto. 2007. Parsing and generation as Datalog queries. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, 176–183. Prague, Czech Republic.

Kanazawa, Makoto. 2009. Second-order abstract categorial grammars. Lecture notes for a course at ESSLLI 2009, available at `http://research.nii.ac.jp/~kanazawa/publications/esslli2009_lectures.pdf`.

Kanazawa, Makoto. 2011. Parsing and generation as Datalog query evaluation. Manuscript under review, available at `http://research.nii.ac.jp/~kanazawa/pagadqe.pdf`.

Kanazawa, Makoto. 2014. Almost affine lambda terms. In A. Indrzejczak, J. Kaczmarek, and M. Zawidzki, eds., *Trends in Logic XIII*, 131–148. Łódź: Łódź University Press.

Kanazawa, Makoto and Sylvain Salvati. 2013. The string-meaning relations definable by Lambek grammars and context-free grammars. In G. Morrill and M.-J. Nederhof, eds., *Formal Grammar, FG 2012/2013*, vol. 8036 of *Lecture Notes in Computer Science*, 191–208. Berlin: Springer.

Kanazawa, Makoto and Ryo Yoshinaka. 2005. Lexicalization of second-order ACGs. NII Technical Report NII-2005-012E, National Institute of Informatics, Tokyo.

Kiselyov, Oleg. 2015. Applicative abstract categorial grammars. In M. Kanazawa, L. Moss, and V. de Paiva, eds., *Proceedings of the Third Workshop on Natural Language and Computer Science*.

Kracht, Marcus. 2003. *The Mathematics of Language*. Berlin: Mouton de Gruyter.

Kroch, Anthony S. 1987. Unbounded dependencies and subjacency in a tree adjoining grammar. In A. Manaster-Ramer, ed., *The Mathematics of Language*, 143–172. Philadelphia: John Benjamins.

Kubota, Yusuke and Robert Levine. 2014a. Against ellipsis: Arguments for the direct licensing of 'non-canonical' coordinations. To appear in *Linguistics and Philosophy*.

Kubota, Yusuke and Robert Levine. 2014b. Gapping as hypothetical reasoning. To appear in *Natural Language and Linguistic Theory*.

Kuno, Susumu. 1976. Gapping: A functional analysis. *Linguistic Inquiry* 7:300–318.

Lang, Bernard. 1994. Recognition can be harder than parsing. *Computational Intelligence* 10(4):486–494.

Lazić, Ranko and Sylvain Schmitz. 2014. Non-elementary complexities for branching VASS, MELL, and extensions. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 61:1–61:10. New York, NY, USA: ACM. ISBN 978-1-4503-2886-9.

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

Moot, Richard. 2014. Hybrid type-logical grammars, first-order linear logic and the descriptive inadequacy of lambda grammars. `https://hal.archives-ouvertes.fr/hal-00996724`.

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

Neijt, Anneke. 1980. *Gapping*. Dordrecht: Foris Publications, second revised edn.

Pentus, Mati. 1993. Lambek grammars are context free. In *Proceedings of the Eighth Annual IEEE Symposium on Logic in Computer Science*, 429–433.

Pentus, Mati. 1997. Product-free Lambek calculus and context-free grammars. *Journal of Symbolic Logic* 62:648–660.

Phillips, Colin. 2006. The real-time status of island phenomena. *Language* 82:795–823.

Phillips, Colin. 2013. Some arguments and nonarguments for reductionist accounts of syntactic phenomena. *Language and Cognitive Processes* 28:156–187.

Pogodalla, Sylvain. 2009. Advances in abstract categorial grammars: Language theory and linguistic modeling. ESSLLI 2009 lecture notes, part II. `https://hal.inria.fr/hal-00749297`.

Pogodalla, Sylvain and Florent Pompigne. 2012. Controlling extraction in abstract categorial grammars. In P. de Groote and M.-J. Nederhof, eds., *Formal Grammar 2010/2011*, vol. 7395 of *Lecture Notes in Computer Science*, 162–177. Berlin: Springer.

Ranta, Aarne. 2004. Grammatical framework: A type-theoretical grammar formalism. *Journal of Functional Programming* 14:145–189.

Sag, Ivan Andrew. 1976. *Deletion and Logical Form*. Ph.D. thesis, Massachusetts Institute of Technology.

Sag, Ivan A., Gerald Gazdar, Thomas Wasow, and Steven Weisler. 1985. Coordination and how to distinguish categories. *Natural Language and Linguistic Theories* 3:117–171.

Salvati, Sylvain. 2005. *Problèmes de filtrage et problèmes d'analyse pour les grammaires catégorielles abstraites*. Ph.D. thesis, l'Institut National Polytechnique de Lorraine.

Salvati, Sylvain. 2007. Encoding second order string ACG with deterministic tree walking transducers. In S. Wintner, ed., *Proceedings of FG 2006: The 11th conference on Formal Grammar*, FG Online Proceedings, 143–156. Stanford, CA: CSLI Publications.

Savateev, Yury. 2012. Product-free lambek calculus is NP-complete. *Annals of Pure and Applied Logic* 163:775–788.

Seki, Hiroyuki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. On multiple context-free grammars. *Theoretical Computer Science* 88:191–229.

Vijay-Shanker, K., David J. Weir, and Aravind K. Joshi. 1987. Characterizing structural descriptions produced by various grammatical formalisms. In *25th Annual Meeting of the Association for Computational Linguistics*, 104–111.

Winkler, Susanne. 2005. *Ellipsis and Focus in Generative Grammar*, vol. 81 of *Studies in Generative Grammar*. Berlin: Mouton de Gruyter.

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

Yoshinaka, Ryo. 2006. *Extensions and Restrictions of Abstract Categorial Grammars*. Ph.D. thesis, University of Tokyo.

A Formal Definitions

The set $\mathrm{Tp}_{\backslash,/}(\mathcal{P})$ of *directional types* over a set $\mathcal{P}$ of atomic types is the smallest superset of $\mathcal{P}$ such that $A, B \in \mathrm{Tp}_{\backslash,/}(\mathcal{P})$ implies $A\backslash B, B/A \in \mathrm{Tp}_{\backslash,/}(\mathcal{P})$. The set $\mathrm{Tp}_{\to}(\mathcal{P})$ of *simple types* over $\mathcal{P}$ is the smallest superset of $\mathcal{P}$ such that $\alpha, \beta \in \mathrm{Tp}_{\to}(\mathcal{P})$ implies $\alpha \to \beta \in \mathrm{Tp}_{\to}(\mathcal{P})$. To each $A \in \mathrm{Tp}_{\backslash,/}(\mathcal{P})$, we associate a simple type $\overline{A} \in \mathrm{Tp}_{\to}(\mathcal{P})$ by $\overline{p} = p$ ($p \in \mathcal{P}$), $\overline{A\backslash B} = \overline{A} \to \overline{B}$, $\overline{B/A} = \overline{A} \to \overline{B}$.

For reasons of space, I use a term notation for natural deduction in the Lambek calculus. Suppose that for each $A \in \mathrm{Tp}_{\backslash,/}(\mathcal{P})$, there is a countably infinite supply $x^A, y^A, z^A, \dots$ of variables of type A. We define the set of *Lambek terms* and their *frontier*, which is a string of variables, as follows ($\circ$ stands for concatenation):

- x^A is a Lambek term of type A and its frontier is $\mathbf{fr}(x^A) = x^A$.
- If M is a Lambek term of type $A\backslash B$ and N is a Lambek term of type A, then $\mathrm{app}_{\backslash}\, MN$ is a Lambek term of type B and its frontier is $\mathbf{fr}(\mathrm{app}_{\backslash}\, MN) = \mathbf{fr}(N) \circ \mathbf{fr}(M)$.
- If M is a Lambek term of type B/A and N is a Lambek term of type A, then $\mathrm{app}_{/}\, MN$ is a Lambek term of type B and its frontier is $\mathbf{fr}(\mathrm{app}_{/}\, MN) = \mathbf{fr}(M) \circ \mathbf{fr}(N)$.
- If M is a Lambek term of type B, $\mathbf{fr}(M) = x^A \circ \gamma$, and γ contains no occurrence of x^A, then $\lambda_{\backslash} x^A.M$ is a Lambek term of type $A\backslash B$ and its frontier is $\mathbf{fr}(\lambda_{\backslash} x^A.M) = \gamma$.
- If M is a Lambek term of type B, $\mathbf{fr}(M) = \gamma \circ x^A$, and γ contains no occurrence of x^A, then $\lambda_{/} x^A.M$ is a Lambek term of type B/A and its frontier is $\mathbf{fr}(\lambda_{/} x^A.M) = \gamma$.

We assume familiarity with *(simply typed) λ-terms* and *linear λ-terms*. A Lambek term of type B with free variables $x_1^{A_1}, \dots, x_n^{A_n}$ is mapped to a simply typed linear λ-term of type $\overline{B}$ with free variables $x_1^{\overline{A_1}}, \dots, x_n^{\overline{A_n}}$, as follows:

$$\begin{aligned}
\overline{x^A} &= x^{\overline{A}},\\
\overline{\mathrm{app}_{\backslash}\, MN} &= \overline{M}\,\overline{N},\\
\overline{\mathrm{app}_{/}\, MN} &= \overline{M}\,\overline{N},\\
\overline{\lambda_{\backslash} x^A.M} &= \lambda x^{\overline{A}}.\overline{M},\\
\overline{\lambda_{/} x^A.M} &= \lambda x^{\overline{A}}.\overline{M}.
\end{aligned}$$

A *Lambek grammar* (over a terminal alphabet Σ and a set Δ of typed "meaning constants", with distinguished type $s \in \mathcal{P}$) is a finite set of triples of the form (A, a, N) ("lexical entries"), where $A \in \mathrm{Tp}_{\backslash,/}(\mathcal{P})$, $a \in \Sigma$, and N is a closed (not necessarily linear) λ-term of type $\tau(\overline{A})$ (which may contain constants from Δ in addition to bound variables) representing the meaning of a, where $\tau\colon \mathcal{P} \to \mathrm{Tp}_{\to}(\{e, t, \dots\})$ is a type substitution such that $\tau(s) = t$. If P is a Lambek term of distinguished type s with $\mathbf{fr}(P) = x_1^{A_1} \dots x_n^{A_n}$, and for each $i = 1, \dots, n$, (A_i, a_i, N_i) is a lexical entry of the grammar, then the pair $(a_1 \dots a_n, \overline{P}[x_i^{\overline{A_i}} := N_i]_{i=1}^n)$ belongs to the *string-meaning relation* defined by the grammar.

An *ACG* (over a terminal alphabet Σ and a set Δ of typed meaning constants, with distinguished type $s \in \mathcal{P}$) is a finite set of triples of the form (α, M, N) ("grammar entries"), where $\alpha \in \mathrm{Tp}_{\to}(\mathcal{P})$, M is a closed (linear) λ-term of type $\sigma(\alpha)$ containing contants from $\Sigma \cup \{\circ, \varepsilon\}$, and N is a closed λ-term of type $\tau(\alpha)$ containing constants from Δ, where $\sigma\colon \mathcal{P} \to \mathrm{Tp}_{\to}(\{str\})$ and $\tau\colon \mathcal{P} \to \mathrm{Tp}_{\to}(\{e, t, \dots\})$ are type substitutions such

that $\sigma(s) = str$ and $\tau(s) = t$. Symbols in Σ are assumed to have type str, and $\circ$ (for concatenation) and ε (for empty string) have types $str \to str \to str$ and str, respectively. If Q is a pure (i.e., constant-free) linear λ-term of type s with free variables $x_1^{\alpha_1}, \ldots, x_n^{\alpha_n}$, and for each $i = 1, \ldots, n$, (α_i, M_i, N_i) is an entry of the grammar, then the pair

$$(Q[x_i^{\alpha_i} := M_i]_{i=1}^n,\, Q[x_i^{\alpha_i} := N_i]_{i=1}^n)$$

is in the string-meaning relation defined by the ACG.

The naive translation from Lambek grammars to ACGs maps a lexical entry (A, a, N) of a Lambek grammar into $(\overline{A}, \mathrm{acg}_A\, a, N)$. The combinators acg_A of type $str \to \widetilde{A}$, where $\widetilde{A} = \sigma(\overline{A})$, $\sigma(p) = str$ for all $p \in \mathcal{P}$, together with the accompanying combinators lamb_A of type $\widetilde{A} \to str$, are defined as follows:

$$\begin{aligned}
\mathrm{acg}_p\, x^{str} &= x, & \mathrm{lamb}_p\, x^{str} &= x,\\
\mathrm{acg}_{A\backslash B}\, x^{str} &= \lambda y^{\widetilde{A}}.\, \mathrm{acg}_B\, ((\mathrm{lamb}_A\, y) \circ x), & \mathrm{lamb}_{A\backslash B}\, x^{\widetilde{A}\to\widetilde{B}} &= \mathrm{lamb}_B\, (x\, (\mathrm{acg}_A\, \varepsilon)),\\
\mathrm{acg}_{B/A}\, x^{str} &= \lambda z^{\widetilde{A}}.\, \mathrm{acg}_B\, (x \circ (\mathrm{lamb}_A\, z)), & \mathrm{lamb}_{B/A}\, x^{\widetilde{A}\to\widetilde{B}} &= \mathrm{lamb}_B\, (x\, (\mathrm{acg}_A\, \varepsilon)).
\end{aligned}$$

It is easy to see that $\mathrm{lamb}_A\, (\mathrm{acg}_A\, x^{str}) = x$ holds for all A. Of course, $\mathrm{acg}_A(\mathrm{lamb}_A\, x^{\widetilde{A}}) = x$ holds only when A is atomic.

Lemma 1. *Let M be a Lambek term of type A in normal form with $\mathbf{fr}(M) = x_1^{A_1} \ldots x_n^{A_n}$.*

(i) *If M is not a $\lambda_\backslash$- or $\lambda_/$-abstract, then $\overline{M}[x_i^{\overline{A_i}} := \mathrm{acg}_{A_i}\, x_i]_{i=1}^n = \mathrm{acg}_A(\mathbf{fr}(M))$.*

(ii) $\mathrm{lamb}_A\, \overline{M}[x_i^{\overline{A_i}} := \mathrm{acg}_{A_i}\, x_i]_{i=1}^n = \mathbf{fr}(M)$.

Proposition 2. *If M is a Lambek term of an atomic type with $\mathbf{fr}(M) = x_1^{A_1} \ldots x_n^{A_n}$, then $\overline{M}[x_i^{\overline{A_i}} := \mathrm{acg}_{A_i}\, x_i]_{i=1}^n = \mathbf{fr}(M)$.*

This means that the output ACG of the naive translation generates all pairs generated by the input Lambek grammar. The problem is that it massively overgenerates.

Clearly, what's wrong in the naive translation is the two identical clauses

$$\mathrm{lamb}_{A\backslash B}\, x^{\widetilde{A}\to\widetilde{B}} = \mathrm{lamb}_B\, (x\, (\mathrm{acg}_A\, \varepsilon)), \quad \mathrm{lamb}_{B/A}\, x^{\widetilde{A}\to\widetilde{B}} = \mathrm{lamb}_B\, (x\, (\mathrm{acg}_A\, \varepsilon)).$$

In the former clause, the value of $x^{\widetilde{A}\to\widetilde{B}}$ should really be restricted to a function that places its argument at its left edge, so to speak, while in the latter, it should be restricted to a function that places its argument at its right edge. Worth's (2014) idea was to use subtyping to narrow down the domains of the combinators $\mathrm{lamb}_{A\backslash B}$ and $\mathrm{lamb}_{B/A}$. Here, we try to obtain a similar effect by refining the atomic types in $\overline{A} \to \overline{B}$, without going beyond the original architecture of the ACG.

We temporarily introduce new constant symbols $\varepsilon_1^L, \varepsilon_2^L, \ldots$ and $\varepsilon_1^R, \varepsilon_2^R, \ldots$ of type str and new function symbols $\mathbf{O}_1^L, \mathbf{O}_2^L, \ldots$ and $\mathbf{O}_1^R, \mathbf{O}_2^R, \ldots$ of type $str \to str$, and change the definitions of acg and lamb as follows:

$$\begin{aligned}
\mathrm{acg}'_p\, x^{str} &= x,\\
\mathrm{acg}'_{A\backslash B}\, x^{str} &= \lambda y^{\widetilde{A}}.\, \mathrm{acg}'_B((\mathrm{lamb}'_A\, 1\, 1\, y) \circ x),\\
\mathrm{acg}'_{B/A}\, x^{str} &= \lambda z^{\widetilde{A}}.\, \mathrm{acg}'_B\, (x \circ (\mathrm{lamb}'_A\, 1\, 1\, z)),\\
\mathrm{lamb}'_p\, n\, m\, x^{str} &= x,
\end{aligned}$$

$$\mathrm{lamb}'_{A\backslash B}\, n\, m\, x^{\widetilde{A}\to\widetilde{B}} = \mathbf{O}^L_n\left(\mathrm{lamb}'_B\,(n+1)\,m\,(x\,(\mathrm{acg}'_A\,\varepsilon^L_n))\right),$$
$$\mathrm{lamb}'_{B/A}\, n\, m\, x^{\widetilde{A}\to\widetilde{B}} = \mathbf{O}^R_m\left(\mathrm{lamb}'_B\,n\,(m+1)\,(x\,(\mathrm{acg}'_A\,\varepsilon^R_m))\right).$$

The combinator lamb'_A is of type $\mathit{int}\to\mathit{int}\to\widetilde{A}\to\mathit{str}$, where int is the type of positive integers. The integer arguments of lamb'_A are there to distinguish different left- (or right-) peripheral gaps bound by the same "binder". Note that we can recover $\mathrm{acg}_A\,a$ from $\mathrm{acg}'_A\,a$ by substituting ε for ε^L_i and ε^R_j and $\lambda x^{\mathit{str}}.x$ for $\mathbf{O}^L_i$ and $\mathbf{O}^R_j$.

The resulting ACG is then filtered through two finite tree automata. Given maximal values $n_{\max}$ and $m_{\max}$ for n and m (the two integer arguments of lamb'_A), the automata have $n_{\max}(n_{\max}+1)/2+1$ and $m_{\max}(m_{\max}+1)/2+1$ states, respectively. The states of the second automaton are $[\textsc{rgap}\,0]$ and $[\textsc{rgap}\,i\ldots j]$ with $1\le i\le j\le m_{\max}$, and its transitions are

$$\begin{aligned}
a &\to [\textsc{rgap}\,0] \quad \text{for each terminal } a,\\
\varepsilon^L_i &\to [\textsc{rgap}\,0] \quad \text{for } i=1,\ldots,n_{\max},\\
\varepsilon^R_i &\to [\textsc{rgap}\,i] \quad \text{for } i=1,\ldots,m_{\max},\\
[\textsc{rgap}\,0]\circ[\textsc{rgap}\,0] &\to [\textsc{rgap}\,0],\\
[\textsc{rgap}\,0]\circ[\textsc{rgap}\,1\ldots i] &\to [\textsc{rgap}\,1\ldots i] \quad \text{for } 1\le i\le m_{\max},\\
[\textsc{rgap}\,i\ldots j]\circ[\textsc{rgap}\,(j+1)\ldots k] &\to [\textsc{rgap}\,i\ldots k] \quad \text{for } 1\le i\le j<k\le m_{\max},\\
\mathbf{O}^L_i\,q &\to q \quad \text{for each state } q \text{ and } i=1,\ldots,n_{\max},\\
\mathbf{O}^R_1\,[\textsc{rgap}\,1] &\to [\textsc{rgap}\,0],\\
\mathbf{O}^R_j\,[\textsc{rgap}\,1\ldots j] &\to [\textsc{rgap}\,1\ldots(j-1)] \quad \text{for } 1<j\le m_{\max}.
\end{aligned}$$

The definition of the first automaton is completely symmetric. We use new atomic types of the form $p[\textsc{lgap}\,l,\textsc{rgap}\,r]$ as a feature-specified variant of p. Given an entry

$$(\alpha, M', N) = (\overline{A}, \mathrm{acg}'_A\,a, N)$$

obtained from a Lambek grammar entry (A,a,N) and a feature specified variant α' of α, the entry (α', M', N) is in the filtered grammar if for $i=1,2$, M' has type $(\alpha')_i$ under the typing τ'_i of the constants determined by the ith automaton, where $(\alpha')_i$ is defined inductively as follows:

$$\begin{aligned}
(p[\textsc{lgap}\,l,\textsc{rgap}\,r])_1 &= [\textsc{lgap}\,l],\\
(p[\textsc{lgap}\,l,\textsc{rgap}\,r])_2 &= [\textsc{rgap}\,r],\\
(\alpha'\to\beta')_i &= (\alpha')_i\to(\beta')_i.
\end{aligned}$$

The typing τ'_i is a simple kind of intersection typing that assigns a set of types to each constant. For example, $q_1\to q_2\to q\in\tau'_i(\circ)$ if and only if $q_1\circ q_2\to q$ is a transition of M_i.

If (α', M', N) is in the filtered grammar, with $M'=\mathrm{acg}'_A\,a$, then (α', M, N) is in the "final product" grammar, where $M=\mathrm{acg}_A\,a$.

Compositional semantics of *same*, *different*, *total*

Oleg Kiselyov
Tohoku University, Japan

Abstract

We propose a strictly compositional and uniform treatment for the internal readings of the symmetric predicates such as *same* and *different* and for the summative phrases such as *a total of*. The analyses handle multiple occurrences of symmetric and summative predicates in the same sentence, alongside of multiple quantificational elements. We treat symmetric predicates and summative phrases as wide-scope generalized existential quantifiers for choice functions. Like Barker's (2007) we treat symmetric and summative expressions as generalized quantifiers and relate them to universal quantification. However, our quantifiers scope wide rather than parasitically and avoid Barker's non-standard interpretation of universal quantification.

We introduce the analyses in terms of the familiar Quantifier Raising framework, which we make more precise and treat as a somewhat informal version of a type-logical grammar.

Keywords: Quantification; QR; TLG; Dependent quantification; Quantifier ambiguity; Distribution

1 Introduction

The internal readings of symmetric predicates in (1-4) challenge the compositional semantics (Barker, 2007; Kubota and Levine, 2013, 2014):

(1) John and Bill checked the same book.
(2a) John and Bill checked the same book at the same store.
(2b) John and Bill checked the same book at different stores.
(2c) John and Bill checked the same book at the same store on the same day for the same topic.
(3) John read and Bill reviewed {the same book/different books}.
(4) I gave the same book to John on Wednesday and to Bill on Friday.

All these sentences have (deictic) readings, referring to some book, store or day identified from the context. For example, when (1) appears in a passage after the sentence "Mary bought the latest Harry Potter novel" it may mean that John and Bill also checked that Harry Potter novel. However, (1) also has the meaning by itself: "the same book" refers to some unspecified book that John and Bill separately checked out. The sentence itself then provides the context of interpreting "the same". This is the internal reading of the sentence, which is the main subject of this paper.

Even in simple cases (1-2) the conjoined NPs seem to need the denotations that expose the conjuncts (so they can be accessed by distributivity operators). The meaning of a phrase then is determined not just from the meaning but also from a part of the structure of the components (see (Kubota and Levine, 2014, §2.1) for detailed discussion.) Problems accumulate when symmetric predicates are used with the non-canonical coordination (3-4).

The compositional semantic analysis of such internal readings has been a challenge, reviewed in (Barker, 2007). There has been even a proof that there can be no compositional analysis with generalized quantifiers (Keenan, 1992). Barker (2007) was first to give the strictly compositional account of the *same*, using the novel idea of "parasitic scope" and a non-standard quantifier. Alas, this analysis do not appear to be adequate (Kubota and Levine, 2014) in cases of multiple symmetric predicates.

Summative predicates such as 'total' in (5a) are just as challenging:

(5a) John gambled and Bill lost the total of 10,000.

(5b) The two men lost the total of 10,000.

(5c) John gave, and Bill lent, a total of 10,000 to {the same students/different students}

In (5a), 10,000 is the sum of John's gambling amount and Bill's losses. To truly complicate the matters, summative and symmetric predicates can also appear together, as in (5c).

We start in §3 with a simpler intuitive analysis, treating *same* as a property of belonging to an equivalence class. Hence *same* behaves like a wide-scope existential quantifier for the reference entity of the class. The internal and deictic *same* are analyzed uniformly. The simple analysis does not work for *different* and it does not account for the empirical fact that the internal reading is only available when there is some element in a sentence that entails multiple events or situations.

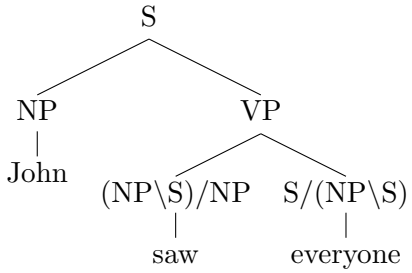
The generalized analysis in §5 overcomes both problems. It uniformly treats *same*, *different* and *total* – as wide-scope quantifiers, not over entities but over Skolem functions whose result is a property. The analysis is based on 'dependent quantification' explained in §4, which may be seen as an alternative to parasitic scope.

We introduce the analyses in terms of quantifier raising (QR) at LF in the style of Heim and Kratzer (1997) – solely because of the wide familiarity of that style and for the sake of comparison with (Barker, 2007). The analysis does not require positing LF. In fact, we treat QR as a somewhat informal version of a type-logical grammar.

2 QR, a less ad hoc version

For reference we briefly describe the QR formalism used in this paper. It is essentially the same as in Heim and Kratzer (1997), but with one less ad hoc construction and with the explicit marking of trace variables (hypotheses) in node types, bringing QR closer to type-logical grammars.

We introduce the formalism on the familiar example:

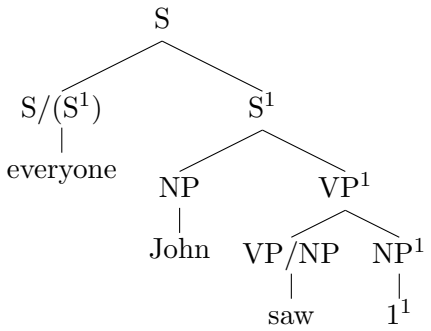


We take S, NP and N as basic categories and write VP as the abbreviation for $NP\backslash S$, *Adj* for N/N, PP for $N\backslash N$ or $VP\backslash VP$, *Det* for NP/N and *Prep* for PP/NP.

If we give everyone the lifted type $S/(NP\backslash S)$, type clash occurs: everyone can neither apply nor be applied to the verb saw. QR resolves this clash: the quantifier is raised and adjoined to its scope target, leaving at its former place a variable (to be called trace) of the simple NP type. We use numerals for such variables. Here is the result:

X	$\mapsto$	$\lceil X \rceil$
NP	$\mapsto$	e
N	$\mapsto$	(et)
S	$\mapsto$	t
$A\backslash B$	$\mapsto$	$(\lceil A \rceil \lceil B \rceil)$
B/A	$\mapsto$	$(\lceil A \rceil \lceil B \rceil)$
$X^{n:A,\Gamma}$	$\mapsto$	$(\lceil A \rceil \lceil X^{\Gamma} \rceil)$

Figure 1: Mapping Node types to semantic types. The type of hypothetical variables, shown as A, is most always NP. We will omit this type for brevity.



Two things are different from the Heim and Kratzer's (1997) presentation. First, we index all nodes with the variables contained therein. After raising, the object of *saw* is the variable 1 of the category NP^1. The VP node likewise gets the 1 superscript, because its meaning depends on that *assumed* NP. In other words, the traces of raised quantifiers are treated as hypotheses; each node of the tree is indexed with its hypotheses.

The raised quantifier changes type: it is no longer $S/(NP\backslash S)$ but S/S^1. In the new type, NP is dropped from the middle and its neighbor S gets the superscript 1, which is the name of the NP variable left at the quantifier's original place. The raised quantifier hence is the binder of that variable, the eliminator of the corresponding hypothesis. Our presentation of QR is less ad hoc, keeping the explicit track of variables and requiring no extra adjoining step with the non-standard interpretation. The former brings the QR analysis closer to the logically-motivated type-logical grammars (TLG).

The denotation of LF also becomes more precise on our analysis: the meaning of a branch X^{Γ} (where Γ is the list of variables within that branch, along with their types) is, informally, a function from the meaning of the hypotheses to the meaning of X. More precisely, the denotation is determined according to Fig. 1. For example, the denotation of the VP^1 node is $(e(et))$. The raised quantifier of the category S/S^1 gets the semantic type $(et)t$ with the obvious denotation $\lambda C : (et).\, \forall x : e.\, C\ x$.

The quantifier-raising procedure is thus a sequence of the following steps (compare with (Heim and Kratzer, 1997) and its presentation in (Barker, 2007, §5.4))

1. Replace the scope-taking expression (generalized quantifier) of the type $S/(X\backslash S)$ with the variable n of the type X;

2. Add the new variable as a superscript to all parent nodes;

3. Change the type of the scope-taking expression to S/S^n and adjoin to its scope target.

The type X is usually NP; in §3 we consider quantifiers over other types, for example, Adj. We will argue that *same* is such quantifier.

We stress that in our presentation of QR, in contrast to that in (Heim and Kratzer, 1997; Barker, 2007), there is no longer an ad hoc step of the adjoining the second occurrence of the trace variable. That second occurrence was crucial for Barker's analysis: that was the scope target for the parasitic scope. Since in our presentation there is no such step, our analysis hence is distinct from parasitic scope.

3 Simple compositional analysis of *same*

As the preface to the full analysis of symmetric and summative predicates we describe in this section a simplified version, which applies only to *same* and the similar *identical*, *nearly identical*, *similar*, etc.

Our analysis takes literally the remark in (Barker, 2007, §3) about the meaning of

(7) Everyone read the same book.

"In addition to the deictic reading (on which we have a specific book in mind, say, Emma), (7) has an internal reading on which it is true if there is any book such that everyone read that book." It appears therefore that the truth conditions of (7) are the same as

Everyone read some book.

on the reading where the existential takes the wide scope. The observation is in agreement with Barker's (2007) conclusion that "it seems inescapable that on the internal reading, some element in the sentence must in effect introduce an existential quantifier over names." We propose that *same* is such an element.

We further observe that the relation of 'being the same' (as well as 'similar', 'identical', 'nearly identical', etc) is an equivalence relation. The adjective 'same' thus signifies the property of an entity of being a member of an equivalence class. A class can be represented by one of its members, to be called reference entity. The first attempt at the analysis of *same* is to index it by the reference element. Thus we have for same_i:

Category	Adj
Semantic type	$(et)(et)$
Denotation	$\lambda P\!:\!(et).\,\lambda x\!:\!e.\,P(x) \wedge x = i$

Our example (7) will then read

(7') Everyone read the same_i book.

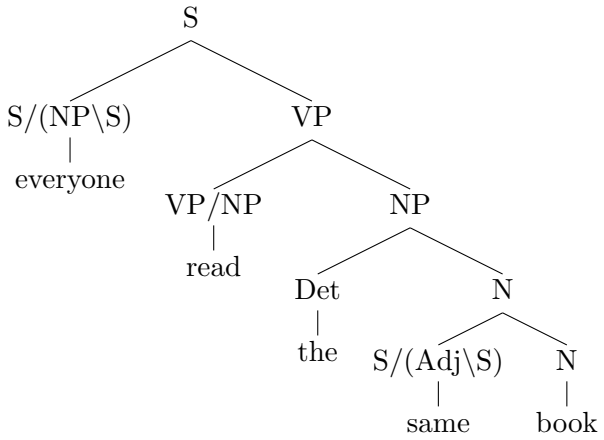
It has the internal reading if i is existentially quantified at the sentence boundary, and the deictic reading if i is left unbound, to be bound somewhere in the context. The symbol $=$ is a shorthand for the "sufficiently the same" relation.

We may as well make *same* to introduce the quantifier for the reference entity. This gives us the final analysis of *same* in this section, cast in terms of the QR framework of §2:

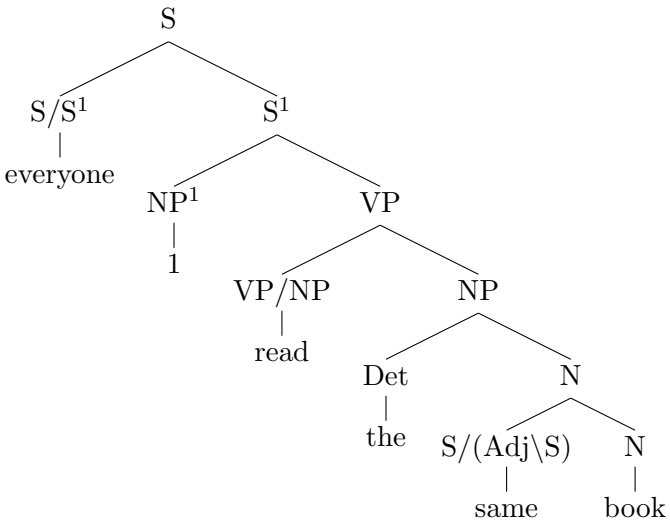
Category	$S/(Adj\backslash S)$
Trace named n	$Adj^{n:Adj}$
Category of raised *same*	$S/S^{n:Adj}$
Its semantic type	$(((et)(et))t)t$
Denotation	$\lambda C\!:\!((et)(et))t.\,\exists i\!:\!e.\,C\ (\lambda P\!:\!(et).\,\lambda x\!:\!e.\,P(x) \wedge x = i)$

We postulate that *same* takes the widest scope.

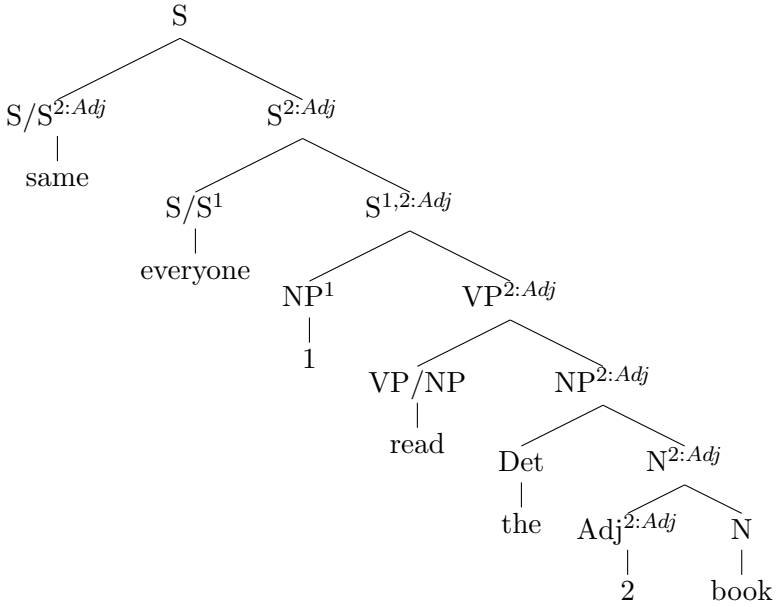
As an illustration, we give the complete analysis of (7). Before raising, (7) is represented by:



We raise *everyone*:



with *same* to follow:



The truth conditions are read off in the standard way:

$\exists i : e. \forall x : e. \iota y.\ \ read(y, x) \wedge book(y) \wedge y = i$

We have postulated that *same* scopes over any other quantifier that may be in the sentence. The difference between the internal and deictic readings now boils down to

quantifying at the sentence vs. paragraph boundary. For example, Barker's (2007) example of the deictic *same* "My friend Heddy's name is highly unusual; nevertheless, two women in this room have the same name." will have the meaning

$$\exists i.\,(i = Heddy) \wedge name(Heddy) \wedge unusual(Heddy) \wedge$$
$$two(\lambda w.\,woman(w) \wedge inThisRoom(w) \wedge hasName(i))$$

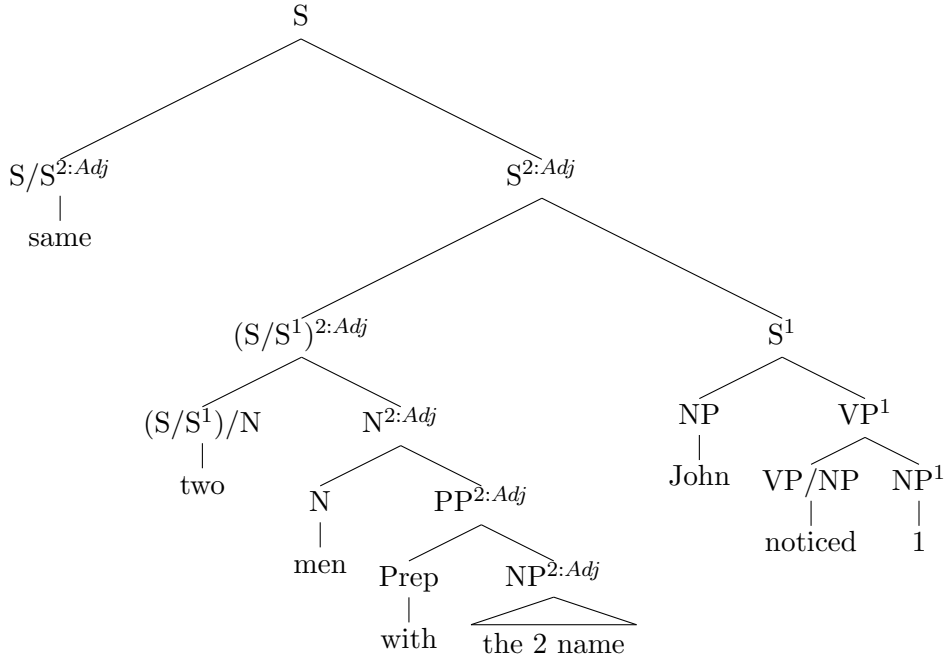
Barker (2007) likewise posits *same* to be a quantifier, which has the uncommon category $N/(Adj\backslash N)$ and takes scope at nominal boundary. It is this nominal boundary that gives rise to the parasitic scope. Our *same* is the common generalized quantifier, taking scope at sentence or clause boundary (or even beyond).

Our analysis easily accommodates several *same* in the same sentence, as in

(2a) John and Bill checked the same book at the same store.

Both *same* are raised, one after the other. As yet another example, here is the final analysis of

John noticed two men with the same name.



The analysis, albeit simple and intuitive, comes short in two respects. First, it does not extend to *different*, which does not denote an equivalence relation. Second, nothing prevents raising *same* in cases with no internal readings, such as (8a):

(8a) John read the same book.

(8b) John read a book.

On our simple analysis, (8a) has the same meaning as (8b), which is odd. We will overcome these drawbacks in the more general analysis in §5.

4 Dependent quantification

Quantifying one of term's variables can be done independently of the others. On the other hand, the other variables may depend on the quantified one. The dependency can be represented by Skolem (choice) functions. This process of eliminating one hypothesis that affects others underlies our general analysis of symmetric predicates and summative phrases. This section describes the process in detail.

The inspiration of our approach is the Kubota and Levine's (2014) observation that *same* and *different* are essentially disambiguators. Indeed, consider (9a) with an indefinite determiner:

(9a) John and Bill checked a book.
(9b) John and Bill checked the same book.
(9c) John and Bill checked a different book.

(9a) is ambiguous: either there is one book that John and Bill checked, or John and Bill each checked a book that may or may not be the same. Then *same* and *different* disambiguate.

We illustrate the mechanism of the disambiguation using first-order logic. Let P be a binary predicate.

(10a) $\exists x.\, \forall y.\, P(x, y)$
(10b) $\forall y.\, \exists x.\, P(x, y)$
(10c) $\exists f.\, \forall y.\, P(f(y), y)$

Formula (10a) asserts that there exists one particular x' such that $P(x', y)$ holds for any choice of y. In contrast, in (10b) the x that makes $P(x, y)$ hold *generally* varies with the choice of y. The choices of x may be all different, all the same, or some different and some the same.

One way of disambiguating (10b) immediately springs to mind: Skolemization, as in (10c). Choosing the Skolem function f to always return the same x' will disambiguate (10b) to its special case (10a). One can also make a Skolem function that ensures all choices are different from each other.

Let us discuss the introduction of Skolem functions more precisely. Let $P(x, y)$ be a formula with two free variables (in other words, derived with two hypotheses named x and y of the types A and B respectively). The formula can be written as

$$p_{xy} = \lambda x : A.\, \lambda y : B.\, P(x, y)$$

making the dependence on the hypotheses explicit. We wish to universally quantify the variable y, by applying the quantifier $qU : ((Bt)t)$

$$qU = \lambda F : (Bt).\, \forall z : B.\, F\ z$$

Quantifying without regard for x gives $\forall y.\, P(x, y)$. In more detail:

$$\lambda x : A.\, qU\ (p_{xy}\ x)$$

In general however the hypothesis x may be correlated with y. Therefore, the general quantification is

$$\lambda f : (BA).\, qU\ (\lambda y : B.\, p_{xy}\ (f\ y)\ y)$$

The Skolem (choice) function f expresses the possible correlation between the hypothesis. The simple quantification is obtained as a special case when f is a constant function. We call this general procedure dependent quantification and use it extensively in the analyses of the next section.

It should be clear that dependent quantification only makes sense for universal quantifiers, which asserted a formula for potentially multiple choices of the quantified variable.

5 General analysis of symmetric and summative predicates

This section presents the general and uniform analysis of *same*, *different* and *total*. The analysis is the generalization of the simple version of §3. Here is how we propose to analyze *same* now:

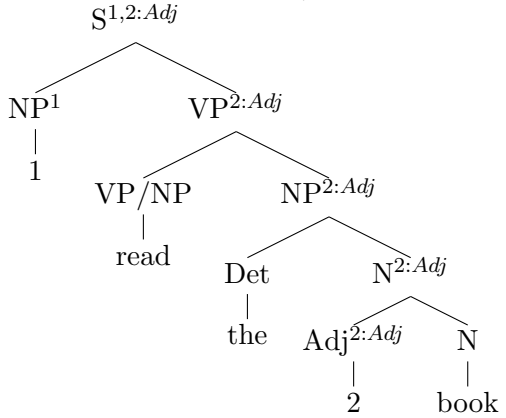
Category	$S/(Adj\backslash S)$
Trace named n	$Adj^{n:Adj}$
Category of raised *same*	$S/S^{n:Adj^{\Gamma}}$
Its semantic type	$((\Gamma((et)(et)))t)t$
Denotation	$\lambda C:(\Gamma((et)(et)))t.\,\exists i:e.\,C\ (\lambda y:\Gamma.\,\lambda P:(et).\,\lambda x:e.\,P(x)\wedge x=i)$

As before, the trace left by the raised *same* is of the type of an adjective $Adj^{n:Adj}$. The raised *same*, however, binds a variable of a higher-type: $n : Adj^{\Gamma}$ rather than $n : Adj$, where Γ is some type (usually NP). The denotation has hardly changed though: the new argument $y : \Gamma$ is ignored. Thus the meaning of *same* is still the property that an entity belongs to an equivalence class, regardless of the choice for other variables that may be in scope.

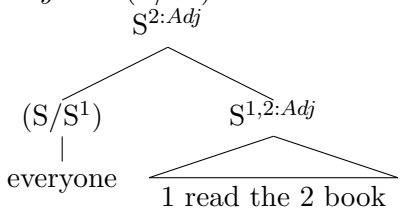
As an illustration, we re-do the analysis of (7) from §3

(7) Everyone read the same book.

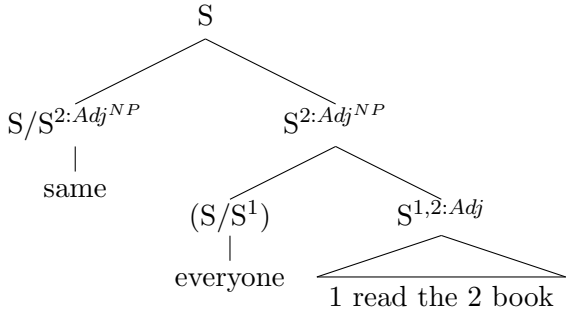
First, *everyone* and *same* are replaced with the corresponding variables and the tree nodes are marked with the variables' names and types: (As usual, we elide NP in marking):



We can adjoin the raised $everyone : (S/S^1)$ as usual obtaining



but then we are stuck. The raised *same* has the higher type $S/S^{n:Adj^{\Gamma}}$ rather than the simple $S/S^{n:Adj}$: it binds choice functions over adjectives rather than just adjectives. Therefore, we have to use the dependent quantification when adjoining *everyone*, which leads to the successful analysis:



The truth conditions are read off as usual (keeping in mind the dependent quantification):

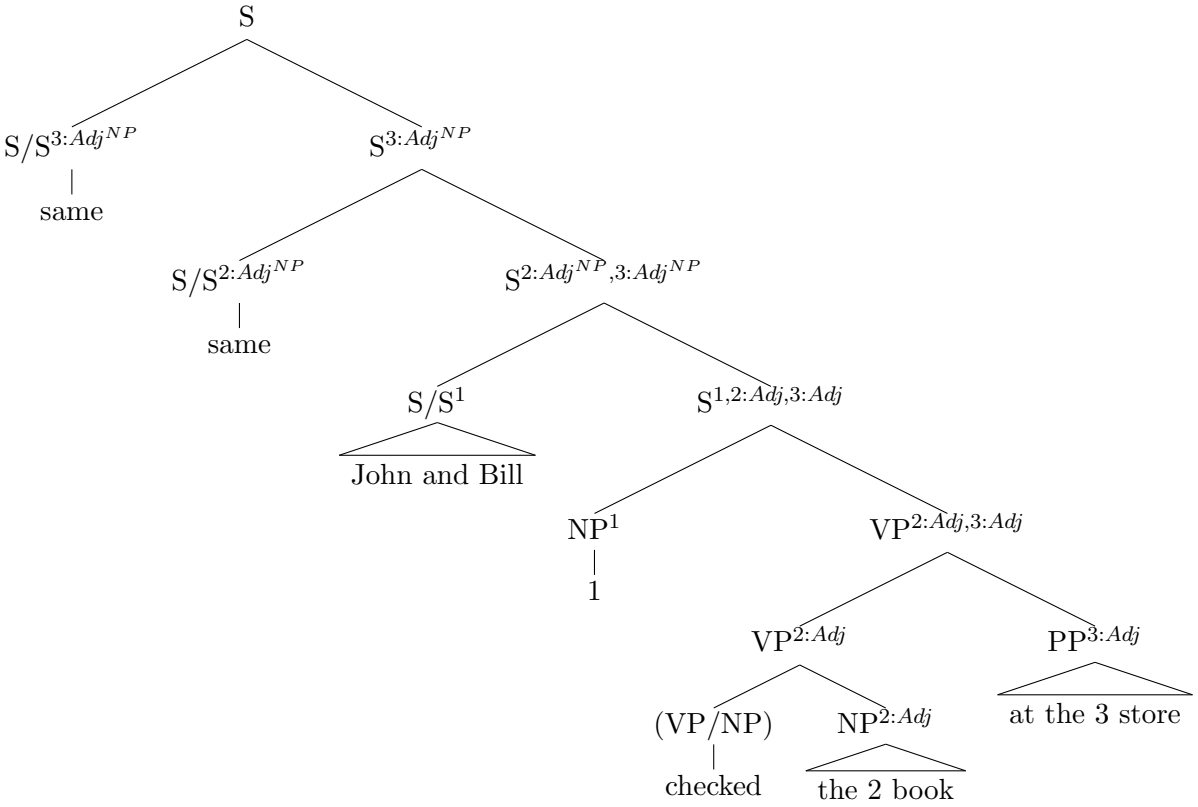
$same\ (\lambda f_2.\, everyone\ (\lambda y_1.\, (\lambda y_1.\, \lambda p_2.\, \iota z.\, read(z, y_1) \wedge p_2(z) \wedge book(z))\ y_1\ (f\ y_1)))$
$\rightsquigarrow\ \exists i.\, \forall y.\, \iota z.\, read(z, y) \wedge z = i \wedge book(z)$

If we replace "everyone" with "John" in the sample sentence, the analysis does not go through: there is no longer a universal quantifier that could do the dependent quantification and introduce the choice function variable for the raised *same* to bind. We thus predict that the internal reading is only possible when some element in a sentence can do dependent quantification.

Like Barker (2007), we relate *same* with the universal quantification and use choice functions (of different types though). However, we existentially quantify the choice function wider than the universal and hence we avoid the parasitic scope. We also avoid the unusual postulate of (Barker, 2007) that the universal quantifier quantifies over groups of entities rather than atomic entities.

Our analysis also easily deals with multiple occurrence of *same*, for example

(2a) John and Bill checked the same book at the same store.



Unlike before, the new analysis of *same* extends to *different*:

Category	$S/(Adj\backslash S)$
Trace named n	$Adj^{n:Adj}$
Category of raised *different*	$S/S^{n:Adj^{\Gamma}}$
Its semantic type	$((\Gamma((et)(et)))t)t$
Denotation	$\lambda C : (\Gamma((et)(et)))t.$
	$\quad \exists p : (e(et)).\, (\forall u : \Gamma.\, \forall v : \Gamma.\, u \neq v \Rightarrow p(u) \cap p(v) = \emptyset) \wedge$
	$\quad C\ (\lambda y : \Gamma.\, \lambda P : (et).\, \lambda x : e.\, P(x) \wedge p(y)(x))$

Thus *different* finds such property p indexed by $y : \Gamma$ that different indices correspond to non-intersecting properties. Since the types and categories are those of *same*, *same* and *different* are freely interchangeable. For example, it is trivial to adjust the above analysis for (2b) and even (2c):

(2b) John and Bill checked the same book at different stores.
(2c) John and Bill checked the same book at the same store on the same day for the same topic.

The final piece of the puzzle are the summative phrases like (5b).

(5b) The two men lost the total of 10,000.

Although *the total of s* is an NP rather than Adj, its analysis is similar:

Category	$S/(NP\backslash S)$
Trace named n	$NP^{n:NP}$
Category of raised *the total of s*	$S/S^{n:NP^{\Gamma}}$
Its semantic type	$((\Gamma e)t)t$
Denotation	$\lambda C:(\Gamma e)t.$ $\exists f:(ee).\,(s = \sum_{v:\Gamma} f(v)) \wedge C\ (\lambda y:\Gamma.\, f\ y)$

6 Conclusions and Future work

To summarize, we propose that on the internal reading, *same*, *different*, as well as *the total of s*, are widest-scope generalized quantifiers. Unusually however, whereas *same* or *different* look like an adjective in situ, what they quantify is a choice function over adjectives rather than merely a variable of the Adj category. The conversion from an Adj to a choice function is due to the dependent quantification. A sentence therefore must have some universal quantificational element, capable of dependent quantification, for *same* and *different* to have internal readings.

Our analysis is strictly compositional and accommodates multiple occurrences of symmetric and summative phrases within the same sentence. It also predicts the ambiguity in

Different waiters served everyone lunch and dinner.

with two quantifiers.

The quantification over choice functions has independent motivations, discussed by Szabolcsi (2000, §3.2.1) (in the context of works by Reinhart and Winter), for example, to explain both readings of

(11) Three relatives of mine inherited a house.
(i) 'there are three relatives of mine who together inherited a house'
(ii) 'there are three relatives of mine who each inherited a house'

especially the first.

The immediate future work is analyzing respective readings of plural and conjoined expressions.

Acknowledgments.

I am indebted to Yusuke Kubota for his encouragement, many insightful conversations and introducing a wealth of the empirical material. I thank anonymous reviewers for the helpful comments and the recommendation to explain the analysis in simple and familiar framework.

This work was partially supported by JSPS KAKENHI Grant Numbers 22300005, 25540001, 15H02681.

References

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

Heim, Irene and Angelika Kratzer. 1997. *Semantics in Generative Grammar*. Oxford: Blackwell Publishers.

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

Kubota, Yusuke and Robert Levine. 2013. Empirical foundations for hybrid type-logical categorial grammar. the domain of phenomena.

Kubota, Yusuke and Robert Levine. 2014. The syntax-semantics interface of 'respective' predication: A unified analysis in hybrid type-logical categorial grammar. in print.

Szabolcsi, Anna. 2000. The syntax of scope. In *Handbook of Contemporary Syntactic Theory*, 607–634. Blackwell.

A Unidimensional Syntax-Semantics Interface for Supplements

Scott Martin
Natural Language Understanding
and Artificial Intelligence Laboratory
Nuance Communications
1198 East Arques Avenue
Sunnyvale, California 94085 USA
http://coffeeblack.org/

Abstract

I extend the recent unidimensional semantics of supplements due to Martin to a full syntax-semantics interface. The grammar formalism employs a two-component syntax, with one component modeling combinatorics and another for surface form. I show that the syntax-semantics for supplements is relatively uncomplicated, requiring only two new lexical entries. I contrast this with the complex machinery needed to model supplement semantics in other accounts, such as transformations, special-purpose rules, continuations, and monads. This account is also more finely grained and empirically adequate than others because it allows supplements to take scope and be denied.

Keywords: supplements; unidimensionality; syntax-semantics interface; scope

1 Introduction

Ever since Potts's (2005) reexamination of *supplements* (nominal appositives, nonrestrictive relative clauses, *as*-parentheticals, etc.), they have been predominantly accounted for by semantics that are *multidimensional* in the sense that a supplement's content is kept separate from surrounding content. In addition to Potts, multidimensional accounts of supplements have been proposed by Nouwen (2007), Giorgolo and Asudeh (2012), and Martin (2013). Others have crafted accounts without a separate dimension for supplements *per se*, but with a separate mode for supplements to contribute their content; examples include Kubota and Uegaki (2009), AnderBois et al. (2010, 2015), Koev (2012, 2014), and Bekki and McCready (2014). Barker et al.'s (2010) treatment of *expressives* also uses a separate contribution mode.

More recently, both the empirical characterization of supplements as semantically separate and accompanying multidimensional accounts have been questioned by authors citing the possibility of anaphoric interactions with supplement content (Amaral et al., 2007; AnderBois et al., 2010, 2015; Martin, 2013) and the capability of supplements to take scope (Schlenker, 2010, ms; Nouwen, 2014; Martin, in press). The account of the semantics of supplements in Martin in press, hereafter *SU*, uses only a single lexical entry, for the *comma intonation* (Huddleston and Pullum, 2002, 1351), in addition to the usual

mechanisms for modeling quantifier scope ambiguity and dynamic semantics. This paper provides a syntax-semantics interface for SU's account that extends Martin and Pollard's (2014) two-component categorial syntax with lexical entries for the comma intonation. Along the way, I make a case for this straightforward account by critiquing the perspective on supplements that has dominated since Potts's work.

As I discuss below in detail, the move to a two-component syntactic framework allows an analysis of the syntax of supplements that not only generates SU's semantics but is also much more parsimonious and economical than many other accounts. In contrast to the analyses due to McCawley (1998), del Gobbo (2007), and Schlenker (2010, ms), no intricate movement rules are required, nor is it necessary to appeal to *E-type pronouns*. There are no dedicated modes of combination or special-purpose interpretation rules, unlike Potts's (2005) account. And it is not necessary to invoke the *continuation passing* technique, as do Kubota and Uegaki (2009), *monads*, as do Giorgolo and Asudeh (2012), or a specialized operator, as do Bekki and McCready (2014).

The rest of the paper is organized as follows. The conventional wisdom about supplements is reviewed in §2, followed by a proposed recharacterization in §2.2. The syntax-semantics interface is presented in §3, starting with a brief overview of the semantic formalism (§3.1). After treating some basic supplement examples in §3.2, the account examines supplement (non)projection (§3.3), anaphoric links between supplements and other content (§3.5), and supplement stacking (§3.6). Some comparisons with other accounts are discussed in §4, and then §5 concludes.

2 Conventional implicature and conventional wisdom

According to Potts's (2005) characterization, which has been adopted by many others, supplements are semantically inert with respect to surrounding content. The major implications of this claim are the following:

1. Supplements are predicted to be *scopeless* (or, equivalently, to always take widest scope) because they do not interact with truth-functional operators.

2. Unlike regular nonsupplement content, supplements should never be *at-issue*, and therefore should not be capable of denial or contradiction in normal circumstances.

3. Any account of the semantics of supplements must be multidimensional, with supplement content contributed on a separate, dedicated dimension from regular at-issue content.

These implications render dubious the claim that supplements are semantically inert for the simple reason that they do not agree with observations. The following section explores this mismatch by looking at some supplement data.

2.1 A fresh look at the data

The first doubts about Potts's (2005) characterization of supplements were expressed by Amaral et al. (2007), who pointed out that supplements participate in both anaphoric and scopal interactions with other content. Amaral et al. give examples of supplements interacting with the quantifiers *every* and *several*.

(1) a. In each class, several students$_i$ failed the midterm exam, which they$_i$ had to retake later.

b. It seems like every time I turn around, my neighbor with a motorcycle is dating a different woman, who always has one too.

(Amaral et al., 2007, (36) and (38))

For both examples in (1), the supplement must be interpreted in the scope of the relevant quantifier. For (1a), several students *in each class* had to take the exam later; in (1b), there is a different woman *on each occasion* who also has a motorcycle. Amaral et al. also cite the ability of supplements to take on a nonspeaker perspective (experimentally attested by Harris and Potts (2009)) as further evidence that supplements can scope narrow, since nonspeaker perspective shift involves a supplement in the scope of an attitude verb.

Context: Joan is delusional, believing that a chip has been installed in her brain allowing her to speak multiple languages.

(2) Joan believes that her chip, which she had installed last month, has a twelve year guarantee. (Amaral et al., 2007, (27))

In (2), the implication that Joan's chip was installed last month does not survive into the matrix, speaker-anchored utterance, holding only in the context of what Joan believes.

Nouwen (2014) shows that supplements can sometimes be outscoped by negation, in addition to other quantifiers.

(3) a. It's not the case that a boxer, a famous one, lives in this street.

b. Every boxer has a coach, a famous one.

(Nouwen, 2014, (25) and (32))

Though a wide-scope supplement reading is also available, both examples in (3) have a reading where the operator outscopes the supplement. For (3a), this reading is equivalent to *It's not true that a famous boxer lives in this street.* For (3b), it is equivalent to *Every boxer has a famous coach.*

There is also evidence that supplements participate in scope interactions based on their behavior in conditionals, as Schlenker (ms) points out.

(4) a. If tomorrow I call the chair, who in turn calls the dean, we'll be in deep trouble. (Schlenker, ms, (66))

b. If tomorrow I call the chair, who has had it out for me for a long time, we'll be in deep trouble.

These examples show an interesting contrast, with (4a) only implying that the chair will call the dean provided I call her, i.e., the supplement is interpreted in the scope of *if.* On the other hand, the supplement in (4b) scopes wide relative to the conditional, so that the entire utterance implies that the chair has long had it out for me whether or not I call her.

SU gives evidence not only of negation and quantifiers outscoping supplements, but also of discourse binding within a supplement and cases where a quantifier seems to outscope a supplement.

(5) a. Every famous boxer I know$_i$ has a devoted brother, who he$_i$ completely relied on back when he$_i$ was just an amateur.

b. But there would always be some student, a photographer or a glassblower, who would simply have taken a piece of newspaper and folded it once and propped it up like a tent and let it go at that.

c. No Tibetan Buddhist$_i$ thinks the Dalai Lama, his$_i$ spiritual mentor, would ever cave to Chinese pressure tactics.

(Martin, in press, (16), (17) and (27))

The supplement in (5a) is obligatorily interpreted in the scope of the quantifier *Every* because of the discourse-bound pronoun *he*. A naturally occurring narrow-scope supplement is given in (5b). SU analyzes examples like (5c), which implies that the Dalai Lama is every Tibetan Buddhist's spiritual mentor, as instances of *telescoping* (Roberts, 1989, 2005).

The examples in (1) and (5) show that a pronoun in a supplement can take its antecedent outside the supplement. The following example shows that supplements can also introduce antecedents for anaphora in outside content:

(6) Kim$_i$'s bike$_j$, which used to have reflectors$_k$ on it$_j$, was pretty safe to ride at night until she$_i$ decided to take them$_k$ off. (Martin, in press, (35))

Taken together, these examples argue for an approach to supplements that allows them to interact with surrounding content, both anaphorically and in term of scope. Amaral et al. (2007) show that these interactions extend to presuppositions, e.g. cases where *too* is licensed by content inside a supplement, such as (1b).

AnderBois et al. (2010, 2015), Koev (2012), and Schlenker (ms) all point out that the linear position of a supplement within its containing utterance can influence its deniability.

(7) a. i. He told her about Luke, who loved to have his picture taken.

ii. No, he didn't like that at all.

iii. No, he told her about Noah.

b. i. Luke, who loved to have his picture taken, was his son.

ii. ?? No, he didn't like that at all.

iii. No, Luke was his nephew.

(AnderBois et al., 2010, (48) and (50))

(8) a. i. Jack invited Edna, who is a fearless leader.

ii. No, she isn't. She is a coward.

b. i. Edna, who is a fearless leader, started the descent.

ii. # No, she isn't. She is a coward.

(Koev, 2012, (4) and (5))

These contrasts show that both of the utterance-final supplements in (7a) and (8a) can be felicitously denied, whereas their respective utterance-medial counterparts in (7b) and (8b) are less straightforwardly denied. Examples like these show that supplement content can sometimes be at-issue, at least when it occurs utterance-finally.

There is also evidence that an utterance-medial supplement can address the *question under discussion (QUD)* (Roberts, 2012a,b), and therefore be at-issue.

Context: The interlocutors are participants at a math conference.

(9) a. Do you know whether the axiom of Choice is independent of ZF?

b. Well, Paul Cohen, who proved it is back in 1963, is sitting in the back row. So you can go ask him.

(Martin, in press, (84))

The supplement in (9b) is directly addressing the question raised in (9a).

2.2 Recharacterizing supplements

Examples like those in (1)–(5) offer strong evidence that supplements are not, in fact, scopeless, as they clearly interact with quantifiers, negation, and conditionals. And (5) and (6) further erode the contention that supplements are inert, since anaphora appears to work for them in basically the same way as it does for nonsupplement content. Lastly, examples like (7)–(9) cast doubt on the idea that supplements are never at-issue, with (7) and (8) showing that they can be denied and (9) showing that they can address the QUD.

In view of the data, SU follows Amaral et al. (2007), AnderBois et al. (2010, 2015), Koev (2012, 2014), and Martin (2013) by claiming that supplements should be receive an incremental interpretation in a dynamic system, an interpretation no different from the one afforded to regular, nonsupplement content. SU also adopts the position advocated by Schlenker (2010, ms) that supplements should be represented in a way that allows them to interact with other content at the level of scope. SU's account of supplement scope strengthens Nouwen's (2014) claim that "[t]he scope of the appositive is always at least as wide as that of its anchor, never narrower" by making a supplement's scope *exactly* that of its anchor. As mentioned above, cases where a supplement appears to take wider scope than its anchor, such as (5c), are analyzed as instances of telescoping in the SU model.

SU's account is *unidimensional*, following Kubota and Uegaki (2009), AnderBois et al. (2010, 2015), Koev (2012, 2014), Murray (2014), and Schlenker (ms). But unlike many of these other accounts, supplements, in SU, participate in scope interactions with operators in exactly the same way as other content. They also update the discourse context via the same method as all other content. As for how it is that supplements *project* (Tonhauser et al., 2013), escaping the effects of semantic operators, SU treats projection as an epiphenomenon of supplement scope: supplements anchored to proper names project because their anchors scope widest; supplements exhibit a preference for surface scope in preference to inverse scope because this preference is in effect for their anchors. As a result, SU's treatment of supplements is more empirically adequate than AnderBois et al.'s (2015), who only allow "*one*-asides" like those in (3) to scope narrow, and therefore cannot account for narrow scoping supplements like those in (1), (4), or (5).

The central conceptual difference between SU's account of supplements and nearly all others is that SU treats supplements as just an extra bit of content attached to the restrictor of a generalized quantifier (GQ). By contrast, other accounts claim that a supplement is endowed with special properties that require it to project, forcing widest scope on its anchors in the process. For SU, projection does not arise because of some special property of supplements, but is rather due to the scope of their anchors, which are themselves influenced by independent mechanisms such as anaphora resolution, the preference for surface over inverse scope, and discourse binding.

A notable consequence of SU's approach is that projection and at-issueness are dissociated, so that whether or not a supplement projects is independent of whether it can be directly denied or address the QUD. Instead of prohibiting supplements from being denied, SU follows Koev (2012), AnderBois et al. (2010, 2015), and Schlenker (ms) in treating supplement deniability as being based on recency of mention, cf. (7)–(8). For SU, a supplement constitutes a separate discourse update, and more recent updates are more salient in the same way that more recent antecedents for anaphora are more salient, following Ginzburg (2012). Anaphoric links between supplements and other content, in SU's system, function in exactly the same way as all other instances of anaphora.

3 Analyzing supplements in a two-component categorial syntax

The syntax-semantics interface for supplements proposed below is based closely on SU's supplement semantics. It represents a very minor extension of the grammar formalism presented by Martin and Pollard (2014) because, like SU's semantic account, the only machinery specific to supplements is the lexical entries for the comma intonation, the distinct pattern of intonational pauses that surrounds a supplement in spoken English. Everything else is handled by independently motivated aspects of the grammar, which models anaphora resolution in a dynamic setting, and models quantifier scope by treating all noun phrases as GQs (Barwise and Cooper, 1981; Keenan and Stavi, 1986). Supplement syntax is treated in a parallel way to SU's semantics, with a model of the comma intonation that piggybacks a supplement's syntactic material onto its anchor. Before delving into the analysis, I first give an introduction to Martin and Pollard's two-component grammar.

3.1 Dynamic Categorial Grammar

The formalism developed by Martin and Pollard (2012a,b), Martin (2013), and Martin and Pollard (2014), called *Dynamic Categorial Grammar*, follows the tradition in categorial grammar of dividing the syntactic labor between multiple components (Oehrle, 1994; de Groote, 2001; Muskens, 2007; Mihaliček, 2012; Worth, 2014). In this syntax, one component (*tectogrammar*) handles the combinatorics, and a separate one (*phenogrammar*) handles surface form. This grammar formalism derives *signs*, triples of the form

$$\varphi \,;\, \tau \,;\, \sigma \,,$$

where φ is the phenogrammatical component, τ the tectogrammar, and σ the semantics.

The tectogrammar models syntactic combinatorics in a very streamlined version of linear propositional logic with basic types like NP, N, S, and D representing the atomic syntactic categories of noun phrases, common nouns, sentences, and discourses, respectively. More complex tecto types are formed by the binary linear implication connective $\multimap$. The phenogrammar's surface forms are derived by a simple type theory with a single nonlogical type, s, of *strings*, and an associative binary concatenation operator $\cdot : \mathrm{s} \to \mathrm{s} \to \mathrm{s}$ with a two-sided identity $\mathbf{e}$ (the *empty string*).

The semantics is a compositional, dynamic theory in the tradition of Muskens (1996), Beaver (2001), and de Groote (2006). It is implemented in dependent type theory with certain types parameterized by the natural number type. Its basic types include e (*entities*), p (*propositions*), and n (*natural numbers*). Discourse contexts are modeled via the type $\mathrm{c}_n =_{\mathrm{def}} \mathrm{e}^n \to \mathrm{p}$, functions from n-ary entity vectors to propositions, and the meanings of declarative utterances are modeled by the type $\mathrm{k}_n =_{\mathrm{def}} \Pi_{c:\mathrm{c}_m}.\mathrm{c}_{m+n}$ of *contents*, functions from contexts to contexts that introduce n discourse referents, modeled by the natural numbers. Contents are promoted to *updates*, also of type $\Pi_{c:\mathrm{c}_m}.\mathrm{c}_{m+n}$, by the *context change* function cc. The type $\mathrm{d}_m =_{\mathrm{def}} \Pi_{n:\mathrm{n}}\Pi_{c:\mathrm{c}_{>n}}.\mathrm{c}_{|c|+m}$ models dynamic properties, where $|c|$ is the arity of c, the length of its input vector. This type is essentially the type of functions from a natural number n (i.e., discourse referent) to a content introducing m discourse referents, with the requirement that the resulting content's input context have a slot for n (via the type $\mathrm{c}_{>n}$). SU gives much more detail about the dynamic semantic component.

Figure 1 gives a natural deduction presentation of the rules of the grammar. In all of these rules, both the types for the phenogrammatical terms and the types for the semantic terms are omitted for clarity. However, with some knowledge of the types of variables

$$\vdash a\,;B\,;c \quad \text{(Lexical Entry)}$$

$$x\,;A\,;y \vdash x\,;A\,;y \quad \text{(Trace)}$$

$$\frac{\Gamma, x\,;A\,;y \vdash a\,;B\,;c}{\Gamma \vdash (\lambda_x a)\,;A \multimap B\,;(\lambda_y c)} \text{(Hypothetical Proof)}$$

$$\frac{\Gamma \vdash f\,;B \multimap C\,;g \qquad \Delta \vdash a\,;B\,;c}{\Gamma, \Delta \vdash (f\,a)\,;C\,;(g\,c)} \text{(Modus Ponens)}$$

$$\vdash \mathbf{e}\,;\mathrm{D}\,;\lambda_{c:\mathrm{c}}.c \quad \text{(Empty Discourse)}$$

$$\frac{\vdash a\,;\mathrm{D}\,;u \qquad \vdash b\,;\mathrm{S}\,;k}{\vdash a \cdot b\,;\mathrm{D}\,;u \circ (\mathsf{cc}\,k)} \text{(Continue)}$$

Figure 1: A natural deduction formulation of the rules of Dynamic Categorial Grammar. The symbols a, b, c, f, and g are metavariables over terms, x and y are metavariables over type-theoretic variables, Γ and Δ range over variable contexts, A, B, C, range over tecto types, u over updates, and k over contents.

and constants, the relevant types for terms in applications of the grammar rules can be inferred. The lexicon is comprised of a set of instantiations of the *Lexical Entry* rule. *Trace* is used to introduce placeholders for long-distance dependencies that can later be bound via *Hypothetical Proof*. Local dependencies are modeled by the *Modus Ponens* rule. The final two rules handle the initiation and continuation of discourses, but I do not discuss them in detail here since they are not central to the analysis of supplements.

To illustrate, the lexicon required to model the simple example

(10) Some cyclist won the Tour de France

is given below.

(11) $\vdash \lambda_{sf}.f\,(\mathrm{some} \cdot s)\,;\mathrm{N} \multimap \mathrm{QP}\,;\textsc{a}$

(12) $\vdash \mathrm{cyclist}\,;\mathrm{N}\,;\textsc{cyclist}$

(13) $\vdash \lambda_s.s \cdot \mathrm{won} \cdot \mathrm{the} \cdot \mathrm{tour} \cdot \mathrm{de} \cdot \mathrm{france}\,;\mathrm{NP} \multimap \mathrm{S}\,;\textsc{win-tdf}$

Here, the dynamic indefinite determiner $\textsc{a} : \mathrm{d}_1 \to \mathrm{k}$ and dynamic properties $\textsc{cyclist} : \mathrm{d}_1$ and $\textsc{win-tdf} : \mathrm{d}_1$ are defined as in SU, the VP *won the Tour de France* is defined syncategorematically for simplicity, and QP abbreviates $(\mathrm{NP} \multimap \mathrm{S}) \multimap \mathrm{S}$. The analysis of (10) starts with the lexical entries in (11)–(13) and invokes the Modus Ponens rule twice, as follows. First, *some* takes *cyclist* as its argument:

(14)
$$\frac{(11) \qquad\qquad (12)}{\vdash \lambda_f.f\,(\mathrm{some} \cdot \mathrm{cyclist})\,;\mathrm{QP}\,;\textsc{a cyclist}}$$

And next, the QP *some cyclist* takes the verb phrase as its argument.

(15)
$$\frac{(14) \qquad\qquad (13)}{\vdash \mathrm{some} \cdot \mathrm{cyclist} \cdot \mathrm{won} \cdot \mathrm{the} \cdot \mathrm{tour} \cdot \mathrm{de} \cdot \mathrm{france}\,;\mathrm{S}\,;\textsc{a cyclist win-tdf}}$$

The proofs above observe some conventions that are maintained throughout this paper: rule labels have been omitted, extra parentheses have been elided when their absence

cannot trigger ambiguity, and β-reduction has been performed where possible, with the β-normal form substituted for the corresponding redex. I also adopt many of SU's and Martin and Pollard's (2014) notational conventions, for example, if $M : (A \to B) \to C$ and $\lambda_x.N : A \to B$, I write $M_x.N$ to abbreviate $(M\, \lambda_x.N)$.

3.2 Basic supplement syntax

In SU's semantics, supplement scope is handled by the independent mechanism of hypothetical proof, and anaphoric links between supplements and other content are also taken care of by built-in machinery. Since, in SU's account, supplements have no special property requiring their content to project, they are modeled via a lexical entry defining $\textsc{comma} : (\mathrm{d}_1 \to \mathrm{k}) \to \mathrm{d}_1 \to \mathrm{d}_1 \to \mathrm{k}$, the meaning of the comma intonation, as

$$\textsc{comma} =_{\mathrm{def}} \lambda_{QDE}.(Q\, D)\ \textsc{and}\ (\textsc{the}\, D\, E)\ . \tag{16}$$

As encoded in (16), the comma intonation's meaning triggers a type of QP modification. Its semantics takes a dynamic GQ Q, of type $\mathrm{d}_1 \to \mathrm{k}$, and two dynamic properties D and E, applying Q to D as its scope, and conjoining the result $(Q\, D)$ with the result of applying E to the uniquely most salient referent entailed to have the property D, obtained by the anaphoric determiner THE.

Giving a syntax-semantics interface for supplements only requires extending (16) with tecto and pheno terms, as follows:

$$\vdash \lambda_{fsg}.g\,(f\,\lambda_t.t \cdot (\mathsf{comma}\, s))\ ;\ \mathrm{QP} \multimap \mathrm{Pred} \multimap \mathrm{QP}\ ;\ \textsc{comma} \tag{17}$$

Here, Pred is the type of predicativized GQs, and the pheno term $\mathsf{comma} : \mathrm{s} \to \mathrm{s}$, left unanalyzed here, represents the phenogrammatical contribution of the comma intonation.

To demonstrate how the lexical entry in (17) works, consider how it models the following example, an extension of (10).

(18) Some cyclist, a doper, won the Tour de France. (Martin, in press, (1))

As a preliminary, the lexical entry for SU's *predicativizer* $\textsc{pred} =_{\mathrm{def}} \lambda_{Qn}.Q_m.m\ \textsc{equals}\ n$ is defined as

$$\vdash \lambda_f.f\,\lambda_s.s\ ;\ \mathrm{QP} \multimap \mathrm{Pred}\ ;\ \textsc{pred}\ . \tag{19}$$

This lexical entry takes a QP to a predicative, in keeping with its semantics PRED, which transforms an GQ into a dynamic property. The QP's pheno term is simply fed the identity function, reducing it to a string. Adding the lexical entries corresponding to *a* and *doper* allows an analysis of the supplement in (18).

$$\vdash \lambda_{sf}.f\,(\mathrm{a} \cdot s)\ ;\ \mathrm{N} \multimap \mathrm{QP}\ ;\ \textsc{a} \tag{20}$$

$$\vdash \mathrm{doper}\ ;\ \mathrm{N}\ ;\ \textsc{doper} \tag{21}$$

The lexical entry for *a* in (20) is identical to the one in (11) except that the string a replaces some. The dynamic property DOPER is as defined in SU. The analysis of (18) starts by deriving the supplement. First, the GQ *a doper* is converted to a predicative in a straightforward way based on lexical entries (19)–(21). The GQ is derived as follows:

$$\frac{(20) \qquad\qquad (21)}{\vdash \lambda_f.f\,(\mathrm{a} \cdot \mathrm{doper})\ ;\ \mathrm{QP}\ ;\ \textsc{a doper}} \tag{22}$$

And next, the predicativizer takes the derived GQ as its argument.

(23)
$$\frac{(19) \qquad (22)}{\vdash \mathrm{a} \cdot \mathrm{doper} \,;\, \mathrm{Pred} \,;\, \textsc{pred a doper}}$$

The comma intonation then combines the QP *some cyclist*, derived above in (14), with the sign representing *a doper*:

(24)
$$\frac{(17) \qquad (14)}{\vdash \lambda_{sg}.g\,(\mathrm{some} \cdot \mathrm{cyclist} \cdot (\mathsf{comma}\; s)) \,;\, \mathrm{Pred} \multimap \mathrm{QP} \,;\, \textsc{comma}\;(\textsc{a cyclist})}$$

Next, the supplement is integrated to form a new QP.

(25)
$$\frac{(24) \qquad (23)}{\vdash \lambda_{g}.g\,(\mathrm{some} \cdot \mathrm{cyclist} \cdot \mathsf{comma}\,(\mathrm{a} \cdot \mathrm{doper})) \,;\, \mathrm{QP} \,;\, \textsc{comma}\;(\textsc{a cyclist})\;(\textsc{pred a doper})}$$

Finally, the verb phrase *won the Tour de France* is taken by the derived QP as its argument.

(26)
$$\frac{(25) \qquad (13)}{\begin{array}{c}\vdash \mathrm{some} \cdot \mathrm{cyclist} \cdot \mathsf{comma}\,(\mathrm{a} \cdot \mathrm{doper}) \cdot \mathrm{won} \cdot \mathrm{the} \cdot \mathrm{tour} \cdot \mathrm{de} \cdot \mathrm{france} \,;\, \mathrm{S} \,; \\ \textsc{comma}\;(\textsc{a cyclist})\;(\textsc{pred a doper})\;\textsc{win-tdf}\end{array}}$$

The sign derived in (26) contains the correct syntactic analysis of (18), with tecto type S of sentences, and the intonational pauses that typically surround the supplement in spoken English demarcated by comma.

By (16), the meaning assigned to (18) reduces as follows.

$$\begin{array}{l}\textsc{comma}\;(\textsc{a cyclist})\;(\textsc{pred a doper})\;\textsc{win-tdf} \\ = (\textsc{a cyclist}\;(\textsc{pred a doper}))\;\textsc{and}\;(\textsc{the}\;(\textsc{pred a doper})\;\textsc{win-tdf})\end{array}$$

For any two contents h and k, SU's theorem A.2 establishes that the conjoined update $\mathsf{cc}\,(h\;\textsc{and}\;k)$ is equivalent to the parataxis of the component updates $(\mathsf{cc}\,h) \circ (\mathsf{cc}\,k)$. By this equivalence, we have

$$\begin{array}{l}\mathsf{cc}\,(\textsc{a cyclist}\;(\textsc{pred a doper}))\;\textsc{and}\;(\textsc{the}\;(\textsc{pred a doper})\;\textsc{win-tdf}) \\ = (\mathsf{cc}\,(\textsc{a cyclist}\;(\textsc{pred a doper}))) \circ (\mathsf{cc}\,(\textsc{the}\;(\textsc{pred a doper})\;\textsc{win-tdf}))\end{array}$$

for the meaning derived in (26). This equivalence is important because it shows how the comma intonation's meaning, for (18), separates out the contribution of the supplement and the main-clause content into different updates. This separation is how the account captures projection, as is made more evident in the examples with operators in §3.3, below.

3.2.1 Quantified supplements

Supplements anchored by GQs whose determiner is a true quantifier are known to be problematic, as in the following variant of (18) in

(27) # Every cyclist, a doper, won the Tour de France.

The lexical entry for *Every* straightforwardly parallels the one for the indefinite determiner in (20), with EVERY as defined in SU:

(28)
$$\vdash \lambda_{sf}.f\,(\mathrm{every} \cdot s) \,;\, \mathrm{N} \multimap \mathrm{QP} \,;\, \textsc{every}$$

Splicing this lexical entry into the proof in (22) in place of (20) gives the following sign:

$$\vdash \text{every} \cdot \text{cyclist} \cdot \mathsf{comma}\,(\text{a} \cdot \text{doper}) \cdot \text{won} \cdot \text{the} \cdot \text{tour} \cdot \text{de} \cdot \text{france}\,;\,\text{S}\,;$$
$$\textsc{comma}\,(\textsc{every cyclist})\,(\textsc{pred a doper})\,\textsc{win-tdf}$$

But the semantics of this sign is infelicitous, because it reduces to

$$(\textsc{every cyclist}\,(\textsc{pred a doper}))\ \textsc{and}\ (\textsc{the}\,(\textsc{pred a doper})\,\textsc{win-tdf})\,.$$

Since the discourse referent introduced for the doping cyclist gets trapped in the scope of *Every*, it cannot be accessed by the anaphoric GQ $\textsc{the}\,(\textsc{pred a doper})$. As for how the account generates (5c), it is treated in SU as an instance of telescoping (Roberts, 1989, 2005), where a discourse referent in the scope of a quantifier can be interpreted outside its scope in certain tightly constrained cases.

3.2.2 Other supplement types

This account can deal with other types of supplements besides the nominal appositive in (18). In fact, extending it to nonrestrictive relative clauses (NRRCs) and *as*-parentheticals (APs) is fairly trivial. When they take a predicative as their argument, both of these supplement types can be seen as prefixing a nominal appositive, as illustrated in

(29) Lance, $\left\{ \begin{array}{l} \text{as} \\ \text{who is} \end{array} \right\}$ a doper, got sanctioned by the UCI.

For the case of APs, the necessary extension is the following lexical entry, which closely resembles the predicativizer (19), with the exception that the word *as* is prefixed.

(30) $$\vdash \lambda_f.\text{as} \cdot (f\ \lambda_s.s)\,;\,\text{QP} \multimap \text{Pred}\,;\,\textsc{pred}$$

For NRRCs like the relevant variant of (29), two lexical entries are needed, one for *who* and one for *is* (or the relevant morphological variant of the copula).

(31) $$\vdash \lambda_f.\text{is} \cdot (f\ \lambda_s.s)\,;\,\text{QP} \multimap \text{Be}_{\text{pred}}\,;\,\textsc{pred}$$

(32) $$\vdash \lambda_s.\text{who} \cdot s\,;\,\text{Be}_{\text{pred}} \multimap \text{Pred}\,;\,\lambda_D.D$$

The lexical entry (31) is very similar to the one for *as* (30): it simply prefixes *is* to the pheno of its QP argument. The other difference is that the resulting tecto type is Be_{pred}, the type of copular predicatives.

Taken together, these lexical entries allow a model of both variants of (29). For the *as* variant, lexical entry (30) is simply applied to the QP *a doper*:

(33)
$$\frac{(30) \qquad\qquad (22)}{\vdash \text{as} \cdot \text{a} \cdot \text{doper}\,;\,\text{Pred}\,;\,(\textsc{pred a doper})}$$

The NRRC variant with *who is* requires one extra proof step, but is also straightforward.

(34)
$$\frac{(32) \qquad \dfrac{(31) \qquad\qquad (22)}{\vdash \text{is} \cdot \text{a} \cdot \text{doper}\,;\,\text{Be}_{\text{pred}}\,;\,(\textsc{pred a doper})}}{\vdash \text{who} \cdot \text{is} \cdot \text{a} \cdot \text{doper}\,;\,\text{Pred}\,;\,(\textsc{pred a doper})}$$

Either of the supplements derived in (33) and (34) could be passed to the comma intonation as its predicative argument.

The motivation for the structure of the lexical entries (30)–(31) is to ensure they not only generate the proper surface form and semantics, but that they also interact in ways that do not generate unobserved surface forms. The lexical entry for *as* in (30) rules out the following:

* as who is a doper

* as as as a doper

The reason is that *as* cannot take a NRRC as its argument due to a tecto type mismatch, and similarly, it cannot take another *as*-parenthetical. Only QP arguments are allowed for *as* by (30). The lexical entries pertaining to NRRCs in (31) and (32) together rule out these unobserved supplements:

* who is as a doper

* who as a doper

* who a doper

* who who who is a doper

As in the case of *as*-parentheticals, the grammar makes these predictions because of type mismatches: *is* can only take QP arguments, and *who* can only take arguments of type Be_{pred}.

A NRRC can also be formed from a nonpredicative complement, as in (1a), (2), and (4)–(9). The lexical entries for the on-restrictive relativizers *who* and *which* are straightforward.

(35) $\vdash \lambda_f.\text{who} \cdot (f\ \mathbf{e})\ ;\ (\text{NP} \multimap \text{S}) \multimap \text{Pred}\ ;\ \textsc{pred}$

(36) $\vdash \lambda_f.\text{which} \cdot (f\ \mathbf{e})\ ;\ (\text{NP} \multimap \text{S}) \multimap \text{Pred}\ ;\ \textsc{pred}$

As these lexical entries show, the nonrestrictive relativizers both take a property to a predicative, and their semantic effect is the same as the predicativizer (19). For example, the NRRC in *Lance, who won the Tour de France, is from Texas* is derived as follows:

(37)
$$\frac{(35) \qquad (13)}{\vdash \text{who} \cdot \mathbf{e} \cdot \text{won} \cdot \text{the} \cdot \text{tour} \cdot \text{de} \cdot \text{france}\ ;\ \text{Pred}\ ;\ (\textsc{pred}\ \textsc{win-tdf})}$$

A further demonstration of nonrestrictive relativizers is given in §3.5.

As an aside, I briefly note a nonviable alternative to the treatment of the comma intonation given here pointed out by Yusuke Kubota (personal communication). Note that it would be possible to give an alternative lexical entry for the comma intonation in which the supplement attached only to the QP's restrictor property, not to the entire QP itself. However, such a lexical entry would not be empirically adequate on the basis of examples with conjoined anchors like

(38) That man and woman, who have been married for years, seem like a happy couple.

Applying the property signaled by the supplement *who have been married for years* only to *woman* would give truth conditions that are out of line with intuitions about the meaning of (38). This account also accords with Nouwen's (2014) observation that the scope of an appositive is always at least as wide as that of its anchor. For example, (39) cannot mean that there is some professor such that if that professor is famous and publishes a book, he will make a lot of money.

(39) If a professor, a famous one, publishes a book, he will make a lot of money. (Nouwen, 2014, (29))

That is, a successful account of (39) must not allow the supplement *a famous one* to scope narrow relative to the conditional while *a professor* scopes wider than it. This proposal avoids both of these issues by modeling supplements as attaching to QPs, rather than just to their restrictor property.

3.3 Supplement projection and nonprojection

To see how the account functions for a projecting supplement, consider the variation of (18) in

(40) It's not true that Lance, a doper, won the Tour de France.

Examples like these show a supplement inside a negated main clause that nevertheless projects: (40) implies that Lance is a doper even though the information that he won the Tour is negated.

Analyzing this example requires some additional lexical entries for the proper name *Lance* and for sentential negation.

(41) $\vdash \lambda_f.f\,\text{lance}\,;\,\text{QP}\,;\,\textsc{lance}$

(42) $\vdash \lambda_s.\text{it}\cdot\text{is}\cdot\text{not}\cdot\text{true}\cdot\text{that}\cdot s\,;\,\text{S}\multimap\text{S}\,;\,\textsc{not}$

Here, *Lance* receives a GQ treatment, with the tecto type QP of quantifier phrases, and semantics $\textsc{lance} =_{\text{def}} \textsc{the named-lance}$, which passes to its scope the unique antecedent entailed to have the name "Lance". The sentential negation is defined syncategorematically for simplicity; its semantics NOT is SU's dynamic negation, which both negates its argument's content and traps any discourse referents introduced in its scope, rendering them inaccessible to subsequent anaphora.

The derivation of the sign corresponding to (40) first generates the QP *Lance, a doper* by combining lexical entry (41) with the proof of the supplement *a doper* in (23):

(43)
$$\dfrac{\dfrac{(17) \qquad (41)}{\vdash \lambda_{sg}.g\,(\text{lance}\cdot(\mathsf{comma}\,s))\,;\,\text{Pred}\multimap\text{QP}\,;\,\textsc{comma lance}} \qquad (23)}{\vdash \lambda_g.g\,(\text{lance}\cdot\mathsf{comma}\,(\text{a}\cdot\text{doper}))\,;\,\text{QP}\,;\,\textsc{comma lance}\,(\textsc{pred a doper})}$$

But in order to get the observed projective reading for (40), this QP must outscope the negation. So rather than immediately combine the sign in (43) with the verb phrase, we first derive the a version of the entire utterance with a slot for a missing QP.

To accomplish this, we first pass a placeholder argument to the verb phrase using Trace.

(44)
$$\dfrac{(13) \qquad s\,;\,\text{NP}\,;\,n\vdash s\,;\,\text{NP}\,;\,n}{s\,;\,\text{NP}\,;\,n\vdash s\cdot\text{won}\cdot\text{the}\cdot\text{tour}\cdot\text{de}\cdot\text{france}\,;\,\text{S}\,;\,\textsc{win-tdf}\,n}$$

Next, the sentential negation takes the sign derived in (44) as its argument, and then the trace is withdrawn via Hypothetical Proof (here, won abbreviates the full verb phrase pheno won · the · tour · de · france).

(45)
$$\dfrac{\dfrac{(42) \qquad (44)}{s\,;\,\text{NP}\,;\,n\vdash\text{it}\cdot\text{is}\cdot\text{not}\cdot\text{true}\cdot\text{that}\cdot s\cdot\text{won}\,;\,\text{S}\,;\,\textsc{not}\,(\textsc{win-tdf}\,n)}}{\vdash\lambda_s.\text{it}\cdot\text{is}\cdot\text{not}\cdot\text{true}\cdot\text{that}\cdot s\cdot\text{won}\,;\,\text{NP}\multimap\text{S}\,;\,\lambda_n.\textsc{not}\,(\textsc{win-tdf}\,n)}$$

Finally, the QP *Lance, a doper* is applied to the sign in (45) to produce the full sign representing (40).

(46)
$$\dfrac{(43) \qquad (45)}{\begin{array}{l}\vdash\text{it}\cdot\text{is}\cdot\text{not}\cdot\text{true}\cdot\text{that}\cdot\text{lance}\cdot\mathsf{comma}\,(\text{a}\cdot\text{doper})\cdot\text{won}\cdot\text{the}\cdot\text{tour}\cdot\text{de}\cdot\text{france}\,;\,\text{S}\,;\\ \quad(\textsc{comma lance}\,(\textsc{pred a doper}))_n.\textsc{not}\,(\textsc{win-tdf}\,n)\end{array}}$$

Expanding the semantics derived in (46) shows how the supplement escapes negation:

$$\begin{aligned}&\textsc{comma lance}\,(\textsc{pred a doper})\,\lambda_n.\textsc{not}\,(\textsc{win-tdf}\,n)\\&=(\textsc{lance}\,(\textsc{pred a doper}))\,\textsc{and}\,(\textsc{the}\,(\textsc{pred a doper})\,\lambda_n.\textsc{not}\,(\textsc{win-tdf}\,n))\end{aligned}$$

Because of the way the comma intonation is defined, only Lance's winning is in the scope of negation, but his being a doper survives unscathed, that is, it projects. As an explanation for why the projective scoping is preferred to the one with negation wide, in which both the supplement and the main clause content are effected by negation, SU appeals to the preference for definites such as proper names to scope as closely as possible to their antecedent's site. This means that the projective reading of (40) is preferred because *Lance*'s antecedent, the unique discourse referent entailed to be named "Lance", does not occur in the utterance itself.

Things are different when the supplement is anchored by an indefinite, which can scope narrow relative to operators, as illustrated in (3). In the case of the variant of (18) in

(47) It is not true that some cyclist, a doper, won the Tour de France,

the narrow-scope reading is the one that might arise by default in situations where the Tour had for some reason been canceled. To generate this reading, sentential negation is simply applied to the sign derived for (18) in (26), as follows.

(48)
$$\frac{(42)\qquad\qquad(26)}{\begin{array}{l}\vdash \text{it}\cdot\text{is}\cdot\text{not}\cdot\text{true}\cdot\text{that}\cdot\text{some}\cdot\text{cyclist}\cdot\mathsf{comma}\,(\text{a}\cdot\text{doper})\cdot\text{won}\,;\,\text{S}\,;\\ \quad\textsc{not}\,(\textsc{comma}\,(\textsc{a cyclist})\,(\textsc{pred a doper})\,\textsc{win-tdf})\end{array}}$$

(Here again, won abbreviates the full verb phrase in (47).) In this narrow-scope reading, both the supplement and the main clause are negated, so this meaning is equivalent to *It's not true that some doping cyclist won the Tour*. Of course, the wide-scope, projective reading is available for this case, too:

(49)
$$\frac{(25)\qquad\qquad(45)}{\begin{array}{l}\vdash \text{it}\cdot\text{is}\cdot\text{not}\cdot\text{true}\cdot\text{that}\cdot\text{some}\cdot\text{cyclist}\cdot\mathsf{comma}\,(\text{a}\cdot\text{doper})\cdot\text{won}\,;\,\text{S}\,;\\ \quad(\textsc{comma}\,(\textsc{a cyclist})\,(\textsc{pred a doper}))_n.\textsc{not}\,(\textsc{win-tdf}\,n)\end{array}}$$

Similar to (46), above, this reading is equivalent to *Some cyclist is a doper, and this cyclist didn't win the Tour*. Due to this flexibility with respect to supplement scope, this account can capture the full range of readings available for (3)–(5), in contrast with other accounts, which rule out narrow-scope readings for supplements.

3.4 Utterance-final supplements

This syntax derives signs with supplements both in utterance-medial and utterance-final position. To generate a sign for (50a), the only required extension to the grammar is a lexical entry for the verb *met*.

(50) a. Some cyclist, a doper, met Lance.

b. Some cyclist met Lance, a doper.

The required lexical entry is

(51) $$\vdash \lambda_{st}.t\cdot\text{met}\cdot s\,;\,\text{NP}\multimap\text{NP}\multimap\text{S}\,;\,\textsc{met}\,,$$

where MET : d_2 is the binary dynamic property corresponding to the verb *met*.

Deriving (50a) starts by providing the verb with its subject, after first generating a trace for the object argument.

$$\dfrac{\dfrac{}{s\,;\,\mathrm{NP}\,;\,m \vdash \lambda_t.t \cdot \mathrm{met} \cdot s\,;\,\mathrm{NP} \multimap \mathrm{S}\,;\,\textsc{met}\,m}\,(25)}{s\,;\,\mathrm{NP}\,;\,m \vdash \mathrm{some} \cdot \mathrm{cyclist} \cdot \mathsf{comma}\,(\mathrm{a} \cdot \mathrm{doper}) \cdot \mathrm{met} \cdot s\,;\,\mathrm{S}\,;\,\textsc{comma}\,(\textsc{a cyclist})\,(\textsc{pred a doper})\,(\textsc{met}\,m)}\,(51) \tag{52}$$

Next, the trace is withdrawn, and the object takes scope:

$$\dfrac{\dfrac{}{\vdash \lambda_s.\mathrm{some} \cdot \mathrm{cyclist} \cdot \mathsf{comma}\,(\mathrm{a} \cdot \mathrm{doper}) \cdot \mathrm{met} \cdot s\,;\,\mathrm{NP} \multimap \mathrm{S}\,;\,\lambda_m.\textsc{comma}\,(\textsc{a cyclist})\,(\textsc{pred a doper})\,(\textsc{met}\,m)}\,(52)}{\vdash \mathrm{some} \cdot \mathrm{cyclist} \cdot \mathsf{comma}\,(\mathrm{a} \cdot \mathrm{doper}) \cdot \mathrm{met} \cdot \mathrm{lance}\,;\,\mathrm{S}\,;\,\textsc{lance}_m.(\textsc{comma}\,(\textsc{a cyclist})\,(\textsc{pred a doper})\,(\textsc{met}\,m))}\,(41) \tag{53}$$

Note that in this derived meaning of (50a), the supplement's content is separated off from the main clause content by the comma intonation in the same way as for (18).

To model the utterance-final supplement in (50b), some work remains to be done in order for the syntax to line up with SU's account of supplement deniability. In SU, a supplement's deniability is influenced by its recency in terms of the temporal sequence of updates comprising the discourse. So for (50b), just as for (7) and (8), the supplement *a doper* can be more easily denied because it comes *after* the main clause content. However, the lexical entry for the comma intonation in (17) expects the verb phrase's contribution to come last, so the supplement would precede another update.

Giving a lexical entry for supplements in utterance-final position is as straightforward as changing the order of the arguments. The variant entry required to model (50b) is

$$\vdash \lambda_{fgs}.g\,(f\,\lambda_t.t \cdot (\mathsf{comma}\,s))\,;\,\mathrm{QP} \multimap (\mathrm{NP} \multimap \mathrm{S}) \multimap \mathrm{Pred} \multimap \mathrm{S}\,;\,\textsc{comma}\,. \tag{54}$$

The difference between (54) and (17) is simply that the verb phrase argument is taken before the supplement argument. The semantics remains the same; only the surface order of the supplement in relation to the verb phrase is changed, which has the effect of making the supplement's update come after the verb phrase's.

To see how this variant lexical entry differs from (17) in practice, consider the model of (50b) it produces. Since the proper name *Lance* takes widest scope, the proof starts by providing *Lance* to the comma intonation.

$$\dfrac{}{\vdash \lambda_{gs}.g\,(\mathrm{lance} \cdot (\mathsf{comma}\,s))\,;\,(\mathrm{NP} \multimap \mathrm{S}) \multimap \mathrm{Pred} \multimap \mathrm{S}\,;\,\textsc{comma lance}}\,(54)\;(41) \tag{55}$$

Next, the verb phrase is derived with a hypothesized slot for its object.

$$\dfrac{\dfrac{\dfrac{}{s\,;\,\mathrm{NP}\,;\,m \vdash \lambda_t.t \cdot \mathrm{met} \cdot s\,;\,\mathrm{NP} \multimap \mathrm{S}\,;\,\textsc{met}\,m}\,(51)}{s\,;\,\mathrm{NP}\,;\,m \vdash \mathrm{some} \cdot \mathrm{cyclist} \cdot \mathrm{met} \cdot s\,;\,\mathrm{S}\,;\,(\textsc{a cyclist})_n.\textsc{met}\,m\,n}\,(14)}{\vdash \lambda_s.\mathrm{some} \cdot \mathrm{cyclist} \cdot \mathrm{met} \cdot s\,;\,\mathrm{NP} \multimap \mathrm{S}\,;\,\lambda_m.(\textsc{a cyclist})_n.\textsc{met}\,m\,n} \tag{56}$$

Then the sign derived in (55) takes the derived verb phrase as its argument.

$$\dfrac{}{\vdash \lambda_s.\mathrm{some} \cdot \mathrm{cyclist} \cdot \mathrm{met} \cdot \mathrm{lance} \cdot (\mathsf{comma}\,s)\,;\,\mathrm{Pred} \multimap \mathrm{S}\,;\,\textsc{comma lance}_m.(\textsc{a cyclist})_n.\textsc{met}\,m\,n}\,(55)\;(56) \tag{57}$$

Finally, the supplement provides the last argument for the comma intonation.

$$\frac{\text{(57)} \qquad\qquad \text{(23)}}{\vdash \text{some} \cdot \text{cyclist} \cdot \text{met} \cdot \text{lance} \cdot \mathsf{comma}\,(\text{a} \cdot \text{doper})\,;\,\text{S}\,;\ \textsc{comma}\,(\textsc{lance}_m.(\textsc{a cyclist})_n.\textsc{met}\,m\,n)\,(\textsc{pred a doper})} \tag{58}$$

Expanding the semantics of this final sign representing (50b) illustrates how the supplement is more deniable than its counterpart in (50a)

$$\begin{aligned}&\textsc{comma}\,(\textsc{lance}_m.(\textsc{a cyclist})_n.\textsc{met}\,m\,n)\,(\textsc{pred a doper})\\ &= (\textsc{lance}_m.(\textsc{a cyclist})_n.\textsc{met}\,m\,n)\ \textsc{and}\\ &\quad\ \textsc{the}\,(\lambda_m(\textsc{a cyclist})_n.\textsc{met}\,m\,n)\,(\textsc{pred a doper})\end{aligned}$$

This dynamic meaning is paraphrasable by *Lance was met by a cyclist, and the person who was met by the cyclist is a doper*. In SU's account, the supplement is the most recent discourse update, and is therefore easier to target via denial. I also propose that the presence of an utterance boundary disambiguates between the medial comma intonation in (17) and the final one in (54), but leave the implementation of the details beyond the scope of this paper.

3.5 Supplements and anaphora

In this account, anaphora between a supplement and other content is just a special case of anaphora more generally, which SU's dynamic semantics is designed to handle. A good demonstration is the example

(59) Melanie$_i$, who bought herself$_i$ a car$_j$, drives it$_j$. (Simplification of Martin in press, (52))

In (59), the anaphoric link labeled i consists of a pronoun inside a supplement with an antecedent outside, while the one labeled j has the antecedent within a supplement but the pronoun outside.

To model (59), the lexicon needs to be extended with entries for *Melanie*, the common noun *car*, the verbs *bought* and *drives*, and the pronouns *herself* and *it*. Pronouns, in this account, are treated as GQs, as are all noun phrases, and so giving lexical entries for both the pronouns and the name *Melanie* is straightforward.

$$\vdash \lambda_f.f\ \text{melanie}\ ;\ \text{QP}\ ;\ \textsc{melanie} \tag{60}$$

$$\vdash \lambda_f.f\ \text{herself}\ ;\ \text{QP}\ ;\ \textsc{herself} \tag{61}$$

$$\vdash \lambda_f.f\ \text{it}\ ;\ \text{QP}\ ;\ \textsc{it} \tag{62}$$

As do all proper names in this account, *Melanie* is anaphoric, following Beaver (2001) and Martin (2013, in press): $\textsc{melanie} =_{\text{def}} \textsc{the named-melanie}$ passes to its scope argument the unique discourse referent entailed to be named "Melanie", similarly to *Lance*, above. The pronouns *herself* and *it* receive the same treatment as in SU, selecting the unique discourse referent consistent with being female and nonhuman, respectively. I assume that some mechanism intervenes to enforce binding constraints for pronouns, but do not go into an explicit modeling of binding here.

The lexical entry for the common noun *car* closely resembles the one for *doper* in (21). In the lexical entry below, CAR is the dynamic property of being a car:

$$\vdash \text{car}\ ;\ \text{N}\ ;\ \textsc{car} \tag{63}$$

As for the verbs, the pheno must spell out the surface locations of the eventual arguments.

(64) $\vdash \lambda_{stu}.u \cdot \text{bought} \cdot t \cdot s \,;\, \text{NP} \multimap \text{NP} \multimap \text{NP} \multimap \text{S} \,;\, \textsc{buy}$

(65) $\vdash \lambda_{st}.t \cdot \text{drives} \cdot s \,;\, \text{NP} \multimap \text{NP} \multimap \text{S} \,;\, \textsc{drive}$

Here, BUY and DRIVE are respectfully the ternary and binary properties corresponding to buying and driving.

The derivation of a sign representing (59) starts by hypothesizing traces for the arguments to *bought*. Starting with the objects, we derive the following:

(66)
$$\dfrac{\dfrac{(64) \qquad s \,;\, \text{NP} \,;\, k \vdash s \,;\, \text{NP} \,;\, k}{s \,;\, \text{NP} \,;\, k \vdash \lambda_{tu}.u \cdot \text{bought} \cdot t \cdot s \,;\, \text{NP} \multimap \text{NP} \multimap \text{S} \,;\, \textsc{buy}\, k} \qquad t \,;\, \text{NP} \,;\, m \vdash t \,;\, \text{NP} \,;\, m}{s \,;\, \text{NP} \,;\, k, t \,;\, \text{NP} \,;\, m \vdash \lambda_u.u \cdot \text{bought} \cdot t \cdot s \,;\, \text{NP} \multimap \text{S} \,;\, \textsc{buy}\, k\, m}$$

Then a trace for the subject is provided and the object trace withdrawn.

(67)
$$\dfrac{\dfrac{(66) \qquad u \,;\, \text{NP} \,;\, n \vdash u \,;\, \text{NP} \,;\, n}{s \,;\, \text{NP} \,;\, k, t \,;\, \text{NP} \,;\, m, u \,;\, \text{NP} \,;\, n \vdash u \cdot \text{bought} \cdot t \cdot s \,;\, \text{S} \,;\, \textsc{buy}\, k\, m\, n}}{t \,;\, \text{NP} \,;\, m, u \,;\, \text{NP} \,;\, n \vdash \lambda_s.u \cdot \text{bought} \cdot t \cdot s \,;\, \text{NP} \multimap \text{S} \,;\, \lambda_k.\textsc{buy}\, k\, m\, n}$$

In preparation for integrating *a car* with *bought*, the GQ is derived.

(68)
$$\dfrac{(20) \qquad (63)}{\vdash \lambda_f.f\ (\text{a} \cdot \text{car}) \,;\, \text{QP} \,;\, \textsc{a car}}$$

Next *a car* takes scope, and the indirect object trace is withdrawn.

(69)
$$\dfrac{\dfrac{(68) \qquad (67)}{t \,;\, \text{NP} \,;\, m, u \,;\, \text{NP} \,;\, n \vdash u \cdot \text{bought} \cdot t \cdot \text{a} \cdot \text{car} \,;\, \text{S} \,;\, (\textsc{a car})_k.\textsc{buy}\, k\, m\, n}}{u \,;\, \text{NP} \,;\, n \vdash \lambda_t.u \cdot \text{bought} \cdot t \cdot \text{a} \cdot \text{car} \,;\, \text{NP} \multimap \text{S} \,;\, \lambda_m.(\textsc{a car})_k.\textsc{buy}\, k\, m\, n}$$

Then *herself* takes scope, and the subject trace is withdrawn.

(70)
$$\dfrac{\dfrac{(61) \qquad (69)}{u \,;\, \text{NP} \,;\, n \vdash u \cdot \text{bought} \cdot \text{herself} \cdot \text{a} \cdot \text{car} \,;\, \text{S} \,;\, \textsc{herself}_m.(\textsc{a car})_k.\textsc{buy}\, k\, m\, n}}{\vdash \lambda_u.u \cdot \text{bought} \cdot \text{herself} \cdot \text{a} \cdot \text{car} \,;\, \text{NP} \multimap \text{S} \,;\, \lambda_n.\textsc{herself}_m.(\textsc{a car})_k.\textsc{buy}\, k\, m\, n}$$

And then the nonrestrictive relativizer *who* is applied to (70) to derive the NRRC in (59).

(71)
$$\dfrac{(35) \qquad (70)}{\vdash \text{who} \cdot \mathbf{e} \cdot \text{bought} \cdot \text{herself} \cdot \text{a} \cdot \text{car} \,;\, \text{Pred} \,;\, (\textsc{pred}\ \lambda_n.\textsc{herself}_m.(\textsc{a car})_k.\textsc{buy}\, k\, m\, n)}$$

With the NRRC sign completed, we turn to deriving the verb phrase *drives it*. As for *bought herself a car*, the process starts by providing traces for the verb's arguments, then withdrawing the object trace.

(72)
$$\dfrac{\dfrac{\dfrac{(65) \qquad s \,;\, \text{NP} \,;\, k \vdash s \,;\, \text{NP} \,;\, k}{s \,;\, \text{NP} \,;\, k \vdash \lambda_t.t \cdot \text{drives} \cdot s \,;\, \text{NP} \multimap \text{S} \,;\, \textsc{drive}\, k} \qquad u \,;\, \text{NP} \,;\, n \vdash u \,;\, \text{NP} \,;\, n}{s \,;\, \text{NP} \,;\, k, u \,;\, \text{NP} \,;\, n \vdash u \cdot \text{drives} \cdot s \,;\, \text{S} \,;\, \textsc{drive}\, k\, n}}{u \,;\, \text{NP} \,;\, n \vdash \lambda_s.u \cdot \text{drives} \cdot s \,;\, \text{NP} \multimap \text{S} \,;\, \lambda_k.\textsc{drive}\, k\, n}$$

The next step is for the pronoun *it* to take scope, and then withdraw the subject trace.

(73)
$$\dfrac{\dfrac{(62) \qquad (72)}{u \,;\, \text{NP} \,;\, n \vdash u \cdot \text{drives} \cdot \text{it} \,;\, \text{S} \,;\, \textsc{it}_k.\textsc{drive}\, k\, n}}{\vdash \lambda_u.u \cdot \text{drives} \cdot \text{it} \,;\, \text{NP} \multimap \text{S} \,;\, \lambda_n.\textsc{it}_k.\textsc{drive}\, k\, n}$$

Now all the ingredients required by the comma intonation are in place, so we can start deriving a sign representing (59) in its entirety. Starting with *Melanie* and the supplement, we derive:

$$(74)\quad \dfrac{\dfrac{(17) \qquad\qquad (60)}{\vdash \lambda_{sg}.g\,(\text{melanie} \cdot (\mathsf{comma}\ s))\ ;\ \text{Pred} \multimap \text{QP}\ ;\ \textsc{comma melanie}} \qquad (71)}{\begin{array}{c}\vdash \lambda_g.g\,(\text{melanie} \cdot \mathsf{comma}\,(\text{who} \cdot \mathbf{e} \cdot \text{bought} \cdot \text{herself} \cdot \text{a} \cdot \text{car}))\ ;\ \text{QP}\ ;\\ \textsc{comma melanie}\,(\textsc{pred}\ \lambda_n.\textsc{herself}_m.(\textsc{a car})_k.\textsc{buy}\ k\ m\ n)\end{array}}$$

Finally, the comma intonation takes the verb phrase derived in (73) as its last argument.

$$(75)\quad \dfrac{(74) \qquad\qquad (73)}{\begin{array}{c}\vdash \text{melanie} \cdot \mathsf{comma}\,(\text{who} \cdot \mathbf{e} \cdot \text{bought} \cdot \text{herself} \cdot \text{a} \cdot \text{car}) \cdot \text{drives} \cdot \text{it}\ ;\ \text{S}\ ;\\ \textsc{comma melanie}\,(\textsc{pred}\ \lambda_n.\textsc{herself}_m.(\textsc{a car})_k.\textsc{buy}\ k\ m\ n)\ \lambda_n.\textsc{it}_k.\textsc{drive}\ k\ n\end{array}}$$

The semantics of this sign shows how the anaphoric links are enabled: *herself* is interpreted in a context that has *Melanie* as an available antecedent, and similarly, the context passed to *it* contains *a car*. As for the surface form, with the empty string $\mathbf{e}$ axiomatized as an identity for inhabitants of the type of strings, the derived pheno term is equivalent to $\text{melanie} \cdot \mathsf{comma}\,(\text{who} \cdot \text{bought} \cdot \text{herself} \cdot \text{a} \cdot \text{car}) \cdot \text{drives} \cdot \text{it}$, as desired.

Since supplements are not special with respect to the incremental interpretation in this framework, anaphora between two supplements is also very straightforward. The following example gives a nice illustration:

(76) Melanie$_i$, who bought herself$_i$ a car$_j$, met some cyclist, its$_j$ former owner.

All of the elements required for deriving a sign representing (76) are already in place except those in the last supplement. The needed lexical extensions are given below.

$$(77)\quad \vdash \lambda_{sf}.f\,(\text{its} \cdot s)\ ;\ \text{N} \multimap \text{QP}\ ;\ \textsc{its}$$

$$(78)\quad \vdash \text{former} \cdot \text{owner}\ ;\ \text{N}\ ;\ \textsc{owner}$$

The entry in (62) corresponds to the possessive determiner *its*, and (78) syncategorematically defines *former owner* to avoid the complexities associated with *former*.

This proof starts by deriving a version of the verb *met* with a hypothesized object.

(79)

$$\dfrac{\dfrac{(51)\quad s\ ;\ \text{NP}\ ;\ m \vdash s\ ;\ \text{NP}\ ;\ m}{s\ ;\ \text{NP}\ ;\ m \vdash \lambda_t.t \cdot \text{met} \cdot s\ ;\ \text{NP} \multimap \text{S}\ ;\ \textsc{met}\ m} \qquad t\ ;\ \text{NP}\ ;\ n \vdash t\ ;\ \text{NP}\ ;\ n}{\dfrac{s\ ;\ \text{NP}\ ;\ m, t\ ;\ \text{NP}\ ;\ n \vdash t \cdot \text{met} \cdot s\ ;\ \text{S}\ ;\ \textsc{met}\ m\ n}{t\ ;\ \text{NP}\ ;\ n \vdash \lambda_s.t \cdot \text{met} \cdot s\ ;\ \text{NP} \multimap \text{S}\ ;\ \lambda_m.\textsc{met}\ m\ n}}$$

Then the QP *its former owner* is derived and predicativized.

$$(80)\quad \dfrac{(19)\quad \dfrac{(77) \qquad\qquad (78)}{\vdash \lambda_f.f\,(\text{its} \cdot \text{former} \cdot \text{owner})\ ;\ \text{QP}\ ;\ \textsc{its owner}}}{\vdash \text{its} \cdot \text{former} \cdot \text{owner}\ ;\ \text{Pred}\ ;\ (\textsc{pred its owner})}$$

The utterance-final comma intonation then takes *some cyclist* as its argument.

(81)

$$\dfrac{(54) \qquad\qquad (14)}{\vdash \lambda_{gs}.g\,(\text{some} \cdot \text{cyclist} \cdot (\mathsf{comma}\ s))\ ;\ (\text{NP} \multimap \text{S}) \multimap \text{Pred} \multimap \text{S}\ ;\ \textsc{comma}\,(\textsc{a cyclist})}$$

Next, the newly derived verb phrase and predicative are integrated, and then the subject trace is withdrawn.

$$\dfrac{\dfrac{(81) \qquad (79)}{\begin{array}{l} t\,;\mathrm{NP}\,;n \vdash \lambda_s.t \cdot \mathrm{met} \cdot \mathrm{some} \cdot \mathrm{cyclist} \cdot (\mathsf{comma}\; s)\,;\mathrm{Pred} \multimap \mathrm{S}\,; \\ \quad \textsc{comma}\;(\textsc{a cyclist})\;\lambda_m.\textsc{met}\;m\;n \end{array}}\ (80)}{\begin{array}{l} t\,;\mathrm{NP}\,;n \vdash t \cdot \mathrm{met} \cdot \mathrm{some} \cdot \mathrm{cyclist} \cdot \mathsf{comma}\;(\mathrm{its} \cdot \mathrm{former} \cdot \mathrm{owner})\,;\mathrm{S}\,; \\ \quad \textsc{comma}\;(\textsc{a cyclist})\;(\lambda_m(\textsc{met}\;m\;n))\;(\textsc{pred its owner}) \end{array}} \tag{82}$$

$$\vdash \lambda_t.t \cdot \mathrm{met} \cdot \mathrm{some} \cdot \mathrm{cyclist} \cdot \mathsf{comma}\;(\mathrm{its} \cdot \mathrm{former} \cdot \mathrm{owner})\,;\mathrm{NP} \multimap \mathrm{S}\,;$$
$$\lambda_n.\textsc{comma}\;(\textsc{a cyclist})\;(\lambda_m(\textsc{met}\;m\;n))\;(\textsc{pred its owner})$$

With the second, utterance-final supplement derived, the rest of the derivation proceeds by providing the utterance-medial comma intonation (17) with its arguments.

$$\dfrac{\dfrac{(17) \qquad (60)}{\vdash \lambda_{sg}.g\;(\mathrm{melanie} \cdot (\mathsf{comma}\; s))\,;\mathrm{Pred} \multimap \mathrm{QP}\,;\textsc{comma melanie}}\ (71)}{\begin{array}{l} \vdash \lambda_g.g\;(\mathrm{melanie} \cdot \mathsf{comma}\;(\mathrm{who} \cdot \mathbf{e} \cdot \mathrm{bought} \cdot \mathrm{herself} \cdot \mathrm{a} \cdot \mathrm{car}))\,;\mathrm{QP}\,; \\ \quad \textsc{comma melanie}\;(\textsc{pred}\;\lambda_n.\textsc{herself}_m.(\textsc{a car})_k.\textsc{buy}\;k\;m\;n) \end{array}} \tag{83}$$

As its final argument, the matrix comma intonation takes the sign in (82).

$$\dfrac{(83) \qquad (82)}{\begin{array}{l} \vdash \mathrm{melanie} \cdot \mathsf{comma}\;(\mathrm{who} \cdot \mathbf{e} \cdot \mathrm{bought} \cdot \mathrm{herself} \cdot \mathrm{a} \cdot \mathrm{car}) \cdot \mathrm{met} \cdot \mathrm{some} \cdot \mathrm{cyclist} \cdot \\ \quad \mathsf{comma}\;(\mathrm{its} \cdot \mathrm{former} \cdot \mathrm{owner})\,;\mathrm{S}\,; \\ \quad \textsc{comma melanie}\;(\textsc{pred}\;\lambda_n.\textsc{herself}_m.(\textsc{a car})_k.\textsc{buy}\;k\;m\;n) \\ \quad \lambda_n.\textsc{comma}\;(\textsc{a cyclist})\;(\lambda_m(\textsc{met}\;m\;n))\;(\textsc{pred its owner}) \end{array}} \tag{84}$$

Since the underlying semantic framework gives an incremental, dynamic interpretation, not only is *Melanie* accessible as the antecedent to *herself*, as before, but also the car introduced is accessible from *its*. And so, in this account, anaphora among supplements and between supplements and other content is treated exactly as all other instances of anaphora are.

3.6 Stacking

This supplement syntax also extends to the phenomenon of supplement *stacking* (Potts, 2005, 100), in which multiple supplements are attached to the same anchor, such as

(85) Lance, a cyclist, a doper, won the Tour de France.

Since the comma intonation takes a QP and a predicativized QP to another QP, chaining together supplements is fairly straightforward. Starting with the first supplement, the QP *a cyclist* is first derived.

$$\dfrac{(19)\ \dfrac{(20) \qquad (12)}{\vdash \lambda_f.f\;(\mathrm{a} \cdot \mathrm{cyclist})\,;\mathrm{QP}\,;\textsc{a cyclist}}}{\vdash \mathrm{a} \cdot \mathrm{cyclist}\,;\mathrm{Pred}\,;\textsc{pred a cyclist}} \tag{86}$$

Applying the (medial) comma intonation to *Lance* and the first supplement yields a QP.

$$\dfrac{\dfrac{(17) \qquad (41)}{\vdash \lambda_{sg}.g\;(\mathrm{lance} \cdot (\mathsf{comma}\; s))\,;\mathrm{Pred} \multimap \mathrm{QP}\,;\textsc{comma lance}}\ (87)}{\vdash \lambda_g.g\;(\mathrm{lance} \cdot \mathsf{comma}\;(\mathrm{a} \cdot \mathrm{cyclist}))\,;\mathrm{QP}\,;\textsc{comma lance}\;(\textsc{pred a cyclist})} \tag{87}$$

A second instance of the comma intonation then takes this QP as its argument.

$$\frac{(17) \qquad (87)}{\begin{array}{c}\vdash \lambda_{sg}.g\,(\text{lance} \cdot \mathsf{comma}\,(\text{a} \cdot \text{cyclist}) \cdot (\mathsf{comma}\,s))\,;\,\text{Pred} \multimap \text{QP}\,; \\ \textsc{comma}\,(\textsc{comma lance}\,(\textsc{pred a cyclist}))\end{array}} \tag{88}$$

Then the second supplement is taken as the argument to this second comma intonation.

$$\frac{(88) \qquad (23)}{\begin{array}{c}\vdash \lambda_{g}.g\,(\text{lance} \cdot \mathsf{comma}\,(\text{a} \cdot \text{cyclist}) \cdot \mathsf{comma}\,(\text{a} \cdot \text{doper}))\,;\,\text{QP}\,; \\ \textsc{comma}\,(\textsc{comma lance}\,(\textsc{pred a cyclist}))\,(\textsc{pred a doper})\end{array}} \tag{89}$$

Lastly, this new QP incorporating both supplements is applied to the verb phrase.

(90)
$$\frac{(89) \qquad (13)}{\begin{array}{c}\vdash \text{lance} \cdot \mathsf{comma}\,(\text{a} \cdot \text{cyclist}) \cdot \mathsf{comma}\,(\text{a} \cdot \text{doper}) \cdot \text{won} \cdot \text{the} \cdot \text{tour} \cdot \text{de} \cdot \text{france}\,;\,\text{S}\,; \\ \textsc{comma}\,(\textsc{comma lance}\,(\textsc{pred a cyclist}))\,(\textsc{pred a doper})\,\textsc{win-tdf}\end{array}}$$

Expanding the semantics of this sign shows how the content of both supplements is integrated.

$$\begin{aligned}&\textsc{comma}\,(\textsc{comma lance}\,(\textsc{pred a cyclist}))\,(\textsc{pred a doper})\,\textsc{win-tdf} \\ &= (\textsc{comma lance}\,(\textsc{pred a cyclist})\,(\textsc{pred a doper}))\,\textsc{and} \\ &\quad (\textsc{the}\,(\textsc{pred a doper})\,\textsc{win-tdf}) \\ &= (\textsc{lance}\,(\textsc{pred a cyclist}))\,\textsc{and}\,(\textsc{the}\,(\textsc{pred a cyclist})\,(\textsc{pred a doper}))\,\textsc{and} \\ &\quad (\textsc{the}\,(\textsc{pred a doper})\,\textsc{win-tdf})\end{aligned}$$

In this semantics for (85), the derived meaning could be paraphrased by *Lance is a cyclist. The one who's a cyclist is also a doper. That doper won the Tour de France.*

For the case of utterance-final supplements like

(91) Some cyclist met Lance, a cyclist, a doper,

a model of stacking is more complicated because the tecto type of the utterance-final comma intonation is less amenable to composition. After the comma intonation is provided with its QP argument *Lance*, it takes a trace as its verb phrase argument.

$$\frac{(55) \qquad g\,;\,\text{NP} \multimap \text{S}\,;\,D \vdash g\,;\,\text{NP} \multimap \text{S}\,;\,D}{g\,;\,\text{NP} \multimap \text{S}\,;\,D \vdash \lambda_s.g\,(\text{lance} \cdot (\mathsf{comma}\,s))\,;\,\text{Pred} \multimap \text{S}\,;\,\textsc{comma lance}\,D} \tag{92}$$

The first supplement is then integrated, and then the trace is withdrawn to make way for the second comma intonation.

$$\frac{\dfrac{(92) \qquad (86)}{\begin{array}{c}g\,;\,\text{NP} \multimap \text{S}\,;\,D \vdash g\,(\text{lance} \cdot \mathsf{comma}\,(\text{a} \cdot \text{cyclist}))\,;\,\text{S}\,; \\ \textsc{comma lance}\,D\,(\textsc{pred a cyclist})\end{array}}}{\begin{array}{c}\vdash \lambda_g.g\,(\text{lance} \cdot \mathsf{comma}\,(\text{a} \cdot \text{cyclist}))\,;\,\text{QP}\,; \\ \lambda_D.\textsc{comma lance}\,D\,(\textsc{pred a cyclist})\end{array}} \tag{93}$$

Now the comma intonation takes the QP derived in (93) as its first argument.

$$\frac{(54) \qquad (93)}{\begin{array}{c}\vdash \lambda_{gs}.g\,(\text{lance} \cdot \mathsf{comma}\,(\text{a} \cdot \text{cyclist}) \cdot (\mathsf{comma}\,s))\,;\,(\text{NP} \multimap \text{S}) \multimap \text{Pred} \multimap \text{S}\,; \\ \textsc{comma}\,\lambda_D.\textsc{comma lance}\,D\,(\textsc{pred a cyclist})\end{array}} \tag{94}$$

Next, the verb phrase containing an object gap is integrated.
(95)

$$\frac{(94) \qquad (56)}{\begin{array}{l}\vdash \lambda_s.\text{some} \cdot \text{cyclist} \cdot \text{met} \cdot \text{lance} \cdot \mathsf{comma}\,(\text{a} \cdot \text{cyclist}) \cdot (\mathsf{comma}\; s)\,;\, \text{Pred} \multimap \text{S}\,; \\ \quad \textsc{comma}\;(\lambda_D(\textsc{comma lance}\; D\;(\textsc{pred a cyclist})))\;\lambda_m.(\textsc{a cyclist})_n.\textsc{met}\; m\; n\end{array}}$$

In the last proof step, the second supplement is provided as the final argument to the comma intonation.
(96)

$$\frac{(95) \qquad (23)}{\begin{array}{l}\vdash \text{some} \cdot \text{cyclist} \cdot \text{met} \cdot \text{lance} \cdot \mathsf{comma}\,(\text{a} \cdot \text{cyclist}) \cdot \mathsf{comma}\,(\text{a} \cdot \text{doper})\,;\, \text{S}\,; \\ \quad \textsc{comma}\;(\lambda_D(\textsc{comma lance}\; D\;(\textsc{pred a cyclist})))\;\lambda_m.(\textsc{a cyclist})_n.\textsc{met}\; m\; n \\ \qquad (\textsc{pred a doper})\end{array}}$$

Like the medial stacking case above, the embedded comma intonations make the semantics derived in (96) somewhat complex. It reduces as follows:

$$\begin{array}{l}\textsc{comma}\;(\lambda_D(\textsc{comma lance}\; D\;(\textsc{pred a cyclist})))\;\lambda_m.(\textsc{a cyclist})_n.\textsc{met}\; m\; n \\ \quad (\textsc{pred a doper}) \\ = (\textsc{comma lance}\;(\lambda_m(\textsc{a cyclist})_n.\textsc{met}\; m\; n)\;(\textsc{pred a cyclist}))\;\textsc{and} \\ \quad \textsc{the}\;(\lambda_m(\textsc{a cyclist})_n.\textsc{met}\; m\; n)\;(\textsc{pred a doper}) \\ = (\textsc{lance}_m(\textsc{a cyclist})_n.\textsc{met}\; m\; n)\;\textsc{and} \\ \quad (\textsc{the}\;(\lambda_m(\textsc{a cyclist})_n.\textsc{met}\; m\; n)\;(\textsc{pred a cyclist}))\;\textsc{and} \\ \quad \textsc{the}\;(\lambda_m(\textsc{a cyclist})_n.\textsc{met}\; m\; n)\;(\textsc{pred a doper})\end{array}$$

Roughly, this meaning for (91) is equivalent to *Lance was met by some cyclist. The one that a cyclist met is a cyclist, and he's also a doper.*

4 Comparison with other accounts

At the level of semantics, this account breaks with nearly all others in treating supplements as ordinary at-issue denotations. Rather than construing supplements as having a special property requiring them to project, their projection is instead directly linked to the scope of their anchors, which is itself influenced by independent processes. The impact of this design choice on the syntax is that supplements can be modeled syntactically as attaching themselves to their anchors. As detailed above, the implication for the two-component syntax of Dynamic Categorial Grammar is that supplement syntax can be captured straightforwardly via two lexical entries, one for utterance-medial supplements and the other for utterance-final ones. I argue that in both its parsimony and empirical coverage, this account compares favorably with previous attempts to model supplements.

The tradition of movement-based analysis exemplified by McCawley (1998), del Gobbo (2007), and Schlenker (2010, ms) share the common feature that they treat supplements as extraposed free-standing sentences. However, as Potts (2005, chapter 6) points out, this has the unsavory consequence of requiring a non-standard redefinition of the syntactic dominance relation. An illustrative example is

(97) Fred, who you met at a party, is a lawyer. (McCawley, 1998, 449, (20))

McCawley treats nonrestrictive relatives as sentence-level adjuncts that are then moved to a position immediately following their anchor. Figure 2 shows the pre-transformation syntax

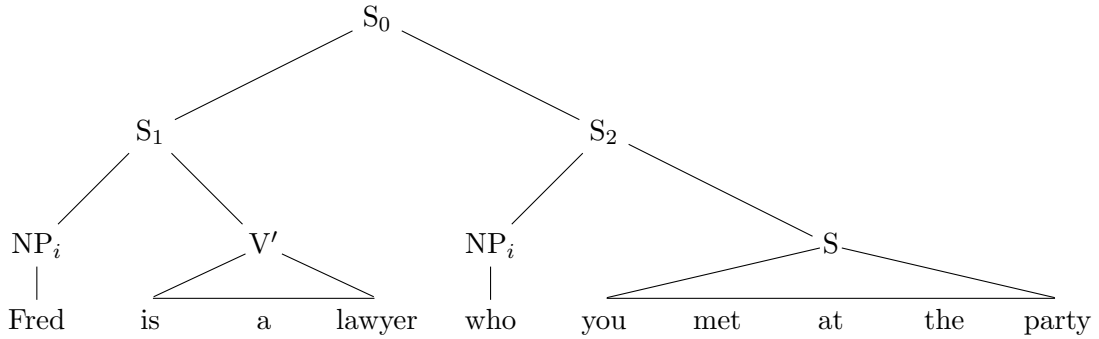


Figure 2: McCawley's (1998) syntax for the pre-transformation variant of (97).

for (97) in McCawley's (1998) account. The syntax in Figure 2 highlights an unfavorable aspect of the transformational analyses pointed out by Schlenker (ms, 20): the anchor *Fred* and the relativizer *who* must be co-indexed, and so a constituent's interpretation depends in part on a neighboring constituent's interpretation, implying that the resulting semantics departs from strong compositionality.

Another complicated aspect of these analyses can been seen in the syntax for (97) that McCawley's account posits as the result of the transformation (Figure 3). The configuration

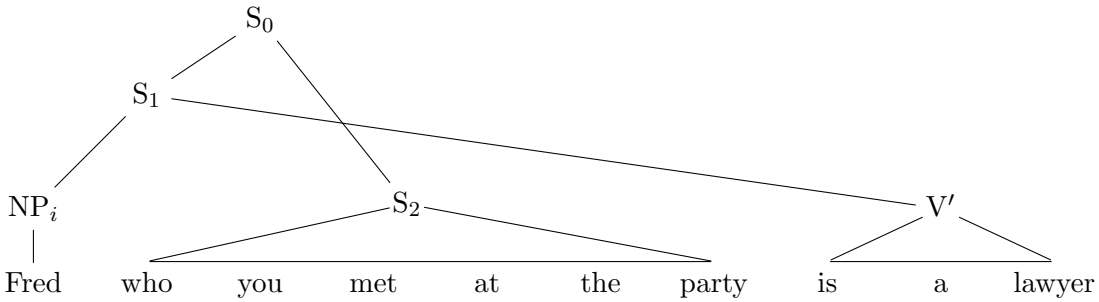


Figure 3: McCawley's (1998) syntax for (97), after transforming it from the syntax in Figure 2.

of the arcs show how McCawley's account needs a non-standard notion of dominance. To derive the correct surface word order, McCawley's transformation generates a tree that does not adhere to the condition of *nontangling*, which stipulates that all children of a node that precedes another node must also precede that other node's children.

Potts (2005, chapter 6) also highlights multiple additional problems with McCawley's approach, casting doubt on the idea that the underlying form in Figure 2 represents a structure that mirrors the interpretation of supplements. Note that, as Figure 2 shows, McCawley's *syntax* of supplements models them as separate but conjoined sentences, whereas the account here models a supplement's syntax as attaching to its anchor's and its *semantics* as generating a separate update. This account's syntax is thus much simpler than the transformational process described by Figures 2 and 3, with lambda terms in the pheno configured to simply attach the supplement's surface form to its anchor's (cf. (17) and (54)). Also, although it does use the anaphoric determiner THE in the meaning of the comma intonation, this account does not require the use of *E-type pronouns*, as del Gobbo's (2007) transformational account does.

Though, strictly speaking, it does not give an account of supplement syntax, Potts's (2005) account of supplements is not without its drawbacks. In addition to requiring that supplement content always inhabit a separate meaning dimension from non-supplement content, which I argue in §2.1 is empirically incorrect, Potts's semantics is already quite complex. For Potts, a special rule applies when ordinary content combines with supplement content, generating a pair of denotations. The semantic contribution of this pair of denotations must later be harvested via another special rule of "parsetree interpretation" that traverses the derivation to collect any supplement content and add it back into the meaning of the entire utterance. Since Potts's account is only semantic, it is difficult to guess how it would be extended to generate the corresponding surface forms. Clearly, though, even more specialized machinery would be needed.

Another approach to keeping supplement and non-supplement content separate is to use the continuation passing technique, as do Kubota and Uegaki (2009), extending Barker and Shan's (2008) continuation semantics. For Kubota and Uegaki, the semantics is set up so that the types require supplements to always take widest scope, i.e., to project. Like most other accounts, the predictions made by Kubota and Uegaki's are too strict, since they do not allow supplements to scope narrow with respect to operators. The added machinery is also quite complicated, requiring very complex lambda terms as well as specialized (though general) rules for dealing with continuations. A similar approach in this same vein is the use of monads to keep supplement content separate from other content (Giorgolo and Asudeh, 2012), but this approach is equally complicated, and still not fine-grained enough to allow supplements to interact with scope-taking operators.

AnderBois et al. (2010, 2015), Bekki and McCready (2014), and Koev (2014) all give accounts of supplements that allows them to interact anaphorically with other content, while keeping their content separate from main-clause content. However, none of these accounts allows supplements to participate in scope interactions. As such, some extensions would be required to each of these accounts in order to model any of the data in (1)–(5), which show that supplements have the potential to scope narrow relative to operators.

By comparison, the account of supplement syntax given here is very simple, requiring only a pair of lexical entries to capture all of the data in §2.1, including cases where supplements take scope and cases where they are deniable. It would be possible to argue that the addition of a second syntactic component is itself a quite complex move, even though the syntax is general purpose. However, both of the syntactic components are well-motivated, with one capturing the underlying combinatorics and the other dedicated to generating the proper surface forms. One might also argue that the heavy use of generalized quantification in this account is not too dissimilar from the continuation passing technique, since generalized quantification can be seen as an instance of continuation passing (Barker and Shan, 2014). I leave open the question of whether an analogous account expressed in Barker and Shan's framework would be more or less complicated.

5 Conclusion

This paper straightforwardly extends SU's semantics for supplements to a full syntax-semantics interface. In comparison to other accounts of supplement syntax, only a single meaning dimension, and not much extra machinery, is required to get this syntax-semantics interface—in fact, only two new lexical entries for the comma intonation are needed specifically to model supplements. Much of the reduced complexity comes from the fact that SU's account departs from the idea that supplements always project, never interacting with other content. Treating supplements as essentially extra property denotations added

into the restrictor of a generalized quantifier allows the grammar to harness independently motivated machinery to handle them.

In place of the complex transformations, special-purpose rules, continuations and monads posited by other accounts, this account instead calls upon the two-component syntax developed by Martin and Pollard (2014), which is used independently to model all other syntactic aspects of the grammar. The resulting formalism handles a wide range of phenomena, including supplements in both medial and final position, stacked supplements, and anaphora between supplements and other content. In addition to its reduced complexity, this account also compares favorably with others because it covers more of the facts, allowing supplements to interact with other content at the level of scope.

References

Amaral, Patricia, Craige Roberts, and E. Allyn Smith. 2007. Review of *The Logic of Conventional Implicatures* by Chris Potts. *Linguistics and Philosophy* 30(6):707–749.

AnderBois, Scott, Adrian Brasoveanu, and Robert Henderson. 2010. Crossing the appositive/at-issue meaning boundary. In *20th Conference on Semantics and Linguistic Theory (SALT)*.

AnderBois, Scott, Adrian Brasoveanu, and Robert Henderson. 2015. At-issue proposals and appositive impositions in discourse. *Journal of Semantics* 32(1):93–138.

Barker, Chris, Rafaella Bernardi, and Chung-chieh Shan. 2010. Principles of interdimensional meaning interaction. In *20th Conference on Semantics and Linguistic Theory (SALT)*.

Barker, Chris and Chung-chieh Shan. 2008. Donkey anaphora is in-scope binding. *Semantics and Pragmatics* 1(1):1–42.

Barker, Chris and Chung-chieh Shan. 2014. *Continuations and Natural Language*. No. 53 in Oxford Studies in Theoretical Linguistics. New York, NY and Oxford: Oxford University Press.

Barwise, Jon and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 4(2):159–220.

Beaver, David I. 2001. *Presupposition and Assertion in Dynamic Semantics*. Stanford, CA: CSLI Publications.

Bekki, Daisuke and Eric McCready. 2014. CI via DTS. In *11th International Workshop on Logic and Engineering of Natural Language Semantics (LENLS)*, 110–123.

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

de Groote, Philippe. 2006. Towards a Montagovian account of dynamics. In *16th Conference on Semantics and Linguistic Theory (SALT)*.

del Gobbo, Francesca. 2007. On the syntax and semantics of appositive relative clauses. In N. Dehé and Y. Kavalova, eds., *Parentheticals*, 173–201. Amsterdam & Philadelphia, PA: John Benjamins.

Ginzburg, Jonathan. 2012. *The Interactive Stance*. New York, NY and Oxford: Oxford University Press.

Giorgolo, Gianluca and Ash Asudeh. 2012. $\langle M, \eta, \star \rangle$ monads for conventional implicatures. In A. A. Guevara, A. Chernilovskaya, and R. Nouwen, eds., *Sinn und Bedeutung 16*, vol. 1. Cambridge, MA: MIT Working Papers in Linguistics.

Harris, Jesse A. and Christopher Potts. 2009. Perspective-shifting with appositives and expressives. *Linguistics and Philosophy* 32(6):532–552.

Huddleston, Rodney and Geoffrey K. Pullum. 2002. *The Cambridge grammar of the English language*. Cambridge: Cambridge University Press.

Keenan, Edward L. and Jonathan Stavi. 1986. A semantic characterization of natural language determiners. *Linguistics and Philosophy* 9(3):253–326.

Koev, Todor. 2012. On the information status of appositive relative clauses. In M. Aloni, V. Kimmelman, F. Roelofsen, G. W. Sassoon, K. Schulz, and M. Westera, eds., *Logic, Language and Meaning*, No. 7218 in Lecture Notes in Computer Science, 401–410. Berlin and Heidelberg: Springer.

Koev, Todor. 2014. Two puzzles about appositives: Projection and perspective shift. In U. Exteberria, A. Fălăuş, A. Irurtzun, and B. Leferman, eds., *Sinn und Bedeutung 18*. University of the Basque Country.

Kubota, Yusuke and Wataru Uegaki. 2009. Continuation-based semantics for conventional implicatures: The case of Japanese benefactives. In *19th Conference on Semantics and Linguistic Theory (SALT)*.

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

Martin, Scott. in press. Supplemental update. *Semantics and Pragmatics* (`http://semanticsarchive.net/Archive/TMzYTZhY/suppdate.pdf`).

Martin, Scott and Carl Pollard. 2012a. A higher-order theory of presupposition. *Studia Logica* 100(4):727–751.

Martin, Scott and Carl Pollard. 2012b. Hyperintensional dynamic semantics: Analyzing definiteness with enriched contexts. In P. de Groote and M.-J. Nederhof, eds., *Formal Grammar*, No. 7395 in Lecture Notes in Computer Science, 114–129. Dordrecht: Springer.

Martin, Scott and Carl Pollard. 2014. A dynamic categorial grammar. In G. Morrill, R. Muskens, R. Osswald, and F. Richter, eds., *19th Conference on Formal Grammar*, No. 8612 in Lecture Notes in Computer Science, 138–154. Dordrecht: Springer.

McCawley, James D. 1998. *The syntactic phenomena of English*. Chicago, IL: University of Chicago Press, 2nd edn.

Mihaliček, Vedrana. 2012. *Serbo-Croatian Word Order: A Logical Approach*. Ph.D. thesis, Ohio State University, Columbus, OH.

Murray, Sarah E. 2014. Varieties of update. *Semantics and Pragmatics* 7(2):1–53.

Muskens, Reinhard. 1996. Combining Montague semantics and discourse representation theory. *Linguistics and Philosophy* 19(2):143–186.

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

Nouwen, Rick. 2007. On appositives and dynamic binding. *Research on Language and Computation* 5(1):87–102.

Nouwen, Rick. 2014. A note on the projection of appositives. In E. McCready, K. Yabushita, and K. Yoshimoto, eds., *Formal Approaches to Semantics and Pragmatics: Japanese and Beyond*, No. 95 in Studies in Linguistics and Philosophy. Dordrecht: Springer.

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

Potts, Christopher. 2005. *The Logic of Conventional Implicatures*. No. 7 in Oxford Studies in Theoretical Linguistics. New York, NY and Oxford: Oxford University Press.

Roberts, Craige. 1989. Modal subordination and pronominal anaphora in discourse. *Linguistics and Philosophy* 12(6):683–721.

Roberts, Craige. 2005. Pronouns as definites. In M. Reimer and A. Bezuidenhout, eds., *Descriptions and Beyond*, 503–543. New York, NY and Oxford: Oxford University Press.

Roberts, Craige. 2012a. Information structure: Afterword. *Semantics and Pragmatics* 5(7):1–19.

Roberts, Craige. 2012b. Information structure in discourse: Towards an integrated formal theory of pragmatics. *Semantics and Pragmatics* 5(6):1–69. Accompanying afterword in Roberts 2012a.

Schlenker, Philippe. 2010. Supplements within a unidimensional semantics I: Scope. In M. Aloni, H. Bastiaanse, T. de Jager, and K. Schulz, eds., *Logic, Language and Meaning*, No. 6042 in Lecture Notes in Computer Science, 77–83. Dordrecht: Springer.

Schlenker, Philippe. ms. Supplements without bidimensionalism. Unpublished manuscript, Institut Jean-Nicod and New York University, February, 2013. (`http://www.semanticsarchive.net/Archive/jgwMjNmM/Supplements_without_Bidimensionalism.pdf`).

Tonhauser, Judith, David Beaver, Craige Roberts, and Mandy Simons. 2013. Towards a taxonomy of projective content. *Language* 89(1):66–109.

Worth, Chris. 2014. The phenogrammar of coordination. In *EACL Workshop on Type Theory and Natural Language Semantics*. Association for Computational Linguistics.

Combining Grammar Logics

Michael Moortgat
Utrecht Institute of Linguistics OTS

Abstract

Current categorial grammar frameworks are generally obtained by putting together formulas with a distinct interpretation in one encompassing logic: multimodal and hybrid typelogical grammar, displacement calculus, bilinear logic or Lambek-Grishin calculus are representative examples. Girard's "unity of logic" (UL) framework sets the standard for this type of combination. The UL formulas denote stable truths or non-reusable finite resources. The distinction splits conjunction in an additive and a multiplicative operation. The communication between these two is controlled by a new logical constant (!) which explicitly licenses copying and/or deletion. In categorial grammar, the required notion of control regards not only the multiplicity of the grammatical material, but also their linear order and (prosodic) constituent structure. The logical theory of control for these linguistically relevant dimensions was worked out twenty years ago. In this talk, I will discuss some empirical consequences of this theory, focusing on the 'action at a distance' infixation and extraction operations pioneered by Emmon Bach in the 1980s. I argue that these operations should be factored in more elementary parts, and that this factorization explains a number of inherent asymmetries between OV and VO languages. The explanation assumes that the merge operation combines (prosodic) phrases, not strings. Support for this view comes from the experimental results obtained in current vector space models of compositional interpretation.

Comparing and evaluating extended Lambek calculi

Richard Moot*
CNRS (LaBRI)
University of Bordeaux

Abstract

Lambeks Syntactic Calculus, commonly referred to as the Lambek calculus, was innovative in many ways, notably as a precursor of linear logic. But it also showed that we could treat our grammatical framework as a logic (as opposed to a logical theory). However, though it was successful in giving at least a basic treatment of many linguistic phenomena, it was also clear that a slightly more expressive logical calculus was needed for many other cases. Therefore, many extensions and variants of the Lambek calculus have been proposed, since the eighties and up until the present day. As a result, there is now a large class of calculi, each with its own empirical successes and theoretical results, but also each with its own logical primitives. This raises the question: how do we compare and evaluate these different logical formalisms?

To answer this question, I present two unifying frameworks for these extended Lambek calculi. Both are proof net calculi with graph contraction criteria.

The first calculus is a very general system: you specify the structure of your sequents and it gives you the connectives and contractions which correspond to it. The calculus can be extended with structural rules, which translate directly into graph rewrite rules.

The second calculus is first-order (multiplicative intuitionistic) linear logic, which turns out to have several other, independently proposed extensions of the Lambek calculus as fragments.

I will illustrate the use of each calculus in building bridges between analyses proposed in different frameworks, in highlighting differences and in helping to identify problems.

Keywords: type-logical grammar; Lambek calculus; proof nets; linear logic

1 Introduction

The Lambek calculus (Lambek, 1958) is a landmark formal system. It is a logically simple system, with a transparent interface to natural language semantics (van Benthem, 1987) and gives a basic account of some interesting phenomena on the syntax-semantics interface, such as quantifiers and extraction. However, researchers working on the Lambek calculus soon suspected that there were problems with the calculus, both of a formal and of a descriptive nature.

*This work has benefitted from the generous support of the French agency Agence Nationale de la Recherche as part of the research project Polymnie (ANR-12-CORD-0004)

Though this was only proved in (Pentus, 1995), it has long been suspected that Lambek grammars generate only context-free languages and there are compelling arguments that natural languages require at least a slightly larger class of languages (Shieber, 1985; Joshi, 1985).

On the descriptive side, the Lambek calculus handles quantifiers and extraction only from peripheral positions: we can approximate medial extraction and medial quantifiers taking wide scope only by multiplying and complicating the lexical entries. Many other phenomena are problematic for the Lambek calculus.

Besides these problems of undergeneration, it has been known since (Lambek, 1961) that the Lambek calculus suffers from overgeneration as well. Lambek lists several problem cases, but the global availability of the associativity rule means that we predict very bad coordinations such as "*The mother of and John thinks that Bill left"[1]. Lambek's (1961) non-associative Lambek calculus, NL, solves these problems of overgeneration but in doing so makes the system too restrictive in other ways. For example the simple treatment of peripheral extraction and quantification of the Lambek calculus is no longer available in NL.

One of the main research goals in type-logical grammars has been to keep the good properties of the Lambek calculus (notably its logical simplicity and its connection to natural language semantics) while solving its empirical problems accounting for linguistic data.

In pursuit of this goal, a rather large number of logics has been proposed, which include the Lambek-Grishin calculus (LG) (Bernardi and Moortgat, 2010), the Displacement calculus (Morrill et al., 2011), multimodal categorial grammars (Moortgat, 1996c), lambda grammars (Oehrle, 1994), NL_λ (Barker and Shan, 2014), Hybrid Type-Logical Grammars (Kubota and Levine, 2012), etc. All of these logics are a shift of perspective with respect to the original Lambek calculus: for example, the Displacement calculus changes the basic objects from strings to string tuples (thereby making it natural to add logical operations for infixes and circumfixes), whereas Hybrid Type-Logical Grammars add the lambda calculus term constructors of abstraction and application (and the associated reduction operations) to the Lambek calculus connectives.

There is a rather large number of logical calculi available all claiming to improve upon the Lambek calculus in one way or another. However, this proliferation of calculi makes it hard to compare and evaluate the benefits of different calculi with respect to one another.

Given the variety of logical primitives used in these different logics, we should not expect a single "silver bullet" which relates all these different calculi. However, I discuss two "meta-logics" which at least provide two general strategies for doing such comparisons and give some examples of how to apply them.

The rest of the paper in structured as follows. I will first briefly discuss the common core shared by type-logical grammars: multiplicative intuitionistic linear logic as the syntax-semantics interface, followed by a very brief discussion of some of the phenomena in syntax and the syntax-semantics interface which we want our type-logical grammars to treat.

Then, I will present two meta-logics both employing a variant of Danos's (1990) graph contractions. The first system is a general and flexible proof net system which can be adapted to different connectives and different structural rules. The second system is first-order linear logic and it has several other type-logical grammars as a special case.

[1]Paul Dekker was the first to note this example which has been mentioned in the literature since the early 90s. Under global associativity and standard lexical assignments, both "the mother of" and "John thinks that" are expressions of type $(s \,/\, (np \setminus s)) \,/\, np$.

In what follows, I will assume the reader is familiar with the Lambek calculus and has some basic knowledge of formal semantics and of proof theory.

2 Multiplicative intuitionistic linear logic and the syntax-semantics interface

One standard architectural choice is shared between the different type-logical grammars discussed in this paper: the syntax-semantics interface takes the form of a homomorphism from proofs in the given source logic to proofs in multiplicative intuitionistic linear logic[2] (also called the Lambek-van Benthem calculus LP, in the categorial grammar literature).

In what follows, we will call the multiplicative intuitionistic linear logic proof the *deep structure* of a sentence and the proof in the source logic the *surface structure*. From the deep structure we obtain the meaning of a sentence by substituting the semantic terms from the lexicon and normalising the result term.

Seen from this perspective, different type-logical grammars generally agree on the deep structure for an analysis, but arrive at this deep structure from different surface structures. As Moortgat (1997, Section 3) remarks, the tension between the LP proofs of deep structure and the more restricted proofs of surface structure has played an important role in the development of type-logical grammars. We have to find a delicate balance between avoiding overgeneration and generating all the desired readings of our sentences.

3 A catalogue of problem cases

What are the linguistic data which motivated these new calculi? The following list gives some of the types of phenomena people have look at. Since a finite list of cases can always be treated by some additional lexical type assignments, we will be interested only in *robust* solutions, that is solutions which generalise beyond the listed examples to more complex cases, and ideally to different *phenomena*. In other words, we want to avoid ad hoc solutions (though, as an illustration, I will sometimes give an additional assignment which solves a particular case).

As usual, an asterisk "*" before a sentence denotes ungrammaticality.

(1) John loved but Peter hated "Syntactic Structures".

(2) *Peter bough the book which John read "Syntactic Structures" and Mindy liked.

(3) Peter bough the book which John read yesterday.

(4) John believes someone left.

(5) (dat) Jan Henk Marie de nijlpaarden zag helpen voeren.

(6) John studies logic and Charles, phonetics.

(7) John left before Mary did.

(8) Mary gave more girls a book than boys a record.

[2]The alternative is to use a simple, applicative logic and to add a set of type-shifting rules in the spirit of combinatorial logic. Two types categorial grammars using this alternative setup are combinatory categorial grammar (Steedman, 2001, CCG) and flexible Montague grammar (Hendriks, 1993). The choice of multiplicative intuitionistic linear logic also has some descriptive repercussions: as discussed in Section 3, the treatment of parasitic gapping and some treatments of anaphora and binding are incompatible with this choice and require a more powerful logic.

Example (1), called "right node raising" in the literature, shows one of the things which works correctly in the Lambek calculus: assuming assignments of $(np \setminus s) / np$ to the transitive verbs and np to "John" and "Peter", L derives "John loved" and "Peter hated" as expressions of type s / np. Though this works correctly, it crucially depends on the presence of associativity. For example, NL no longer allows us to derive sentence (1) unless we add a second lexical assignment $np \setminus (s / np)$ to the transitive verbs.

Examples like (2), and related examples, were first discussed in (Lambek, 1961, p. 167). The problem with this ungrammatical sentence is that "John read *Syntactic Structures* and Mindy liked" is a sentence missing a noun phrase (at its right edge) and can therefore combine with "which" given its standard Lambek calculus assignment $(n \setminus n) / (s / np)$. The same derivability pattern which helped us for sentence (1) leads to overgeneration for sentence (2). The ungrammaticality of this sentence is usually ascribed to "island constraints".

Examples (3) and (4) illustrate that the Lambek calculus has problems with quantifiers in medial position taking wide scope ("someone" in the *de re* reading of example (4)) and extraction from non-peripheral positions: sentence (3) is derivable without the sentence-final adverb "yesterday", again since "John read" is an expression of type s / np. However, the presence of "yesterday" blocks this derivation.

Another classic example is the treatment of Dutch verb clusters (Moortgat and Oehrle, 1994; Oehrle, 2011) (Sentence (5) above). This sentence exhibits the well-known crossed dependencies: "Henk" is the object of "zag" (saw), "Marie" the object of "helpen" (help) and "de nijlpaarden" (the hippopotami) the object of "voeren" (feed). Though Pullum and Gazdar (1982) show that such examples can be treated by context-free grammars, and hence by the Lambek calculus, the Dutch verb cluster analysis of mildly context-sensitive formalisms, which expresses the desired dependencies between objects and verbs is generally preferred.

Gapping (Hendriks, 1995) (Sentence (6), which has the same meaning as "John studies logic and Charles *studies* phonetics"), ellipsis (Sentence (7), which has the same meaning as "John left before Mary *left*") and comparative subdeletion (Sentence (8), which means "The number of girls Mary gave a book is greater than the number of boys *Mary gave* a record"), are some of the other interesting phenomena on the syntax-semantics interface which have been treated in type-logical grammars.

One general type of example where type-logical grammars do well are coordination and similar phenomena. The type-logical grammar treatment of coordination tells us we can conjoin two phrases whenever they can be assigned the same formula. Sentence (1) is an example, but also, in the analysis of Hendriks (1995), Sentence (6)[3]. The easiest way to implement this is to assign the second-order formula $\forall X.((X \setminus X) / X)$ to words like "and" and "but", though how to add second-order quantifiers while keeping type-logical grammars decidable is an open question.

A general type of extension to the Lambek calculus is exemplified by Moortgat's (1996a) $q(A, B, C)$ constructor, which has a rather simple left rule.

$$\frac{\Delta[A] \vdash B \quad \Gamma[C] \vdash D}{\Gamma[\Delta[q(A, B, C)]] \vdash D}\ qL$$

Its instantiation $q(np, s, s)$ gives a good account of quantifier scope, whereas reflexives can be assigned $q(np, np \setminus s, np \setminus s)$. In addition, it can be extended to treat comparative subdeletion and other phenomena as well (Sentence (8)) (Moortgat, 1996a). Carpenter's

[3]Though in the case of (Hendriks, 1995) and also in the similar case of (Morrill et al., 2011) this is a coordination of two formulas which are *almost* the same.

Formalism	Islands	RNR	Rel	q	$\wedge$	gap	Language	Complexity
NL	+	−	−	−	+	−	= CFL	P
L	−	+	−	−	+	−	= CFL	NP
D_{core}	−	+	+	+	+	+	$\supseteq$ MCFL_{wn}	NP
D_{full}	+	+	+	+	+	+	$\supsetneq$ MCFL	?
ACG_2	−	−	−	−	−	−	= MCFL	NP
ACG	−	−	+	(+)	−	−	$\supseteq$ MCFL	NP
LG	−	−	−	+	+	+	$\supsetneq$ MCFL	NP
NL_λ	+	−	+	+	+	+	$\supsetneq$ MCFL	NP(?)
HTLG	−	+	+	+	+	+	$\supsetneq$ MCFL	NP
MILL1	+	+	+	+	+	+	$\supsetneq$ MCFL	NP
$\mathrm{NL}\Diamond_{\mathcal{R}}$	+	+	+	+	+	+	= CSL	PSPACE

Table 1: Scorecard for the different extended Lambek calculi

(1998) textbook gives an introduction to semantics and type-logical grammar using mostly the Lambek calculus extended with the q operator. Now, semantically, the q operator behaves like a combination of an introduction rule and an elimination rule, so it seems the q connective may need to be decomposed and though we have given a left rule for q there is no easy way to formulate its *right* rule for q. Because of the simplicity and many applications of the q operator, these questions have led to many decompositions of q (Morrill, 1994; Moortgat, 1996b; Barker and Shan, 2014, to cite but a few)

The above list of sentences gives only a very partial picture of the descriptive work done in the various type-logical frameworks, but it provides a starting point for discussion.

Table 1 is my attempt to, as Lakatos (1978) suggests, keep score honestly. It is a table cross-referencing formalisms and phenomena, assigning a "+" if the formalism has a good treatment of the given empirical data and "−" if it does not.

The formalisms are, from top to bottom: the non-associative Lambek calculus, NL (Lambek, 1961), the associative Lambek calculus L (Lambek, 1958), the core, multiplicative fragment of the Displacement calculus, D_{core} (Morrill et al., 2011, this includes the multiplicative connectives, with the synthetic connectives restricted to $\hat{}$, $\triangleright^{-1}$ and $\triangleleft^{-1}$, when I use the term "D" or "the Displacement calculus" without further qualification in the rest of this paper, I mean this fragment), the full Displacement calculus, D_{full} (Morrill and Valentín, 2015, this volume). Abstract categorial grammars/lambda grammars, ACG and their second-order restriction ACG_2 (de Groote, 2001; de Groote and Pogodalla, 2004), the Lambek-Grishin calculus, LG (Bernardi and Moortgat, 2010), NL_λ (Barker and Shan, 2014), Hybrid Type-Logical Grammars, HTLG (Kubota and Levine, 2012, 2013), first-order linear logic, MILL1 (Moot and Piazza, 2001) and the multimodal Lambek calculus $\mathrm{NL}\Diamond_{\mathcal{R}}$ (Moortgat, 1996c, 1997).

The phenomena are, from left to right: does the formalism have a way, such a non-associativity, to encode island constraints? Does the formalism have a way to encode right-node raising (RNR)? Does the formalism have a way to handle medial extraction? Does the formalism have a way to simulate the q operator[4]? Does the formalism handle coordination ($\wedge$)[5]? Does the formalism treat gapping and related phenomena? What do we know about the language class of the given formalism (the classes are context-free

[4]A "(+)" means the formalism can only do so partially. For example ACG does not handle $q(A, B, C)$ for complex formulas, for example those used for reflexives (Moot, 2014b).

[5]Coordination for atomic formulas only is marked as "−", a "+" for coordination means: "the formula $\forall X.((X \setminus X) \,/\, X)$ for conjunction works correctly for all formulas X in our lexicon"

languages, well-nested multiple context-free languages, multiple context-free languages and context-sensitive languages)? And finally, what is the computational complexity of the formal system?

Our knowledge of the precise language classes is very partial: this is partly due to the difficulty of extending a proof like (Pentus, 1995) to extended Lambek calculi (Buszkowski, 1997), which means few upper bounds are known and in general few formal proofs of generated language classes exist and the table entries represent answers to questions like "are there grammars of the language $a^n b^n c^n$, and ww?". Formalisms which generate (at least) the well-nested multiple context-free languages can correctly treat phenomena such as the Dutch verb clusters discussed above.

Not listed in the table is parasitic gapping. The full Displacement calculus with contraction modalities is the only calculus among those listed which has a treatment of parasitic gapping, as exemplified by a noun such as.

(9) contract which John filed without reading.

In these cases, the object noun phrase bound by "which" has been extracted twice, once as the object of "filed" and once as the object of "reading".

Also absent from the table are treatments of anaphora rooted in logical syntax, such as those of (Jäger, 2001; Morrill and Valentín, 2014; Barker and Shan, 2014). Like parasitic gapping, such treatments require a modification of the hypothesis, discussed in Section 2, of a linear syntax-semantics interface and like parasitic gapping, they complicate the proof theory. I believe that for anaphora (and binding theory) the proper division of labour between syntax, syntax-semantics interface and semantics/pragmatics is still very much open for debate and that integrating the many interacting facts on these different levels into a single, coherent system will be a difficult but important task.

The computational complexity column lists the complexity of the parsing or universal recognition problem (fixed recognition is a property of formal languages and has little relation to parsing complexity). All results listed "NP" or "PSPACE" indicate NP- or PSPACE-completeness. The complexity of the full Displacement calculus is unknown, and may be undecidable. The complexity of NL_λ is unknown; Barker and Shan (2014) show only decidability, but the close connection to HTLG — both employ a combination of a standard Lambek calculus (L and NL respectively) together with lambda-calculus like operations (exactly those of the lambda calculus for HTLG and operations very close to it in NL_λ) — makes it likely to be an NP-complete system (since it appears polynomial time verifiable whether or not an HTLG proof is also an NL_λ proof[6]). NL can be parsed in polynomial time (Aarts and Trautwein, 1995; de Groote, 1999).

Though it is easy to criticise this table and talk about important missing phenomena or incorrectly assigned entries, I hope this will provide a starting point for discussion and provide an impetus for people to "improve their score" on the table. I would like to add that systems with a "low score", such as the Lambek calculus, are still worthwhile systems to use and study, as long as we are aware of their limitations for certain applications. My goal is to allow "people to do their own thing but only as long as they publicly admit what the score is between them and their rivals" (Lakatos, 1978, p. 100).

[6]Of course this presupposes first that NL_λ proofs are a subset of the HTLG proofs, in the same sense that NL proofs are a subset of L proofs.

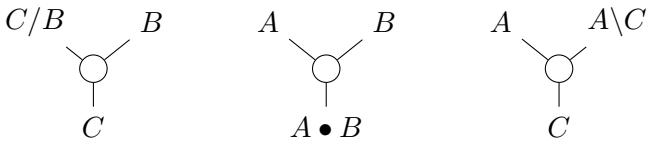


Figure 1: Binary branches and the connectives of NL

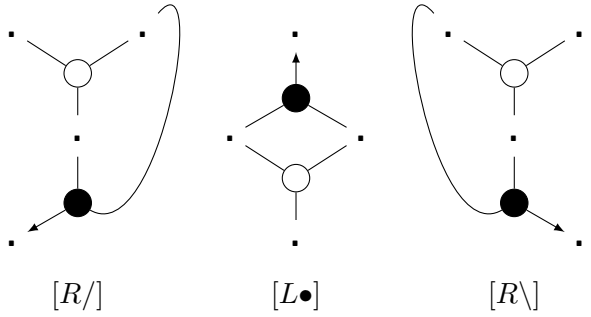


Figure 2: Contractions — Lambek connectives

4 Graph rewriting

The first meta-logic for creating bridges between the different syntactic derivations of extended Lambek calculi is a simple generalisation of the proof net calculus of Moot and Puite (2002). Its basic idea is that you only need to specify the *structures* of your logic (ie. the way your formulas are organised in the sequence, for example, as a list, a tree, or as a graph) and then you will automatically obtain the possible connectives and a correctness criterion based on graph rewriting and contractions in the style of Danos (1990) and close to the interaction nets of (Lafont, 1995).

For example, if we choose binary branching trees as structures, then we obtain three connectives depending on which of the three related nodes contains the main formula and these are just the familiar connectives of the non-associative Lambek calculus NL.

In addition to the constructors shown in Figure 1, each connective, by logical symmetry, induces a destructor[7] as well. These destructors have the property that an appropriately connected constructor/destructor pair contracts to a point. When it is impossible to contract a destructor, the graph does not correspond to a provable statement; provable statements contract to a (constructor) tree.

The contractions for NL are shown in Figure 2. Each destructor is drawn in black and has a single outgoing arrow (this arrow points to the main formula of the link).

We can contract only when the branches of a constructor and destructor are connected by the two branches without the arrow. For the $[L\bullet]$ contraction, this means we connect the two left branches to each other and similarly for the two right branches. The contractions for the implications look a bit odd when we draw them on the plane and preserve left/right and top/bottom of the links. The $[R/]$ contraction connects a constructor and destructor

[7]For the binary connectives of linear logic, the constructors are tensor links and the destructors are par links. In the terminology of focusing proof search (Andreoli, 1992), we would talk about synchronous or non-invertible rules instead of constructors and asynchronous or invertible rules instead of destructors.

top-to-bottom while connecting the two right branches (the left branch has the arrow pointing towards it). Once we have connected top and bottom, we are forced to bend one of the right branches (alternatively, we could write the graph on a cylinder). The idea behind the $[R/]$ contraction is that it withdraws a hypothesis occurring as the immediate right daughter of the root of the tree, just like the sequent calculus or natural deduction rule for NL would do. The bent link can be seen as a graphical representation of the coindexing of the hypothesis with the rule application for the $/I$ rule. By simple pattern-matching with Figure 1 we can see that these simple instances of the $[R/]$ contraction corresponds to proofs of $A \,/\, B \vdash A \,/\, B$, $A \vdash (A \bullet B) \,/\, B$ and $A \vdash C \,/\, (A \setminus C)$.

The important idea of take away from this is that we specify our structures (in the case of NL, these were simple binary branching trees), and that from this we automatically obtained three connectives, where each of the three possible "main edges" induced a corresponding contraction.

Moving from NL to the multimodal Lambek calculus $\mathrm{NL}\Diamond_{\mathcal{R}}$ means changing our structures from unlabelled binary trees to labelled binary-unary trees, that is labelled trees where both binary and unary branches are allowed. The unary branches corresponds to the connectives $\Box$ and $\Diamond$, and the labels, which we draw inside the central circle of a branch, correspond to mode information. The contractions have the additional condition that the modes of the constructor and destructor must be the same.

The final component of the calculus is the set of structural rules, which are rewrites from one subtree of the graph to another. The unary branches and the mode information control which structural rules can apply, so we can have a non-associative base logic yet still handle right-node-raising and quantifier scope (Kurtonina and Moortgat, 1997; Moortgat, 1996b). Another way to see the structural rules it that they specify a partial order on our structures.

For reasons of space, this introduction to the graph rewriting approach to proof nets is a bit impressionistic, for much more detail and the fundamental results, the reader is referred to (Moot and Puite, 2002; Moot and Retoré, 2012). Though created as a proof net calculus for multimodal categorial grammars, it can easily be generalised to handle the Lambek-Grishin calculus, which adds up-down symmetric structures to NL (see the left hand side of Figure 3), and the Lambek calculus with Galois connections (Areces et al., 2001), which adds negation-like polarity changes to the calculus (see the middle of Figure 3). Other calculi also have a presentation which is equivalent to a multimodal calculus, so we can use this proof net calculus also for Valentín's (2014) multimodal version of the Displacement calculus and for NL_{CL}, a multimodal version of NL_{λ} (Barker and Shan, 2014, Chapter 17), which adds zero-ary connectives ("constants") to the multimodal calculus. All these proof net calculi are simple instances of the general methodology: specify the structures (and possibly a partial order on them) and obtain the connectives and the contractions for a proof net calculus (Moot, 2007).

In the next sections, I illustrate with the help of two examples how this proof net calculus allows us to compare different systems.

4.1 Lambek-Grishin and Galois connections

First, we can see that the non-associative Lambek calculus (NL) extended with Galois connectives has the Lambek-Grishin calculus as a special case. As Figure 3 shows, LG extends NL with up-down symmetric connectives whereas the Galois connectives give a negation-like polarity change operation. In particular, as shown in the center of Figure 3, this allows us to simulate the Grishin connectives: a combination of an NL constructor with a Galois connective as shown in the figure forms a subgraph which to the outside "looks

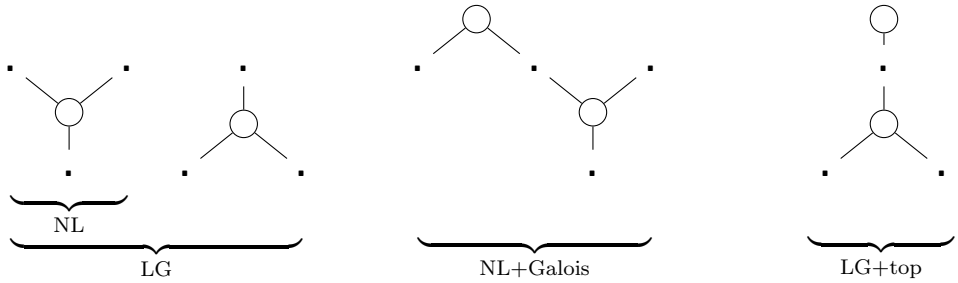


Figure 3: NL, LG and Galois connections

the same", up until the cyclic order of the external vertices, as the Grishin connective (there is an alternative solution, which is left-right symmetric to the one shown).

This mapping goes only in one direction, however. The translation back to LG is partial: though the images of translations of LG graphs are sent back to the original graph, a graph consisting of a single Galois connective cannot be translated to LG, since only the "unit" displayed in the center of the figure is a meaningful candidate for such a translation.

However, suppose there is something interesting in NL+Galois and we are interested in translating this analysis to LG. The graphs immediately suggest the solution shown on the right of Figure 3: a combination of a Grishin connective with a 0-ary "top" connective can function as a Galois connective. Like before the two graphs look the same to the outside world: both have no premisses and two conclusions in the same order (a 0-ary connective is not the same as an identity element: if we want such a connective to function as an identity element for a binary connective we need to add the appropriate structural rules, an example of such a structural rule can be found on the left of Figure 6 below).

To complete the picture, we have the following relations between NL, LG and the Galois/top connectives ($A \subset B$ means here that we can translate structures of logic A into structures of logic B in such a way that logic B is a conservative extension of logic A).

$$\text{NL} \subset \text{LG} \subset \text{NL+Galois} \subset \text{LG+top} = \text{NL+Galois+top}$$

If we want a fully up-down symmetrical system, we need to complete the system with dual-Galois connectives and a bottom connective in the obvious way.

4.2 Multimodal versions of q

As a second illustration, Moortgat (1996b) and Barker and Shan (2014) both provide an implementation of the in situ binder $q(A, B, C)$, shown in Figure 4 and Figure 5 on the right and center resp. left and center (both have an additional "start" rule shown in Figure 6, discussed below). For the rules of Figures 4 and 5 the two substructures displayed in gray give us a mapping from one system to the other. This highlights both similarities and differences between the two approaches: the Barker and Shan implementation, which they call NL_{CL}, works unchanged when multiple in situ binders "overlap", as used for their analysis of the semantics of "*same*", whereas Moortgat's approach works correctly even in the presence of associativity. Both properties are a consequence of the unary branches and their position in Moortgat's calculus: the unary branches block associativity, but also

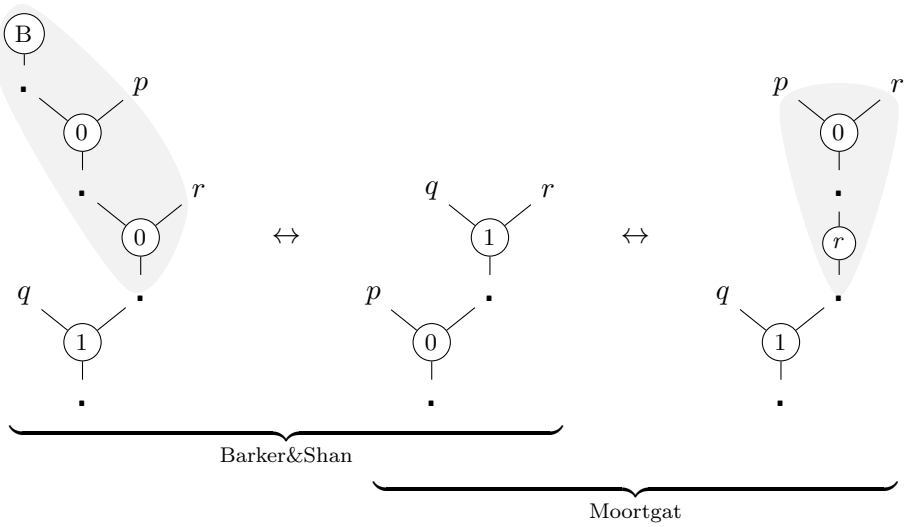


Figure 4: Mixed commutativity in (Barker and Shan, 2014) compared to (Moortgat, 1996b)

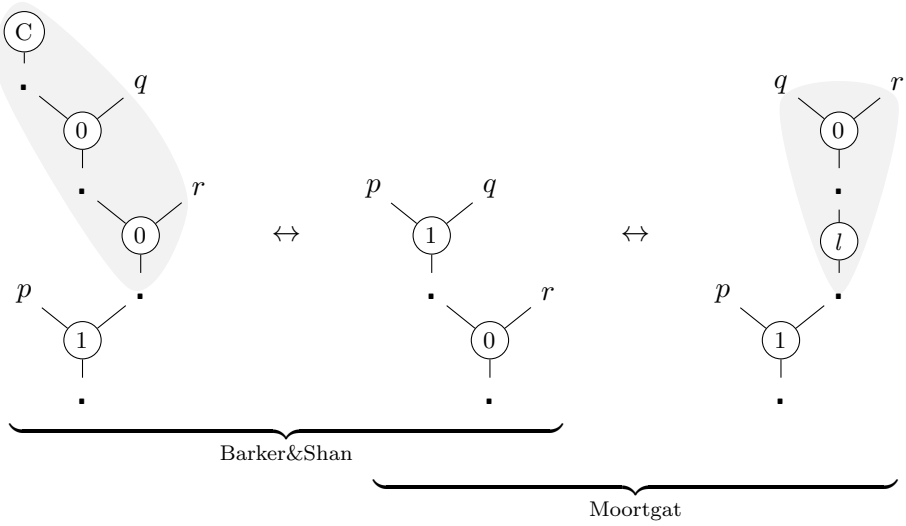


Figure 5: Mixed associativity in (Barker and Shan, 2014) compared to (Moortgat, 1996b)

prevent the rules from operating in the direction from center to right when there is a unary branch between the two constructors.

In Moortgat's original system, the unary branches leave a "trail" telling us at each binary branch whether to go left or right. In the Barker and Shan system, there is a similar trail but stored at a deeper position. Barker and Shan's system breaks down when we add associativity to the system — though the addition of modally controlled associativity seems delicate even in Moortgat's system and needs to be carefully investigated.

Finally, there are the start rules of Figure 6. For the Barker and Shan system the I connective is simply an right identity element for the binary mode 1 (the structural rule looks strange compared to the ones we've seen before, but it is of the same shape: we replace a subtree with leaf p, binary mode 1 and 0-ary mode I by the leaf p, which is a

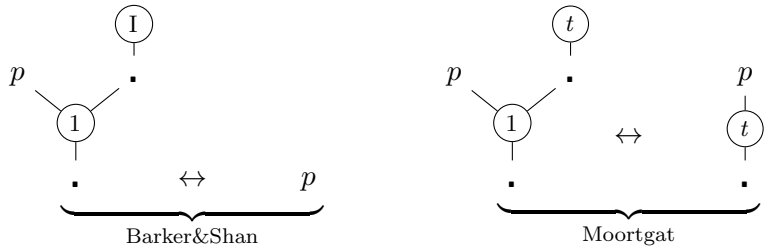


Figure 6: Start rules of (Barker and Shan, 2014) compared to (Moortgat, 1996b)

subtree with the same leaves). In Moortgat's version, we replace a unary branch t by a binary branch with a right identity element t (t occurs both as a unary branch and as a 0-ary branch in this rule). This formulation has the advantage that we can lexically control the application of the rule (the current form of these rules is at least partially motivated from the point of view of facilitating automated deduction for the system). For the Barker and Shan implementation, we need to be careful when applying the rule from right-to-left, since naive application of this rule can lead us into an infinite loop.

To intertranslate the start rules, we can either use the fact that we can eliminate "useless" applications of the I rule from NL_{CL} — we know from the embedding results of Sections 17.5 and 17.7 and Definition 304 on page 172 of (Barker and Shan, 2014) that we can, without loss of generality, restrict NL_{CL} structures containing I only at the end of a path of B and C nodes — or import Moortgat's analysis directly into NL_{CL}. In either case, it is then a simple translation between 0-ary I and 0-ary t.

Looking at Figures 4 and 5 also shows us exactly when we can translate NL_{CL} structures with BCI connectives to Moortgat's calculus: only when B and C nodes occur as the right daughter of two consecutive $\circ_0$ branches, that is only when they occur as $(B \circ_0 X) \circ_0 Y$ or $(C \circ_0 X) \circ_0 Y$ (for some structures X and Y). But this property crucially fails when there are multiple binders, such as in the term below, which occurs in the analysis of "the same waiter served everyone" on page 166 of (Barker and Shan, 2014).

$$\mathit{same} \circ_1 ((C \circ_0 ((B \circ_0 \mathbf{B}) \circ_0 ((B \circ_0 \mathit{the}) \circ_0 ((C \circ_0 I) \circ_0 \mathit{waiter})))) \circ_0 ((B \circ_0 \mathit{served}) \circ_0 I))$$

In the above tree (presented as a flat term to preserve space) the B indicated in bold is not translated when we simply intertranslate the structures indicated in gray in Figures 4 and 5, since it occurs on a right branch. The translation produces the following term which still contains this B which is not a part of Moortgat's calculus (the circumfix $\langle . \rangle^i$ denotes a unary branch with label i).

$$\mathit{same} \circ_1 \langle \langle \mathbf{B} \circ_0 \langle \mathit{the} \circ_0 \langle t \circ_0 \mathit{waiter} \rangle^l \rangle^r \rangle^r \circ_0 \langle \mathit{served} \circ_0 t \rangle^r \rangle^l$$

In the Barker and Shan analysis, this B is part of the second path; after infixation of "same" using the structural rules, we obtain the following term, with the same B shown in bold.

$$(\mathbf{B} \circ_0 (\mathit{the} \circ_0 (\mathit{same} \circ_0 \mathit{waiter}))) \circ_0 ((B \circ_0 \mathit{served}) \circ_0 I)$$

This second term translates unproblematically to the following term.

$$\langle (\mathit{the} \circ_0 (\mathit{same} \circ_0 \mathit{waiter})) \circ_0 \langle \mathit{served} \circ_0 t \rangle^r \rangle^r$$

Though this analysis does not immediately show us the best way to extend Moortgat's calculus to handle multiple binders, it does give us an indication of where to look: we want

to extend the translation in such a way that B terms can no longer appear and which allows us to rewrite our new terms, no longer containing B, to the term above. It is not hard to add modes and structural rules to allow exactly this, but I'll leave more elegant solutions to future research.

4.3 Graph rewriting, graph grammars and modal logic

I have briefly talked about how graph rewriting can be a tool for providing mappings from one type-logical grammar to another. There are some other connections I briefly want to mention.

Since most type-logical grammars have models in modal logic, can we use some of the rich set of tools of modal logic to discover relations between different formalisms? For example by using bisimulations between the model-theoretic interpretations of two calculi (Blackburn et al., 2001). This would be a model-theoretic counterpart to the proof-theoretic approach of the current paper. Also, since the modern view of modal logic sees modal logic as (decidable) fragments of first-order logic this ties in nicely with the first-order perspective of the next section[8].

A second point is that graph rewriting has been extensively studied in computer science. However, these tend to be nicer classes of graphs than those presented here. Is there a connection between some context-free graph grammars (Engelfriet, 1997) and the proof nets studied here or do we, as seems likely, need richer classes? If so, which ones and what are their properties? Some preliminary investigations into this topic can be found in (Moot, 2008).

5 First-order linear logic

The second meta-logic is first-order (multiplicative, intuitionistic) linear logic. Though the idea of adding first-order quantifiers to type-logical grammars is very old (Morrill, 1994, Section 6.2), the important shift in Moot and Piazza (2001) was that the first-order quantifiers could also be used for word order. So instead of adding first-order quantifiers to one of the other variants of type-logical grammar, we can add first-order quantifiers to multiplicative linear logic and embed the Lambek calculus. Several of the known problems of the Lambek calculus have a simple solution in first-order linear logic. First-order linear logic also has a very nice, clean proof net calculus (Girard, 1991).

5.1 Proof search in first-order linear logic

I will give a very brief introduction to first-order linear logic from the point of view of proof search and parsing. Figure 7 presents the logical links for first-order linear logic proof nets. To try and prove a sequent $A_1, \ldots, A_n \vdash C$ we decompose the formulas according to the links of the figure, starting with

$$\overset{-}{A_1} \ldots \overset{-}{A_n} \overset{+}{C}$$

and continuing the unfolding until we reach the atomic subformulas. The links essentially produce a subformula tree with the additional marking of the polarity of each subformula and with some dotted lines. There are two types of dotted lines: those which pair two branches (for the positive $\multimap$ and negative $\otimes$ link) and those which are labelled with the

[8] I thank one of the anonymous referees for pointing out the connection to modal logic.

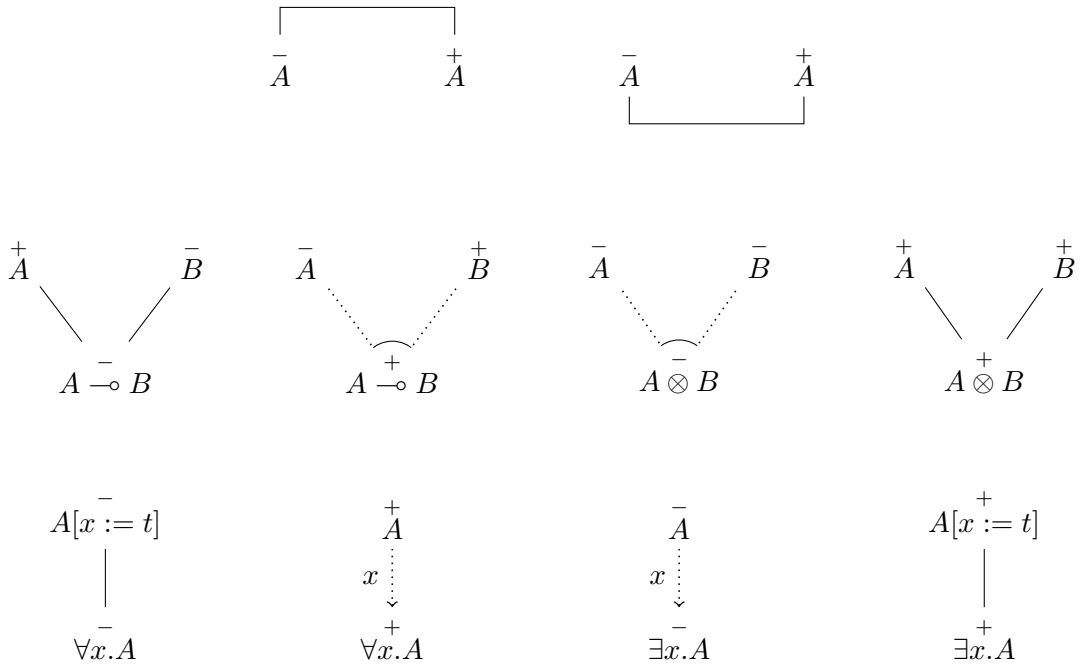


Figure 7: Logical links for MILL1 proof structures

eigenvariable of a quantifier (for the positive $\forall$ and the negative $\exists$ link). We use the standard convention that each quantifier in a sequent has a different eigenvariable.

After this unfolding step, we connect atomic formulas of opposite polarity using the axiom link (Figure 7, top left). When all atomic formulas have been connected this way, the resulting structure is called a proof structure. As an example, the sequent $\forall x.a(x) \multimap b \vdash \exists y.[a(y) \multimap b]$ has the proof structure shown in Figure 8.

In practice, we do not choose a term for the negative $\forall$ or for the positive $\exists$ link during formula unfolding but we rather use meta-variables for unfolding and unification during the axiom link connections. So the axiom link does not connect identical formulas but rather unifiable formulas and performs this unification. This is a rather standard theorem-proving strategy and has the result that we can read off the most general term for each negative $\forall$ and positive $\exists$ rule in our proof net. Some care must be taken when we repeatedly unify the positive and negative atomic subformulas of $\forall x.a(x) \multimap a(f(x,x))$, where the size of the term argument grows exponentially in the number of occurrences of the given formula ($a(x)$, $a(f(x,x))$, $a(f(f(x,x),f(x,x)))$, $a(f(f(f(x,x),f(x,x)),f(f(x,x),f(x,x))))$, ...) . Even in these cases, we can ensure linear time unification by adopting a sharing strategy (Martelli and Montanari, 1982; Patterson and Wegman, 1978). In addition, for the typical uses of first-order linear logic which interest us here, each quantifier binds at most two occurrences of its eigenvariable (Moot, 2014a), so we might even decide to exclude the case above, where the quantifier $\forall x$ binds three occurrences.

Returning to the example proof structure of Figure 8, the given sequent is underivable in linear logic and, in general, proof structures need not correspond to proofs. Proof structures which correspond to proofs are called *proof nets* and we can distinguish them from other proof structures using properties of the graph. Girard (1991) gives a switching criterion whereas Moot (2014a) presents a contraction criterion for first-order linear logic in the style of Danos (1990).

The contractions for first-order linear logic are shown in Figure 9. As a first step, we

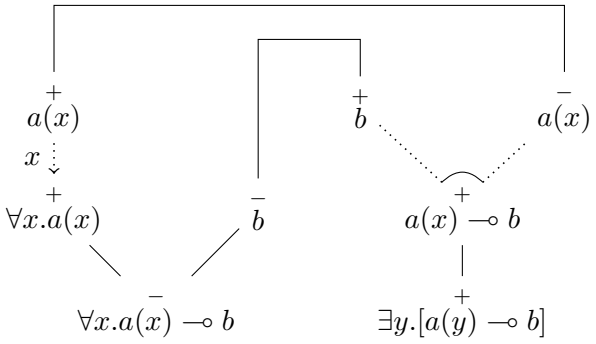


Figure 8: Proof structure which is not a proof net

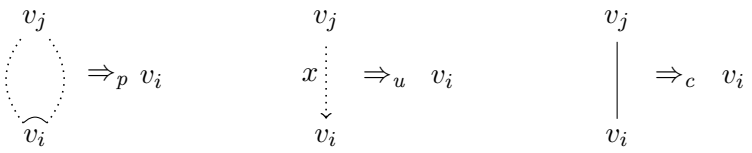


Figure 9: Contractions for first-order linear logic. Conditions: $v_i \neq v_j$ and, for the u contraction, all free occurrences of x are at v_j.

forget about all formulas in the proof structure keeping track only of the free variables at each vertex in the graph. For Figure 8 this produces the graph shown on the left hand side of Figure 10. Then we progressively shrink the proof structure using the contractions shown in the figure. Each contraction removes an edge (a linked pair of edges in the case of the p contraction) and identifies the two nodes which were connected by it. A contraction can only be performed on two distinct vertices, that is, we are not allowed to eliminate self-loops. The free variables of the result vertex are the union of the sets of free variables of the two input vertices (in the case of the u contraction, we can remove the x variable, since it has become redundant). A proof structure is a proof net if and only if it contracts to a single point using the contractions of Figure 9.

As an example, Figure 10 shows the contractions performed on the proof structure of Figure 8, with the initial structure on the left and the structure after all c contractions on the right. The arrow and eigenvariable of the $\forall$ link and the connection between the

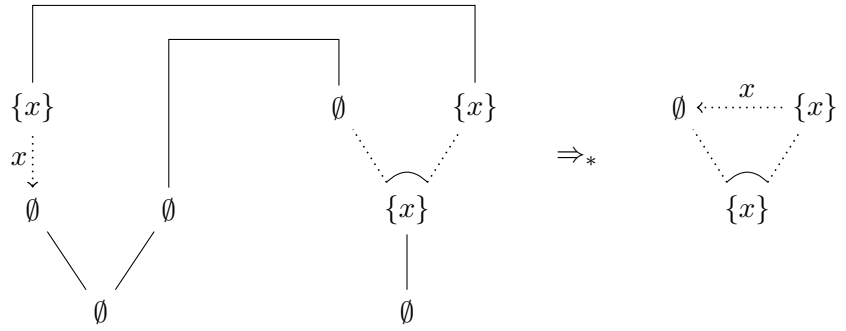


Figure 10: Contractions for the proof structure of Figure 8

two other dotted links ensure the notation is unambiguous. The displayed graph is not a single vertex but it cannot be further contracted: the universal contraction u cannot apply since the variable x occurs at the bottom vertex instead of only at the right vertex as required for the contraction and the contraction p cannot apply until its two branches end at the same vertex. Since the contractions of Figure 9 are confluent[9], any graph which is not further contractible but is not a unique vertex, such as the one shown at the right of Figure 10, suffices to show that the given sequent is underivable.

Summarising, parsing/proof search in first-order linear logic operates as follows.

1. For each word in the sentence, we find a first-order formula in the lexicon.
2. We unfold a sequent using the rules of Figure 7.
3. We connect atomic formulas of opposite polarity, unifying variables.
4. We contract the resulting proof structure to a single vertex.

Combinatorially, the complex steps are step 1 (in the case of high lexical ambiguity) and step 3 where we connect the atomic formulas. For an actual implementation, such as (Moot, 2015), it is therefore desirable to contract early — thereby keeping a compact representation of the current state of the proof — and develop ways of detecting "doomed" configurations, that is graphs which can never be contracted to a single vertex, no matter how we continue the construction of our proof structure. Examples of such configurations are connections between a node and its ancestor with a path of dotted links (this corresponds to a cycle in the proof structure and though we can validly reduce the size of this cycle, such a configuration will, at best, end up producing a self-loop) or isolated vertices (an isolated vertex is a vertex which is not connected to the rest of the graph but which also doesn't have any unlinked atomic formulas; unless it is the last vertex of the graph, such a vertex corresponds to a disconnected proof structure). Combining early failure with a smart backtracking strategy for selecting which atomic formulas to unify (Knuth, 2000) produces an effective algorithm, though I suspect many improvements are still possible.

5.2 First-order linear logic as a grammar formalism

Whereas Moot and Piazza (2001) show that the Lambek calculus has a natural translation into first-order linear logic, in (Moot, 2014a,b), this idea is further developed and translations are given which show that Displacement grammars, Hybrid Type-Logical Grammars and lambda grammars are all natural fragments of first-order linear logic, providing much simpler proof-theoretic foundations for these calculi. In addition, the analyses proposed in these different frameworks agree to a large extent upon translation into first-order linear logic. The basic idea of the translation into first-order linear logic is that formulas are assigned pairs of string positions. For example, an atomic Lambek calculus formula np becomes an atomic first-order linear logic formula $np(0,1)$ when it spans the string from position 0 to 1. Similarly, the Lambek calculus formula $np \setminus s$ spanning the string from 1 to 2 becomes the formula $\forall x.np(x,1) \multimap s(x,2)$, indicating it is looking for a noun phrase to its right to form a sentence. Instantiating x to 0, we can combine it with $np(0,1)$ to derive $s(0,2)$, a sentence spanning string positions 0 to 2. The translations for Hybrid

[9] As discussed in (Moot, 2002), this is not true for the contractions of Section 4 which require us to explore the entire search space to show underivability. The culprits in that case are the unary contractions and the structural rules.

Type-Logical Grammars and the Displacement calculus, though based on essentially the same idea, are a bit more involved and will not be repeated here.

Compared to the graph rewriting of Section 4, this setup has the advantage that instead of comparing *structures* and giving mappings from structures to structures — which has the difficulty that we need to show these structures behave the same in *all* contexts — we can translate to first-order linear logic and compare *formulas*. This is much easier and more immediate: we translate to first-order linear logic and compare the formulas we obtain. So we can see, for example, that the treatment of gapping in Hybrid Type-Logical Grammars and the one of the Displacement calculus, in spite of being formulated using very different logical primitives, are actually *equivalent* upon translation into first-order linear logic (in the sense that the translated formulas are interderivable). There is no need to specify a translation of the primitives of one calculus into another nor to provide a mapping from proofs to proofs.

As a simple example, the lambda grammar/ACG lexical entry $np \multimap s$ with prosodic term $\lambda S.(S + sleeps)$ becomes, according to the translation into first-order linear logic, the formula $\forall x.np(x,1) \multimap s(x,2)$, just like the Lambek calculus (and Displacement calculus) formula $np \setminus s$. Even though these translations follow rather different paths, they end up at the same destination and it is this agreement on many of the "basic" lexical entries which forms the basis of the comparison of formalisms using first-order linear logic.

5.3 Relative pronouns

As a more interesting example of this way of comparing formulas, here are five different first-order linear logic formulas expressing extraction. These formulas would be assigned to a relativiser such as "which" occurring at position 3-4.

(1) $\forall x_0.[(\forall x_1.[np(x_1,x_1)] \multimap s(4,x_0)) \multimap \forall x_2.[n(x_2,3) \multimap n(x_2,x_0)]]$

(2) $\forall x_0.[\exists x_1.[np(x_1,x_1) \multimap s(4,x_0)] \multimap \forall x_2.[n(x_2,3) \multimap n(x_2,x_0)]]$ D

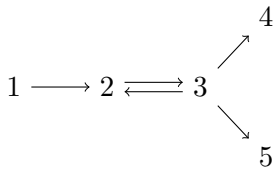
(3) $\forall x_0 \forall x_1 \forall x_2.[((np(x_1,x_1) \multimap s(4,x_0)) \multimap (n(x_2,3) \multimap n(x_2,x_0))]$ λ-grammar

(4) $\forall x_0.[\forall x_1.[np(x_1,4) \multimap s(x_1,x_0)] \multimap \forall x_2.[n(x_2,3) \multimap n(x_2,x_0)]]$ L : $(n\backslash n)/(np\backslash s)$

(5) $\forall x_0.[\forall x_1.[np(x_0,x_1) \multimap s(4,x_1)] \multimap \forall x_2.[n(x_2,3) \multimap n(x_2,x_0)]]$ L : $(n\backslash n)/(s/np)$

The first three formulas, though with slightly different scopes for the quantifiers, intuitively mean that a relative pronoun spanning positions 3-4 is looking to its left for a noun n (a noun spanning positions x_2-3 for some x_2 of our choice) and to its right for a sentence s, which itself is missing a noun phrase anywhere (where this sentence spans positions 4-x_0 for some x_0 of our choice, the relation between the position of the np and this sentence is not specified, though the proof theory will ensure this np will occur "inside" the sentence). The result will be a noun from position x_2, the start of the n argument, to x_0, the end of the s argument.

The first formula is a possibility which I have not seen before. Formula 2 is the formula from (Moot and Piazza, 2001) as well as the translation into first-order linear logic of the extraction formula for the Displacement calculus (D). Formula 3 is the translation of the lambda grammar lexical entry proposed by Muskens (2001). Finally, formulas 4 and 5 are the translations of the two Lambek calculus formulas for peripheral extraction. These formulas are related as follows (where a directed path between the two formulas denotes derivability of the target from the source).



So the formulas of the Displacement calculus and the one proposed directly for first-order linear logic are identical (formula 2). This formula is equivalent to formula 3 proposed for λ-grammars; though we cannot always transform a linear logic formula into an equivalent prenex normal form (Lincoln and Shankar, 1994), formula 2 does allow such a form which is formula 3. When we look at lambda grammars in isolation, we cannot even directly ask the question about the relation to Lambek calculus formulas, though here it is clear that formulas 1 to 3 all have the Lambek calculus formulas 4 and 5 as special cases. The new formula 1 is the most general formula, but it is unclear whether or not there is any useful (or harmful!) difference in behaviour between this formula and the formulas corresponding to those use in the Displacement calculus and lambda grammars.

This brings up an important question: since, in (classical) first-order logic, formula 1 is equivalent to formulas 2 and 3 maybe first-order *linear* logic is too fine-grained a tool and the suitable notions of equivalence are better formulated directly in first-order logic. Is the difference between classical first-order equivalence and linear first-order equivalence important, and if, so which is the more suitable notion in the current context?

5.4 Adverbs, higher-order formulas and lambda grammars

The first-order linear logic perspective also clarifies the limitations of abstract categorial grammars/lambda grammars. For adverbs, for example, we are looking for a lexical entry which functions at least as well as the Lambek calculus formula $(np \setminus s) / (np \setminus s)$. However, as shown in (Moot, 2014b), we can simply enumerate all possible ACG lexical entries l, compute their translation into first-order linear logic, compute the translation of $(np \setminus s) / (np \setminus s)$ into first-order linear logic and compare. Keeping only the plausible lexical entries (that is, those which generate the right semantics and right word order) leaves us with three possibilities, which are shown as items 7 to 9 below, together with the translation of $(np \setminus s) / (np \setminus s)$ as item 6 (note the narrow scope of $\forall x_1$ in this translation). The adverb is assumed to span positions 1-2.

(6) $\forall x_0 \forall x_2.[\forall x_1.[np(x_1, 2) \multimap s(x_1, x_2)] \multimap (np(x_0, 1) \multimap s(x_0, x_2))]$

(7) $\forall x_0 \forall x_1 \forall x_2.[(np(x_1, x_1) \multimap s(2, x_2)) \multimap (np(x_0, 1) \multimap s(x_0, x_2))]$

(8) $\forall x_0 \forall x_1 \forall x_2.[(np(1, 2) \multimap s(x_1, x_2)) \multimap (np(x_0, x_1) \multimap s(x_0, x_2))]$

(9) $\forall x_0 \forall x_1 \forall x_2.[(np(x_1, 2) \multimap s(x_0, x_2)) \multimap (np(x_1, 1) \multimap s(x_0, x_2))]$

The translations of ACG lexical entries are always formulas with only universal quantifiers and in prenex normal form[10]. It is easy to verify that all of items 7 to 9 are strictly more general than the translation of $(np \setminus s) / (np \setminus s)$, shown as item 6.

However, where in the case of relative pronouns, a more general formula turned out to be a benefit, in the case of adverbs, it turns out to be a source of overgeneration. For example, item 7, the adverb lexical entry most commonly used in the ACG literature,

[10]The term Skolem normal form is often used for a prenex normal form with universal quantifiers, but I don't use it here since it suggests that existential quantifiers have been replaced by universally quantified Skolem terms and 1) there are no terms in the translations of ACG formulas 2) Skolemization is unsound in first-order linear logic. (Lincoln and Shankar, 1994)

predicts that an adverb selects to its right, a sentence missing a noun phrase *anywhere*. In other words, the lexical entry for adverbs is modelled after the lexical entry for relative pronouns and therefore follows a "medial extraction" analysis, whereas items 8 and 9 predict a type of quantifying-in behaviour: item 8 is modelled after the type assigned to a generalised quantifier but with an extra *np* argument; it takes as its argument a sentences missing a noun phrase at the position of the adverb (just like a generalised quantifier takes a sentence missing an *np* at the position of the quantifier as its argument), making the odd prediction that adverbs occur at the same place as noun phrases. Formulas 7 and 8 therefore predict Sentences (10) and (11), along with many other strange possibilities, are correct.

(10) John deliberately Mary hit.

(11) Mary the friend of deliberately left.

Other higher-order Lambek calculus formulas have similar problems when we try to translate them into ACG. For example, the word "and" when used for the coordination of transitive verbs has 3024 possible translations, with 420 generating the correct surface structure and 148 having, in addition, the correct semantics as a possible reading. However, these many possibilities all follow the same pattern we have seen above for adverbs: they use a combination of extraction-like and quantifying-in constructions and therefore overgenerate.

5.5 Reflexives in D and in first-order linear logic

There is a minor difference between the Displacement calculus and its translation into first-order linear logic. The object reflexive of Morrill et al. (2011).

$$((vp \uparrow_{>} np) \uparrow_{<} np) \downarrow_{<} (vp \uparrow_{>} np)$$

is translated as follows, when it occurs at string positions 3-4 (the *vp* subformulas have been left untranslated).

$$\forall x_0, x_1, x_2, x_5[np(3,4) \multimap np(x_1, x_2) \multimap \|vp\|^{x_0,x_5}] \multimap np(x_1, x_2) \multimap \|vp\|^{x_0,x_5}$$

The first-order linear logic formula more or less states that it transforms a ditransitive verb (with one of its objects occurring at position 3-4) into a transitive verb. However, the original D formula specifies something more, namely that the other *np* argument, $np(x_1, x_2)$ occurs before this first *np*, in other words that $x_2 \leq 3$. It is this property which is used to ensure the following two grammaticality judgments.

(12) Mary talked to John about himself.

(13) $*$ Mary talked about himself to John.

For formulas of the form $(A \multimap B \multimap C) \multimap D$, the Displacement calculus can encode the precedence between the A and B subformulas, whereas its translation in first-order linear logic cannot, in can only unify positions but has no mechanism for making statements like $x_2 \leq 3$[11].

[11] It would be possible to enrich first-order linear logic with order constraints on the variables, specifying things like $x_1 \leq x_2$. Though such an extension would be simple to implement, especially in a constraint-programming context, it seems to be rather powerful and it would complicate the proof theory, since such constraints would in all likelihood need to be non-linear.

Though I think the application of this property to reflexives is a clever solution, I am unsure if its unavailability in first-order linear logic is truly a handicap. The Displacement calculus refers to its "separators" by number in the linear order, whereas the lambda calculus-like formalisms give their "points for future insertion" a name (ie. a lambda calculus variable). Exploiting the core mechanism of the Displacement calculus to enforce constraints on linear order is a delicate undertaking since it risks interfering with the many other operations depending on these separators. For the treatment of Dutch verb clusters, Morrill et al. (2011, p. 28) are already aware that they sometimes need to refer to separators by mechanisms other than linear order.

5.6 Open questions

One important question must remain unanswered here: can we give (partial) translations from the Displacement calculus to Hybrid Type-Logical Grammars, or vice versa? We can, of course, translate into first-order linear logic and see if can translate it back to the other system (Moot, 2015, uses such a strategy, but to translate first-order linear logic proofs back to proofs in the "source language"). However, such a strategy depends on the exact formula obtained, for example the Formulas 2 and 3 on page 16 are equivalent and therefore this elementary case would already need some additional mechanism to cope with equivalence. A direct translation might side-step this problem and help crystallise the differences between the systems even more clearly.

There is another open question, which was already briefly alluded to when discussing unification. The variables and quantifiers of the formulas which interest us follow specific patterns. For example, each quantifier binds two occurrences of its eigenvariable (this is true for position variables such as those we have seen here, for agreement variables, such as case, grammatical gender etc., we may allow quantifiers which bind only one occurrence of their eigenvariable). Position variables further have one positive and one negative occurrence (where the switch from left position to right position is treated as a polarity change). Are there further patterns to discover here?

6 Towards convergence

Figure 6 gives a slightly simplified picture of the main logical calculi and their relations. The two meta-logics are not exclusive: the Displacement calculus can be embedded both in a graph rewriting calculus (as shown by (Valentín, 2014)) and in first-order linear logic (Moot, 2014a). Similarly, NL_λ seems to have a representation in both systems (Barker and Shan, 2014; Moot, 2014a,b), though the full formal details of this possibility still need to be worked out. However, the connection between a multimodal setup and a lambda calculus-like setup explored by Barker and Shan (2014, Chapter 17) seems an important step towards convergence of the meta-logics discussed in this paper.

It remains an open question whether there is a natural way to guarantee convergence of the two logics: what restriction on graph rewriting for proof nets would ensure that the calculus can be translated in first-order linear logic?

Similarly, it is unclear at the moment whether there is a connection between the Lambek-Grishin calculus (LG) and first-order linear logic. The multi-conclusioned nature of the Lambek-Grishin calculus makes it hard to compare it directly to its intuitionistic peers.

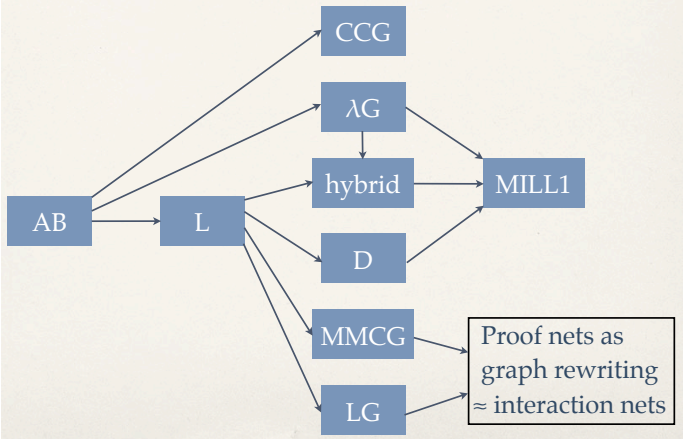


Figure 11: A (slightly simplified) representation of principal extensions and variants of the Lambek calculus (L). The Ajdukiewicz/Bar-Hillel calculus (AB), Combinatory Categorial Grammars (CCG), lambda grammars (λG), Hybrid Type-Logical Grammars (hybrid), the Displacement calculus (D), multimodal categorial grammars (MMCG) and the Lambek-Grishin (LG) calculus and the two meta-logics: first-order linear logic (MILL1) and the proof nets of Moot & Puite

7 Conclusions

In this paper I have given two tools which I hope will create bridges between the many formalisms in the family of type-logical grammars. These tools are not only theoretical tools, but they are also the basis of two implementations (Moot et al., 2015; Moot, 2015), a theorem prover based on (for the moment only the intuitionistic part of) the graph rewriting calculus and a theorem prover for first-order linear logic, which can output proofs for the Displacement calculus and Hybrid Type-Logical Grammars by translation.

I believe open discussion of the benefits and disadvantages of one system over another as well as a growing body of linguistic data which can find a satisfactory account in type-logical grammars will be an important factor in ensuring that our community stays vibrant and healthy.

Acknowledgements

I would like to thank Chris Barker, Philippe de Groote, Yusuke Kubota, Robert Levine and Michael Moortgat for their discussion on the topics which led to the publication of this paper.

I would also like the referees of the Empirical Advances in Categorial Grammar (2015) workshop for the remarks and comments on the submitted abstract.

References

Aarts, Erik and Kees Trautwein. 1995. Non-associative Lambek categorial grammar in polynomial time. *Mathematical Logic Quarterly* 41:476–484.

Andreoli, Jean-Marc. 1992. Logic programming with focussing proofs in linear logic. *Journal of Logic and Computation* 2(3).

Areces, Carlos, Raffaella Bernardi, and Michael Moortgat. 2001. Galois connections in categorial type logic. In G.-J. Kruijff, L. Moss, and R. T. Oehrle, eds., *Proceedings of FGMOL 2001*, vol. 53 of *Electronic Notes in Theoretical Computer Science*, 3–20. Elsevier.

Barker, Chris and Chung-Chieh Shan. 2014. *Continuations and Natural Language*. Oxford Studies in Theoretical Linguistics. Oxford University Press.

Bernardi, Raffaella and Michael Moortgat. 2010. Continuation semantics for the lambek–grishin calculus. *Information and Computation* 208(5):397–416.

Blackburn, Patrick, Maarten de Rijke, and Yde Venema. 2001. *Modal Logic*. New York, NY, USA: Cambridge University Press. ISBN 0-521-80200-8.

Buszkowski, Wojciech. 1997. Mathematical linguistics and proof theory. In J. van Benthem and A. ter Meulen, eds., *Handbook of Logic and Language*, chap. 12, 683–736. Amsterdam: North-Holland Elsevier.

Carpenter, Bob. 1998. *Type-logical Semantics*. Cambridge, Massachusetts: MIT Press.

Danos, Vincent. 1990. *La Logique Linéaire Appliquée à l'étude de Divers Processus de Normalisation (Principalement du λ-Calcul)*. Ph.D. thesis, University of Paris VII.

de Groote, Philippe. 1999. The non-associative Lambek calculus with product in polynomial time. In N. V. Murray, ed., *Automated Reasoning With Analytic Tableaux and Related Methods*, vol. 1617 of *Lecture Notes in Artificial Intelligence*, 128–139. Springer.

de Groote, Philippe. 2001. Towards abstract categorial grammars. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, 252–259. Association for Computational Linguistics.

de Groote, Philippe and Sylvain Pogodalla. 2004. On the expressive power of abstract categorial grammars: Representing context-free formalisms. *Journal of Logic, Language and Information* 13(4):421–438.

Engelfriet, Joost. 1997. Context-free graph grammars. In G. Rosenberg and A. Salomaa, eds., *Handbook of Formal Languages 3: Beyond Words*, 125–213. New York: Springer.

Girard, Jean-Yves. 1991. Quantifiers in linear logic II. In G. Corsi and G. Sambin, eds., *Nuovi problemi della logica e della filosofia della scienza*, vol. II. Bologna, Italy: CLUEB. Proceedings of the conference with the same name, Viareggio, Italy, January 1990.

Hendriks, Herman. 1993. *Studied Flexibility: Categories and Types in Syntax and Semantics*. Ph.D. thesis, ILLC, University of Amsterdam.

Hendriks, Petra. 1995. Ellipsis and multimodal categorial type logic. In G. Morrill and R. T. Oehrle, eds., *Proceedings of Formal Grammar 1995*, 107–122. Barcelona, Spain.

Jäger, Gerhard. 2001. Anaphora and quantification in categorial grammar. In M. Moortgat, ed., *Logical Aspects of Computational Linguistics*, vol. 2014 of *Lecture Notes in Computer Science*, 70–90. Springer.

Joshi, Aravind. 1985. Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In D. Dowty, L. Karttunen, and A. Zwicky, eds., *Natural Language Parsing*, 206–250. Cambridge University Press.

Knuth, Donald E. 2000. Dancing links. *arXiv preprint cs/0011047* .

Kubota, Yusuke and Robert Levine. 2012. Gapping as like-category coordination. In D. Béchet and A. Dikovsky, eds., *Logical Aspects of Computational Linguistics*, vol. 7351 of *Lecture Notes in Computer Science*, 135–150. Nantes: Springer.

Kubota, Yusuke and Robert Levine. 2013. Determiner gapping as higher-order discontinuous constituency. In G. Morrill and M.-J. Nederhof, eds., *Formal Grammar*, vol. 8036 of *Lecture Notes in Computer Science*, 225–241. Springer.

Kurtonina, Natasha and Michael Moortgat. 1997. Structural control. In P. Blackburn and M. de Rijke, eds., *Specifying Syntactic Structures*, 75–113. Stanford: CSLI.

Lafont, Yves. 1995. From proof nets to interaction nets. In J.-Y. Girard, Y. Lafont, and L. Regnier, eds., *Advances in Linear Logic*, 225–247. Cambridge University Press.

Lakatos, Imre. 1978. The problem of appraising scientific theories: Three approaches. In J. Worrall and G. Currie, eds., *Mathematics, Science and Epistemology*, vol. 2 of *Philosophical Papers*, chap. 6, 107–120. Cambridge University Press.

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

Lambek, Joachim. 1961. On the calculus of syntactic types. In R. Jakobson, ed., *Structure of Language and its Mathematical Aspects, Proceedings of the Symposia in Applied Mathematics*, vol. XII, 166–178. American Mathematical Society.

Lincoln, Patrick and Natarajan Shankar. 1994. Proof search in first-order linear logic and other cut-free sequent calculi. In *Proceedings of Logic in Computer Science (LICS'94)*, 282–291. IEEE Computer Society Press.

Martelli, Alberto and Ugo Montanari. 1982. An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4(2):258–282.

Moortgat, Michael. 1996a. Generalized quantifiers and discontinuous type constructors. In H. Bunt and A. van Horck, eds., *Discontinuous Constituency*, 181–207. Berlin: Mouton de Gruyter.

Moortgat, Michael. 1996b. In situ binding: A modal analysis. In P. Dekker and M. Stokhof, eds., *Proceedings 10th Amsterdam Colloquium*, 539–549. ILLC, Amsterdam.

Moortgat, Michael. 1996c. Multimodal linguistic inference. *Journal of Logic, Language and Information* 5(3–4):349–385.

Moortgat, Michael. 1997. Categorial type logics. In J. van Benthem and A. ter Meulen, eds., *Handbook of Logic and Language*, chap. 2, 93–177. Elsevier/MIT Press.

Moortgat, Michael and Richard T. Oehrle. 1994. Adjacency, dependency and order. In *Proceedings 9th Amsterdam Colloquium*, 447–466.

Moot, Richard. 2002. *Proof Nets for Linguistic Analysis*. Ph.D. thesis, Utrecht Institute of Linguistics OTS, Utrecht University.

Moot, Richard. 2007. Proof nets for display logic. Technical Report, CNRS and INRIA Futurs.

Moot, Richard. 2008. Lambek grammars and hyperedge replacement grammars. Technical Report, CNRS and Bordeaux University.

Moot, Richard. 2014a. Extended Lambek calculi and first-order linear logic. In C. Casadio, B. Coecke, M. Moortgat, and P. Scott, eds., *Categories and Types in Logic, Language, and Physics: Essays dedicated to Jim Lambek on the Occasion of this 90th Birthday*, No. 8222 in Lecture Notes in Artificial Intelligence, 297–330. Springer.

Moot, Richard. 2014b. Hybrid type-logical grammars, first-order linear logic and the descriptive inadequacy of lambda grammars. Technical Report, LaBRI (CNRS), Bordeaux University. Submitted to the IfCoLog Journal of Logic and Its Applications.

Moot, Richard. 2015. Linear one: A theorem prover for first-order linear logic. `https://github.com/RichardMoot/LinearOne`.

Moot, Richard and Mario Piazza. 2001. Linguistic applications of first order multiplicative linear logic. *Journal of Logic, Language and Information* 10(2):211–232.

Moot, Richard and Quintijn Puite. 2002. Proof nets for the multimodal Lambek calculus. *Studia Logica* 71(3):415–442.

Moot, Richard and Christian Retoré. 2012. *The Logic of Categorial Grammars: A Deductive Account of Natural Language Syntax and Semantics*. No. 6850 in Lecture Notes in Artificial Intelligence. Springer.

Moot, Richard, Xander Schrijen, Gert Jan Verhoog, and Michael Moortgat. 2015. Grail0: A theorem prover for multimodal categorial grammars. `https://github.com/RichardMoot/Grail0`.

Morrill, Glyn. 1994. *Type Logical Grammar*. Dordrecht: Kluwer Academic Publishers.

Morrill, Glyn and Oriol Valentín. 2014. Displacement logic and anaphora. *Journal of Computer and System Sciences* 80(2):390–409.

Morrill, Glyn and Oriol Valentín. 2015. Computational coverage of TLG: Displacement. In Y. Kubota and R. Levine, eds., *Empirical Advances in Categorial Grammar*. European Summer School in Logic, Language and Information.

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

Muskens, Reinhard. 2001. Categorial grammar and lexical-functional grammar. In *Proceedings of the LFG01 Conference*, 259–279. University of Hong Kong.

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

Oehrle, Richard T. 2011. Multi-modal type-logical grammar. In R. Borsley and K. Börjars, eds., *Non-transformational Syntax: Formal and Explicit Models of Grammar*, chap. 6, 225–267. Wiley-Blackwell.

Patterson, M. S. and M. N. Wegman. 1978. Linear unification. *Journal of Computing and Systems Sciences* 16:158–167.

Pentus, Mati. 1995. Lambek grammars are context free. In *Proceedings of the Eighth Annual IEEE Symposium on Logic in Computer Science*, 429–433. Montreal, Canada.

Pullum, Geoffrey K. and Gerald Gazdar. 1982. Natural languages and context-free languages. *Linguistics and Philosophy* 4(4):471–504.

Shieber, Stuart. 1985. Evidence against the context-freeness of natural language. *Linguistics & Philosophy* 8:333–343.

Steedman, Mark. 2001. *The Syntactic Process*. Cambridge, Massachusetts: MIT Press.

Valentín, Oriol. 2014. The hidden structural rules of the discontinuous Lambek calculus. In C. Casadio, B. Coecke, M. Moortgat, and P. Scott, eds., *Categories and Types in Logic, Language, and Physics: Essays dedicated to Jim Lambek on the Occasion of this 90th Birthday*, No. 8222 in Lecture Notes in Artificial Intelligence, 402–420. Springer.

van Benthem, Johan. 1987. Categorial grammar and lambda calculus. In D. Skordev, ed., *Mathematical Logic and Its Applications*, 39–60. New York: Plenum Press.

Computational Coverage of TLG: Displacement*

Glyn Morrill
Universitat Politècnica de Catalunya
Barcelona

Oriol Valentín
Universitat Politècnica de Catalunya
Barcelona

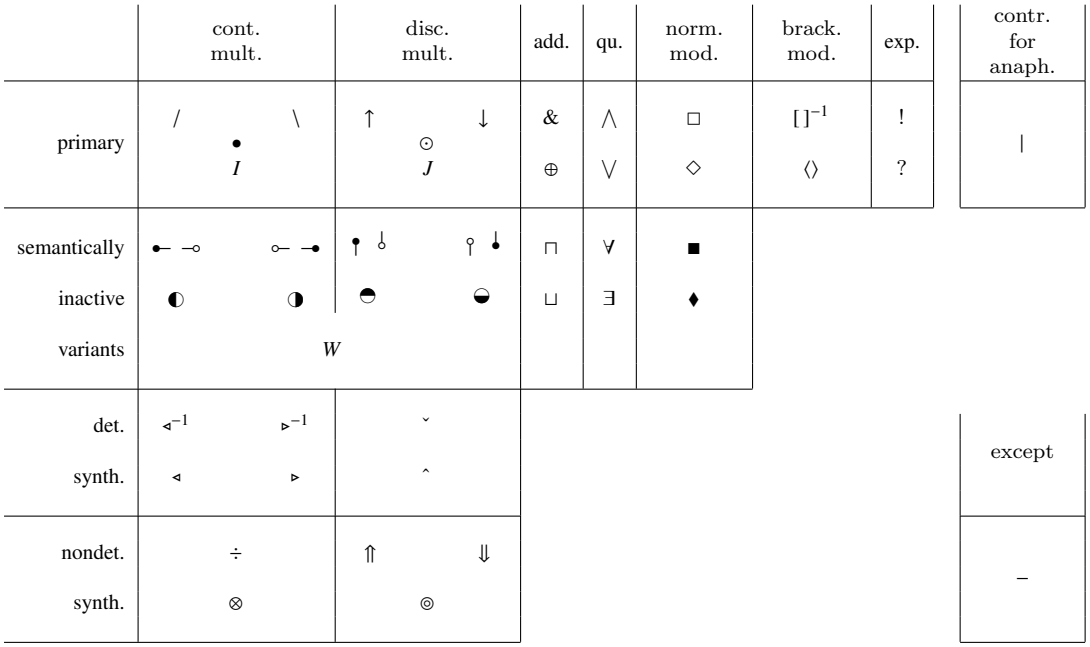
Abstract

This paper reports on the coverage of TLG of Morrill (1994) and Moortgat (1997), and on how it has been computer implemented. We computer-analyse examples of displacement: discontinuous idioms, quantification, (medial) relativisation, VP ellipsis, (medial) pied piping, appositive relativisation, parentheticals, gapping, comparative subdeletion, and reflexivisation, and, in the appendix, Dutch verb raising and cross-serial dependency.

Keywords: logical syntax and semantics; parsing as deduction; displacement

1. Introduction

The version of the formalism used is essentially the categorial type logic of Morrill (2014) plus Morrill and Valentín (2014b), and comprises 50 connectives as shown:



*Research partially supported by an ICREA Acadèmia to the first author, and, for both authors, by BASMATI MICINN project (TIN2011-27479-C04-03) and grant 2014SGR 890 (MACDA) from AGAUR, Generalitat de Catalunya. We thank two anonymous referees for comments and suggestions.

The heart of the logic is the displacement calculus of Morrill and Valentín (2010) and Morrill et al. (2011) which comprises twin continuous and discontinuous residuated families of connectives having a pure sequent calculus, the tree-based hypersequent calculus, and enjoying Cut-elimination (Valentín, 2012). Other primary connectives are additives, 1st order quantifiers, normal (i.e. distributive) modalities, bracket (i.e. nondistributive) modalities, and the non-linear exponentials, and contraction for anaphora.

We can draw a clear distinction between these primary connectives and the semantically inactive connectives and defined connectives which are abbreviatory and therefore merely for convenience.

There are semantically inactive variants of the continuous and discontinuous multiplicatives, including the words as types predicate W, and semantically inactive variants of the additives, 1st order quantifiers, and normal modalities.

Defined connectives divide into the continuous deterministic (unary) synthetic connectives of projection and injection, and the discontinuous, split and bridge, and the continuous nondeterministic (binary) synthetic connectives of nondirectional division and unordered product, and the discontinuous, nondeterministic extract, infix, and discontinuous product.

Finally there is the negation as failure of 'except' (formerly difference), a powerful device for expressing linguistic exceptions (Morrill and Valentín, 2014a).

2. Rules and linguistic applications for primary connectives

1. $$\frac{\Gamma \Rightarrow B{:}\,\psi \qquad \Delta\langle \overrightarrow{C}{:}\,z\rangle \Rightarrow D{:}\,\omega}{\Delta\langle \overrightarrow{C/B}{:}\,x, \Gamma\rangle \Rightarrow D{:}\,\omega\{(x\ \psi)/z\}}\,/L \qquad \frac{\Gamma, \overrightarrow{B}{:}\,y \Rightarrow C{:}\,\chi}{\Gamma \Rightarrow C/B{:}\,\lambda y\chi}\,/R$$

2. $$\frac{\Gamma \Rightarrow A{:}\,\phi \qquad \Delta\langle \overrightarrow{C}{:}\,z\rangle \Rightarrow D{:}\,\omega}{\Delta\langle \Gamma, \overrightarrow{A\backslash C}{:}\,y\rangle \Rightarrow D{:}\,\omega\{(y\ \phi)/z\}}\,\backslash L \qquad \frac{\overrightarrow{A}{:}\,x, \Gamma \Rightarrow C{:}\,\chi}{\Gamma \Rightarrow A\backslash C{:}\,\lambda x\chi}\,\backslash R$$

3. $$\frac{\Delta\langle \overrightarrow{A}{:}\,x, \overrightarrow{B}{:}\,y\rangle \Rightarrow D{:}\,\omega}{\Delta\langle \overrightarrow{A\bullet B}{:}\,z\rangle \Rightarrow D{:}\,\omega\{\pi_1 z/x, \pi_2 z/y\}}\,\bullet L \qquad \frac{\Gamma_1 \Rightarrow A{:}\,\phi \qquad \Gamma_2 \Rightarrow B{:}\,\psi}{\Gamma_1, \Gamma_2 \Rightarrow A\bullet B{:}\,(\phi, \psi)}\,\bullet R$$

4. $$\frac{\Delta\langle \Lambda\rangle \Rightarrow A{:}\,\phi}{\Delta\langle \overrightarrow{I}{:}\,x\rangle \Rightarrow A{:}\,\phi}\,IL \qquad \frac{}{\Lambda \Rightarrow I{:}\,0}\,IR$$

Figure 1: Continuous multiplicatives

The continuous multiplicatives of Figure 1, the Lambek connectives, are the basic means of categorial categorization and subcategorization. The directional divisions over, /, and under, \, are exemplified by assignments such as **the**: N/CN for **the man**: N and **sings**: $N\backslash S$ for **John sings**: S, and **loves**: $(N\backslash S)/N$ for **John loves Mary**: S. The continuous product $\bullet$ is exemplified by a 'small clause' assignment such as **considers**: $(N\backslash S)/(N\bullet(CN/CN))$ for **John considers Mary socialist**: S.[1] The continuous unit can be used together with additive disjunction to express the optionality of a complement as in **eats**: $(N\backslash S)/(N\oplus I)$

[1]But this makes no different empirical predictions from the more standard type of analysis in CG and G/HPSG which simply treats verbs like consider as taking a noun phrase and an infinitive.

for **John eats fish**: S and **John eats**: S.[2] It can also be used in conjunction with the connective 'except' to prevent the null string being supplied as argument to an intensifier as in **very**: $(CN/CN)/((CN/CN)-I)$ for **very tall man**: CN but $*$**very man**: CN.

5. $$\frac{\Gamma \Rightarrow B\colon\psi \qquad \Delta\langle\overrightarrow{C}\colon z\rangle \Rightarrow D\colon\omega}{\Delta\langle\overrightarrow{C\uparrow_k B}\colon x\,|_k\,\Gamma\rangle \Rightarrow D\colon\omega\{(x\ \psi)/z\}}\uparrow_k L \qquad \frac{\Gamma\,|_k\,\overrightarrow{B}\colon y \Rightarrow C\colon\chi}{\Gamma \Rightarrow C\uparrow_k B\colon\lambda y\chi}\uparrow_k R$$

6. $$\frac{\Gamma \Rightarrow A\colon\phi \qquad \Delta\langle\overrightarrow{C}\colon z\rangle \Rightarrow D\colon\omega}{\Delta\langle\Gamma\,|_k\,\overrightarrow{A\downarrow_k C}\colon y\rangle \Rightarrow D\colon\omega\{(y\ \phi)/z\}}\downarrow_k L \qquad \frac{\overrightarrow{A}\colon x\,|_k\,\Gamma \Rightarrow C\colon\chi}{\Gamma \Rightarrow A\downarrow_k C\colon\lambda x\chi}\downarrow_k R$$

7. $$\frac{\Delta\langle\overrightarrow{A}\colon x\,|_k\,\overrightarrow{B}\colon y\rangle \Rightarrow D\colon\omega}{\Delta\langle\overrightarrow{A\odot_k B}\colon z\rangle \Rightarrow D\colon\omega\{\pi_1 z/x,\pi_2 z/y\}}\odot_k L \qquad \frac{\Gamma_1 \Rightarrow A\colon\phi \qquad \Gamma_2 \Rightarrow B\colon\psi}{\Gamma_1\,|_k\,\Gamma_2 \Rightarrow A\odot_k B}\odot_k R$$

8. $$\frac{\Delta\langle 1\rangle \Rightarrow A\colon\phi}{\Delta\langle\overrightarrow{J}\colon x\rangle \Rightarrow A\colon\phi}JL \qquad \frac{}{1 \Rightarrow J\colon 0}JR$$

Figure 2: Discontinuous multiplicatives

The discontinuous multiplicatives of Figure 2, the displacement connectives, are defined in relation to intercalation. When the value of the k subscript is 1 it may be omitted. Circumfixation, $\uparrow$, is exemplified by a discontinuous idiom assignment **gives**+1+**the**+**cold**+**shoulder**: $(N\backslash S)\uparrow N$ for **Mary gives John the cold shoulder**: S, and infixation, $\downarrow$, and circumfixation together are exemplified by a quantifier phrase assignment **everyone**: $(S\uparrow N)\downarrow S$ simulating Montague's S14 treatment of quantifying in. Circumfixation and discontinuous product, $\odot$, are illustrated in an assignment to a relative pronoun **that**: $(CN\backslash CN)/((S\uparrow N)\odot I)$ allowing both peripheral and medial extraction: **that John likes**: $CN\backslash CN$ and **that John saw today**: $CN\backslash CN$. Use of the discontinuous product unit, J, in conjunction with except is illustrated in a pronoun assignment **him**: $(((S\uparrow N)\uparrow_2 N)-(J\bullet((N\backslash S)\uparrow N)))\downarrow_2 (S\uparrow N)$ preventing a subject antecedent (Principle B effect).

The additives of Figure 3 have application to polymorphism. For example the additive conjunction & can be used for **rice**: $N\&CN$ as in **rice grows**: S and **the rice grows**: S,[3] and the additive disjunction $\oplus$ can be used for **is**: $(N\backslash S)/(N\oplus(CN/CN))$ as in **Bond is 007**: S and **Bond is teetotal**: S.

The quantifiers of Figure 4 have application to features. For example, singular and plural number in **sheep**: $\bigwedge nCNn$ for **the sheep grazes**: S and **the sheep graze**: S. And for a past, present or future tense finite sentence complement: **said**: $(N\backslash S)/\bigvee tS\,f(t)$ in **John said Mary walked**: S, **John said Mary walks**: S and **John said Mary will walk**: S.

With respect to the normal modalities of Figure 5, the universal has application to intensionality. For example, for a propositional attitude verb **believes**: $\Box((N\backslash S)/\Box S)$ with a modality outermost since the word has a sense, and its sentential complement is an intensional domain, but its subject is not.

The bracket modalities of Figure 6 have application to syntactical domains such as

[2]Note the advantage of this over simply listing intranstive and transitive lexical entries: empirically this latter does not capture the generalisation that in both cases the verb eats combines with a subject to the left, and computationally every lexical ambiguity doubles the lexical insertion search space. Appeal to lexical ambiguity is always available and never interesting, except where there is true ambiguity.

[3]Note the advantage of this approach over assuming an empty determiner: computationally it is not forbidden that there be any number of empty operators in any positions.

9. $$\frac{\Gamma\langle\overrightarrow{A}:x\rangle \Rightarrow C:\chi}{\Gamma\langle\overrightarrow{A\&B}:z\rangle \Rightarrow C:\chi\{\pi_1 z/x\}}\,\&L_1 \qquad \frac{\Gamma\langle\overrightarrow{B}:y\rangle \Rightarrow C:\chi}{\Gamma\langle\overrightarrow{A\&B}:z\rangle \Rightarrow C:\chi\{\pi_2 z/y\}}\,\&L_2$$

$$\frac{\Gamma \Rightarrow A:\phi \qquad \Gamma \Rightarrow B:\psi}{\Gamma \Rightarrow A\&B:(\phi,\psi)}\,\&R$$

10. $$\frac{\Gamma\langle\overrightarrow{A}:x\rangle \Rightarrow C:\chi_1 \qquad \Gamma\langle\overrightarrow{B}:y\rangle \Rightarrow C:\chi_2}{\Gamma\langle\overrightarrow{A\oplus B}:z\rangle \Rightarrow C:z \to x.\chi_1; y.\chi_2}\,\oplus L$$

$$\frac{\Gamma \Rightarrow A:\phi}{\Gamma \Rightarrow A\oplus B:\iota_1\phi}\,\oplus R_1 \qquad \frac{\Gamma \Rightarrow B:\psi}{\Gamma \Rightarrow A\oplus B:\iota_2\psi}\,\oplus R_2$$

Figure 3: Additives

11. $$\frac{\Gamma\langle\overrightarrow{A[t/v]}:x\rangle \Rightarrow B:\psi}{\Gamma\langle\overrightarrow{\bigwedge vA}:z\rangle \Rightarrow B:\psi\{(z\ t)/x\}}\,\bigwedge L \qquad \frac{\Gamma \Rightarrow A[a/v]:\phi}{\Gamma \Rightarrow \bigwedge vA:\lambda v\phi}\,\bigwedge R^{\dagger}$$

12. $$\frac{\Gamma\langle\overrightarrow{A[a/v]}:x\rangle \Rightarrow B:\psi}{\Gamma\langle\overrightarrow{\bigvee vA}:z\rangle \Rightarrow B:\psi\{\pi_2 z/x\}}\,\bigvee L^{\dagger} \qquad \frac{\Gamma \Rightarrow A[t/v]:\phi}{\Gamma \Rightarrow \bigvee vA:(t,\phi)}\,\bigvee R$$

Figure 4: Quantifiers, where † indicates that there is no a in the conclusion

prosodic phrases and extraction islands. For example, **walks**: $\langle\rangle N\backslash S$ for the subject condition, and **before**: $[\,]^{-1}(VP\backslash VP)/VP$ for the adverbial island constraint.

Finally, there are non-linear connectives. The exponentials of Figure 7 have application to sharing. Using the universal exponential, !, for which contraction induces island brackets, we can assign a relative pronoun type **that**: $(CN\backslash CN)/(S/!N)$ allowing parasitic extraction such as **paper that John filed without reading**: CN, where parasitic gaps can appear only in (weak) islands, but can be iterated in subislands, for example, **man who the fact that the friends of admire without praising surprises**.[4]

Using the existential exponential, ?, we can assign a coordinator type **and**: $(?N\backslash N)/N$ allowing iterated coordination as in **John, Bill, Mary and Suzy**: N, or **and**: $(?(S/N)\backslash(S/N))/(S/N)$ for **John likes, Mary dislikes, and Bill hates, London** (iterated right node raising), and so on.

The limited contraction for anaphora, |, of Figure 8 also has application to sharing; it can be used for anaphora in an assignment like **it**: $(S\uparrow N)\downarrow(S|N)$ for, e.g., the company$_i$ said it$_i$ **flourished**: S, and it can be used for **such that** relativisation in an assignment **such that**: $(CN\backslash CN)/(S|N)$ for, say, **man such that$_i$ he$_i$ thinks Mary loves him$_i$**: CN.

[4]Morrill (2011b), Chapter 5. In the case that island violations are grammatical, as they are under certain conditions, we assume that the relative pronoun type is not $(CN\backslash CN)/(S/!N)$ but $(CN\backslash CN)/(S/\bigcirc N)$ where $\bigcirc$ is an association and commutation structural modality (Morrill, 1994), Chapter 7. This explains how island violation is possible combinatorially but we leave unanswered the question of how the choice of the relative pronoun type is conditioned by processing factors.

13. $$\frac{\Gamma\langle\overrightarrow{A}:x\rangle \Rightarrow B:\psi}{\Gamma\langle\overrightarrow{\Box A}:z\rangle \Rightarrow B:\psi\{^{\vee}z/x\}}\,\Box L \qquad \frac{\Box/\blacksquare\Gamma \Rightarrow A:\phi}{\Box/\blacksquare\Gamma \Rightarrow \Box A:{}^{\wedge}\phi}\,\Box R$$

14. $$\frac{\Box/\blacksquare\Gamma\langle\overrightarrow{A}:x\rangle \Rightarrow \Diamond/\blacklozenge B:\psi}{\Box\Gamma\langle\overrightarrow{\Diamond A}:z\rangle \Rightarrow \Diamond/\blacklozenge B:\psi\{^{\cup}z/x\}}\,\Diamond L \qquad \frac{\Gamma \Rightarrow A:\phi}{\Gamma \Rightarrow \Diamond A:{}^{\cap}\phi}\,\Diamond R$$

Figure 5: Normal modalities, where $\Box/\blacksquare\Gamma$ signifies a structure all the types of which have main connective $\Box$ or $\blacksquare$

15. $$\frac{\Delta\langle\overrightarrow{A}:x\rangle \Rightarrow B:\psi}{\Delta\langle\overrightarrow{[\,]^{-1}A}:x]\rangle \Rightarrow B:\psi}\,[\,]^{-1}L \qquad \frac{[\Gamma] \Rightarrow A:\phi}{\Gamma \Rightarrow [\,]^{-1}A:\phi}\,[\,]^{-1}R$$

16. $$\frac{\Delta\langle[\overrightarrow{A}:x]\rangle \Rightarrow B:\psi}{\Delta\langle\overrightarrow{\langle\rangle A}:x\rangle \Rightarrow B:\psi}\,\langle\rangle L \qquad \frac{\Gamma \Rightarrow A:\phi}{[\Gamma] \Rightarrow \langle\rangle A:\phi}\,\langle\rangle R$$

Figure 6: Bracket modalites

3. Implementation

A computational lexicon and parser integrates the grammatical features of the previous section, and of the remaining connectives, which defines a fragment including:

- the PTQ examples of Dowty et al. (1981), Chapter 7;
- the discontinuity examples of Morrill et al. (2011);
- relativisation, including islands and parasitic gaps;
- constituent coordination, non-constituent coordination, coordination of 'unlike' types, ATBE, and a unitary lexical type analyses of simplex and complex gapping.

The implementation is CatLog2, a categorial parser/theorem-prover comprising 6000 lines of Prolog using backward chaining proof-search in the tree-based hypersequent calculus (Morrill, 2011a), and the focusing of Andreoli (Andreoli, 1992) rather than normalisation as in CatLog (Morrill, 2012). In addition to focusing, the implementation exploits the count-invariance of van Benthem (1991) and Valentín et al. (2013). In this paper we present the second item in the list above.

4. Displacement examples

In this section we analyse the displacement examples of the article Morrill et al. (2011) presenting the displacement calculus.[5] The first example, however, is modified in view of Morrill and Valentín (2014b). It is a discontinuous idiom (we include the indexation of CatLog, which contains the numeration of the source, within the example displays):

(1) (tdc(43)) **[mary]+gave+the+man+the+cold+shoulder** : $S\,f$

[5]Note how in the input to CatLog brackets mark islands: single brackets for weak islands such as subjects and double brackets for strong islands such as relative clauses and coordinate structures (Morrill, 2011b), Chapter 5.

17. $$\frac{\Gamma\langle A\colon x\rangle \Rightarrow B\colon \psi}{\Gamma\langle !A\colon x\rangle \Rightarrow B\colon \psi}\,!L \qquad \frac{!A_1\colon x_1,\ldots,!A_n\colon x_n \Rightarrow A\colon \phi}{!A_1\colon x_1,\ldots,!A_n\colon x_n \Rightarrow !A\colon \phi}\,!R$$

$$\frac{\Delta\langle !A\colon x,\Gamma\rangle \Rightarrow B\colon \psi}{\Delta\langle \Gamma,!A\colon x\rangle \Rightarrow B\colon \psi}\,!P \qquad \frac{\Delta\langle \Gamma,!A\colon x\rangle \Rightarrow B\colon \psi}{\Delta\langle !A\colon x,\Gamma\rangle \Rightarrow B\colon \psi}\,!P$$

$$\frac{\Delta\langle !A_0\colon x_0,\ldots,!A_n\colon x_n,[!A_0\colon y_0,\ldots,!A_n\colon y_0,\Gamma]\rangle \Rightarrow B\colon \psi}{\Delta\langle !A_0\colon x_0,\ldots,!A_n\colon x_n,\Gamma\rangle \Rightarrow B\colon \psi\{x_0/y_0,\ldots,x_n/y_n\}}\,!C$$

18. $$\frac{\Gamma \Rightarrow A\colon \phi}{\Gamma \Rightarrow ?A\colon [\phi]}\,?R \qquad \frac{\Gamma \Rightarrow A\colon \phi \qquad \Delta \Rightarrow ?A\colon \psi}{\Gamma,\Delta \Rightarrow ?A\colon [\phi|\psi]}\,?E$$

Figure 7: Exponentials

19. $$\frac{\Gamma \Rightarrow A\colon \phi \qquad \Delta\langle \overrightarrow{A}\colon x;\overrightarrow{B}\colon y\rangle \Rightarrow D\colon \omega}{\Delta\langle \Gamma;\overrightarrow{B|A}\colon z\rangle \Rightarrow D\colon \omega\{\phi/x,(z\ \phi)/y\}}\,|L$$

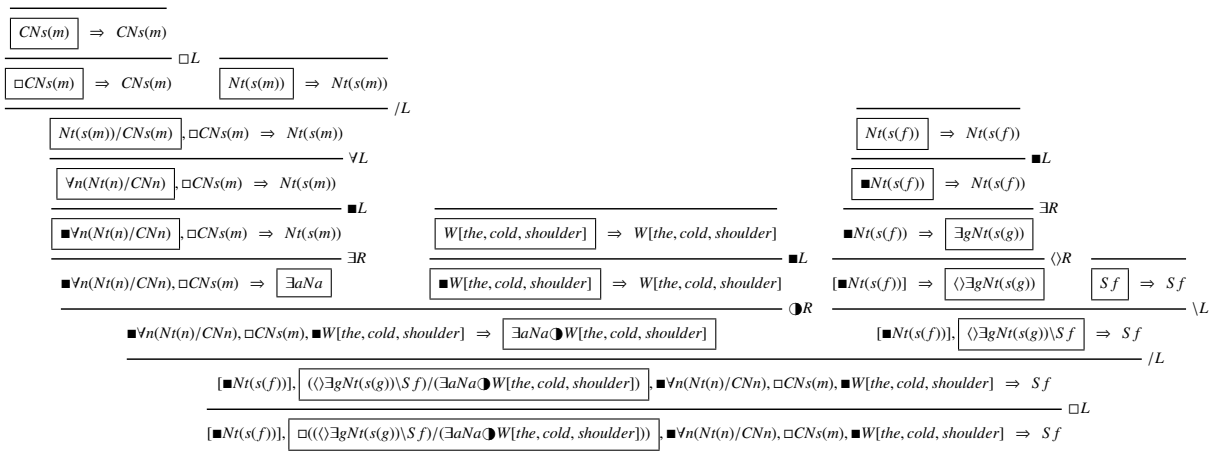
$$\frac{\Gamma\langle \overrightarrow{B_0}\colon y_0;\ldots;\overrightarrow{B_n}\colon y_n\rangle \Rightarrow D\colon \omega}{\Gamma\langle \overrightarrow{B_0|A}\colon z_0;\ldots;\overrightarrow{B_n|A}\colon z_n\rangle \Rightarrow D|A\colon \lambda x\omega\{(z_0\ x)/y_0,\ldots,(z_n\ x)/y_n\}}\,|R$$

Figure 8: Limited contraction for anaphora

Lexical lookup yields:

(2) $[\blacksquare Nt(s(f)) : m], \square((\langle\rangle \exists g Nt(s(g)) \backslash S\, f)/(\exists aNa \blacklozenge W[the, cold, shoulder])) :$
$\hat{}\lambda A \lambda B(Past\ ((\check{}shun\ A)\ B)), \blacksquare \forall n(Nt(n)/CNn) : \iota, \square CNs(m) : man,$
$\blacksquare W[the, cold, shoulder] : 0 \ \Rightarrow\ S\, f$

There is the derivation:



This delivers semantics:

(3) $(Past\ ((\check{}shun\ (\iota\ \check{}man))\ m))$

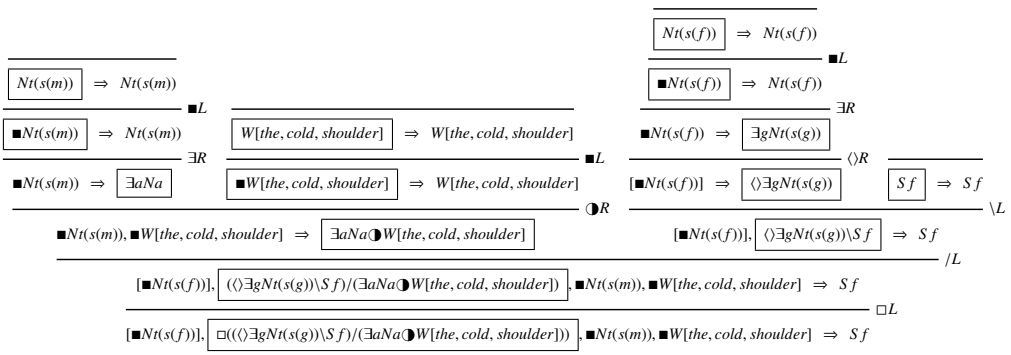
Similarly:

(4) (tdc(4343)) **[mary]+gave+john+the+cold+shoulder** : $S\,f$

Lexical lookup yields:

(5) $[\blacksquare Nt(s(f)) : m], \square((\langle\rangle\exists gNt(s(g))\backslash S\,f)/(\exists aNa\blacklozenge W[the, cold, shoulder])) :$
$\hat{}\lambda A\lambda B(Past\ ((\check{}shun\ A)\ B)), \blacksquare Nt(s(m)) : j, \blacksquare W[the, cold, shoulder] : 0\ \Rightarrow\ S\,f$

There is the derivation:



This delivers semantics:

(6) $(Past\ ((\check{}shun\ j)\ m))$

The next example has medial quantification:

(7) (tdc(47)) **[john]+gave+every+book+to+mary** : $S\,f$

Lexical lookup yields:

(8) $[\blacksquare Nt(s(m)) : j], \square((\langle\rangle\exists aNa\backslash S\,f)/(\exists bNb\bullet PPto)) : \hat{}\lambda A\lambda B(Past\ (((\check{}give\ \pi_2 A)\ \pi_1 A)\ B)),$
$\blacksquare\forall g(\forall f((S\,f\uparrow Nt(s(g)))\downarrow S\,f)/CNs(g)) : \lambda C\lambda D\forall E[(C\ E) \rightarrow (D\ E)], \square CNs(n) : book,$
$\blacksquare((PPto/\exists aNa)\sqcap\forall n((\langle\rangle Nn\backslash S\,i)/(\langle\rangle Nn\backslash S\,b))) : \lambda FF, \blacksquare Nt(s(f)) : m\ \Rightarrow\ S\,f$

There is the derivation:

This delivers semantics:

(9) $\forall C[(\check{}book\ C) \rightarrow (Past\ (((\check{}give\ m)\ C)\ j))]$

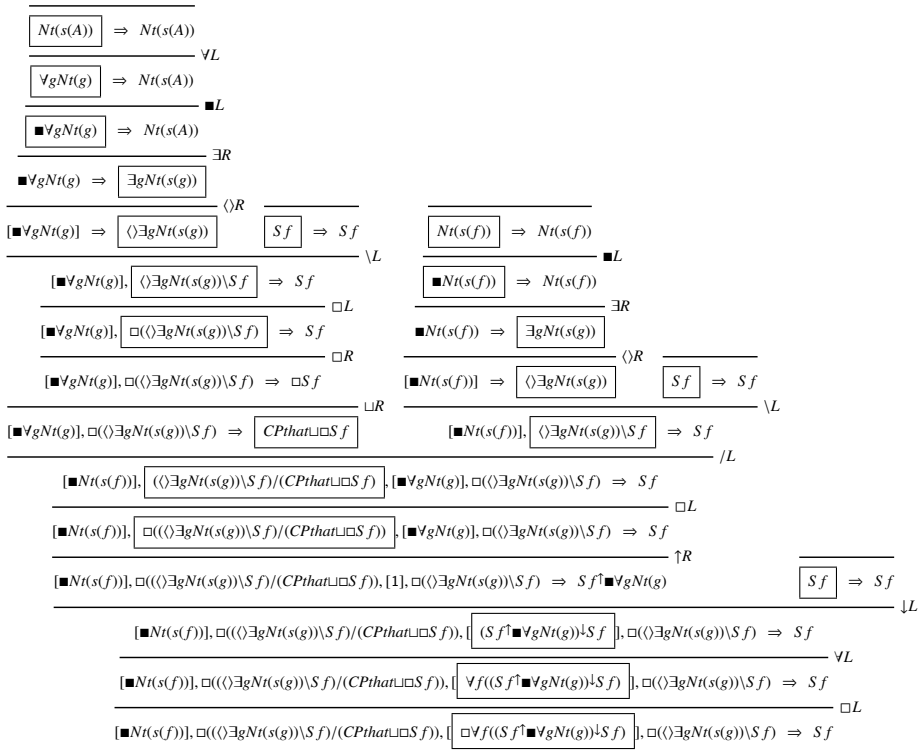
The following example has subordinate clause existential quantification, exhibiting de re/de dicto ambiguity:

(10) (tdc(50)) **[mary]+thinks+[someone]+left** : $S\,f$

Lexical lookup yields:

(11) $[\blacksquare Nt(s(f)) : m], \Box((\langle\rangle \exists gNt(s(g))\backslash S\, f)/(CPthat \sqcup \Box S\, f)) : \hat{}\lambda A \lambda B(Pres\ ((\check{}\, think\ A)\ B)),$
$[\Box \forall f((S\, f\uparrow \blacksquare \forall gNt(g))\downarrow S\, f) : \hat{}\lambda C \exists D[(\check{}\, person\ D) \wedge (C\ D)]], \Box(\langle\rangle \exists gNt(s(g))\backslash S\, f) :$
$\hat{}\lambda E(Pres\ (\check{}\, leave\ E)) \Rightarrow S\, f$

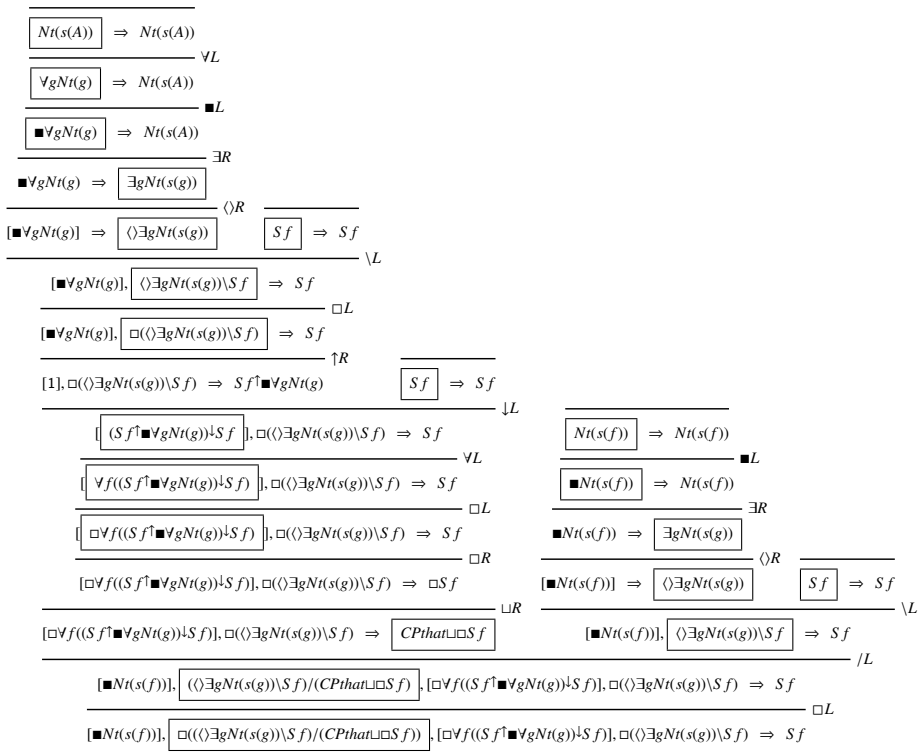
There is the de re derivation:



This delivers semantics:

(12) $\exists B[(\check{}\, person\ B) \wedge (Pres\ ((\check{}\, think\ \hat{}(Pres\ (\check{}\, leave\ B)))\ m))]$

And the de dicto derivation:



This delivers semantics:

(13) (*Pres* ((ˇ*think* ˆ∃*D*[(ˇ*person D*) ∧ (*Pres* (ˇ*leave D*))]) *m*))

The next example exhibits classical quantifier scope ambiguity:

(14) (tdc(53)) **[everyone]+loves+someone** : *S f*

There is the subject wide scope reading (cf. everyone loves their (respective) mother) and the object wide scope reading (cf. everyone loves (one and) the (same) queen). Lexical lookup yields:

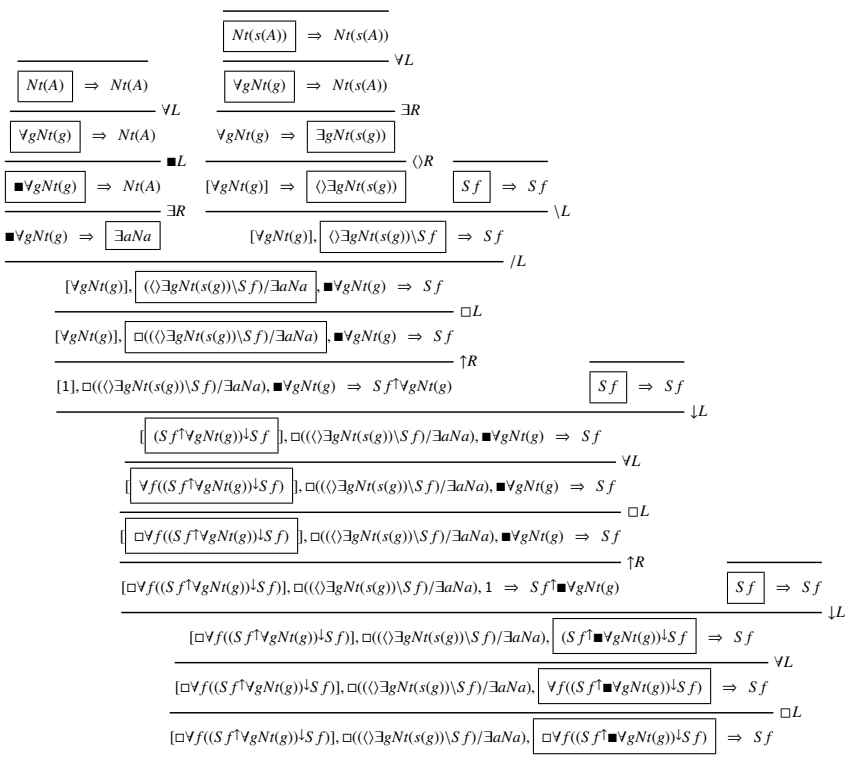
(15) [□∀*f*((*S f*↑∀*gNt*(*g*))↓*S f*) : ˆ*λA*∀*B*[(ˇ*person B*) → (*A B*)]], □((⟨⟩∃*gNt*(*s*(*g*))*S f*)/∃*aNa*) : ˆ*λC*λ*D*(*Pres* ((ˇ*love C*) *D*)), □∀*f*((*S f*↑■∀*gNt*(*g*))↓*S f*) : ˆ*λE*∃*F*[(ˇ*person F*) ∧ (*E F*)] ⇒ *S f*

There is the subject wide scope derivation as follows in which the subject quantifier is processed closest to the root:

This delivers semantics:

(16) $\forall B[(\check{}\mathit{person}\ B) \rightarrow \exists E[(\check{}\mathit{person}\ E) \wedge (\mathit{Pres}\ ((\check{}\mathit{love}\ E)\ B))]]$

And there is the object wide scope derivation as follows in which the object quantifier is processed closest to the root:



This delivers the semantics:

(17) $\exists B[(\check{}person\ B) \wedge \forall E[(\check{}person\ E) \rightarrow (Pres\ ((\check{}love\ B)\ E))]]$

The next example is of medial relativisation:

(18) (tdc(54)) **dog**+[[**that**+[**mary**]+**saw**+**today**]] : $CNs(n)$

But it is not analysed as in Morrill et al. (2011) but as in Morrill (2011b), Chapter 5. Note the double brackets for the strong island relative clause. The lexical lookup yields:

(19) $\Box CNs(n) : dog, [[\blacksquare\forall n([]^{-1}[]^{-1}(CNn\backslash CNn)/\blacksquare((\langle\rangle Nt(n) \sqcap !\blacksquare Nt(n))\backslash S\,f)) :$
$\lambda A\lambda B\lambda C[(B\ C) \wedge (A\ C)], [\blacksquare Nt(s(f)) : m], \Box((\langle\rangle\exists aNa\backslash S\,f)/(\exists aNa \oplus CPthat)) :$
$\hat{}\lambda D\lambda E(Past\ ((D \rightarrow F.(\check{}seee\ F); G.(\check{}seet\ G))\ E)), \Box\forall a\forall f((\langle\rangle Na\backslash S\,f)\backslash(\langle\rangle Na\backslash S\,f)) :$
$\hat{}\lambda H\lambda I(\check{}today\ (H\ I))]] \Rightarrow CNs(n)$

There is the derivation in Figure 9. This delivers semantics:

(20) $\lambda C[(\check{}dog\ C) \wedge (\check{}today\ (Past\ ((\check{}seee\ C)\ m)))]$

The next example is of VP ellipsis:

(21) (tdc(58a)) [**john**]+**slept**+**before**+[**mary**]+**did** : $S\,f$

(22) $[\blacksquare Nt(s(m)) : j], \Box(\langle\rangle\exists gNt(s(g))\backslash S\,f) : \hat{}\lambda A(Past\ (\check{}sleep\ A)), \blacksquare(\forall a\forall f((\langle\rangle Na\backslash S\,f)\backslash(\langle\rangle Na\backslash S\,f))/S\,f) :$
$\lambda B\lambda C\lambda D((before\ B)\ (C\ D)), [\blacksquare Nt(s(f)) : m], \blacksquare\forall a\forall g\forall b\forall h(((((\langle\rangle Na\backslash S\,g)\uparrow(\langle\rangle Nb\backslash S\,h))/(\exists c\langle\rangle Nc\backslash S\,f))\backslash$
$(((\langle\rangle Na\backslash S\,g)\uparrow(\langle\rangle Nb\backslash S\,h))) : \lambda E\lambda F((E\ F)\ F) \Rightarrow S\,f$

There is the derivation in Figure 10. This delivers the semantics:

(23) $((before\ (Past\ (\check{}sleep\ m)))\ (Past\ (\check{}sleep\ j)))$

In tdc(64) there is medial pied-piping:

(24) (tdc(64)) **mountain**+[[**the**+**painting**+**of**+**which**+**by**+**cezanne**+[**john**]+**sold**+
for+**tenmilliondollars**]] : $CNs(n)$

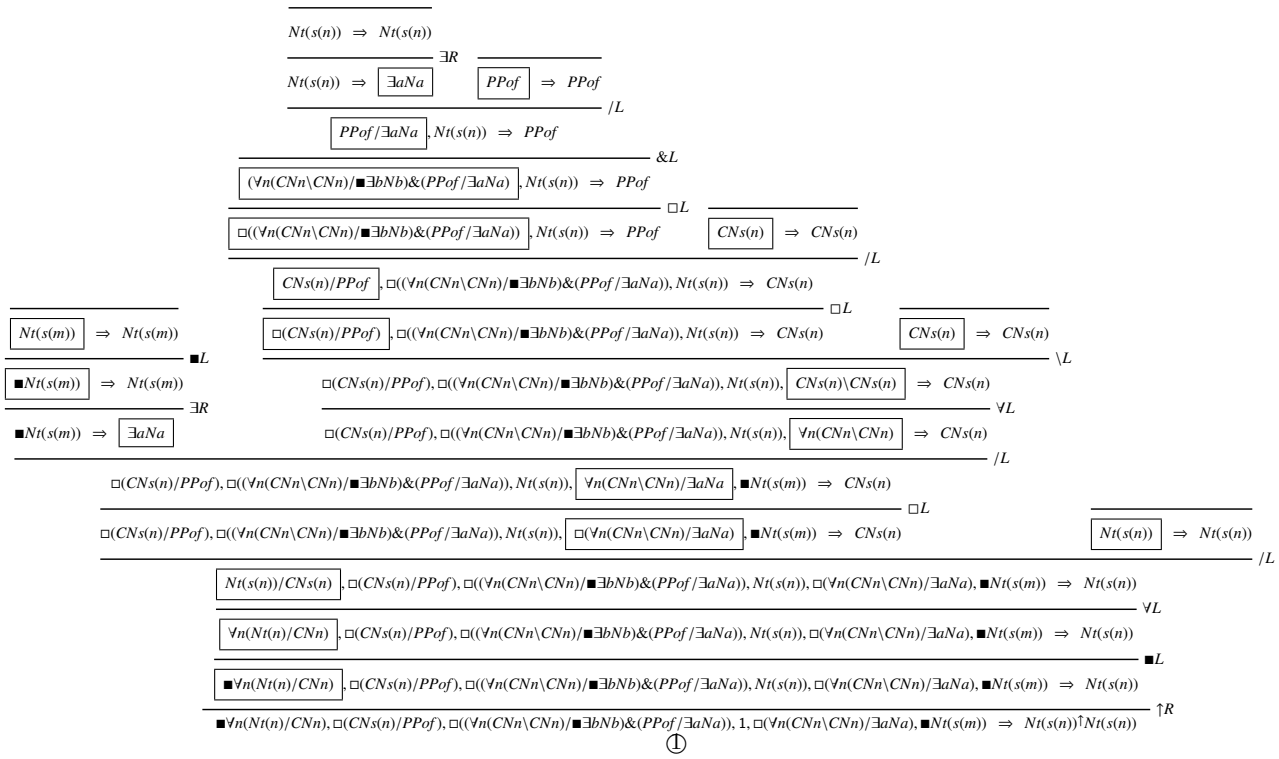
Lexical lookup yields:

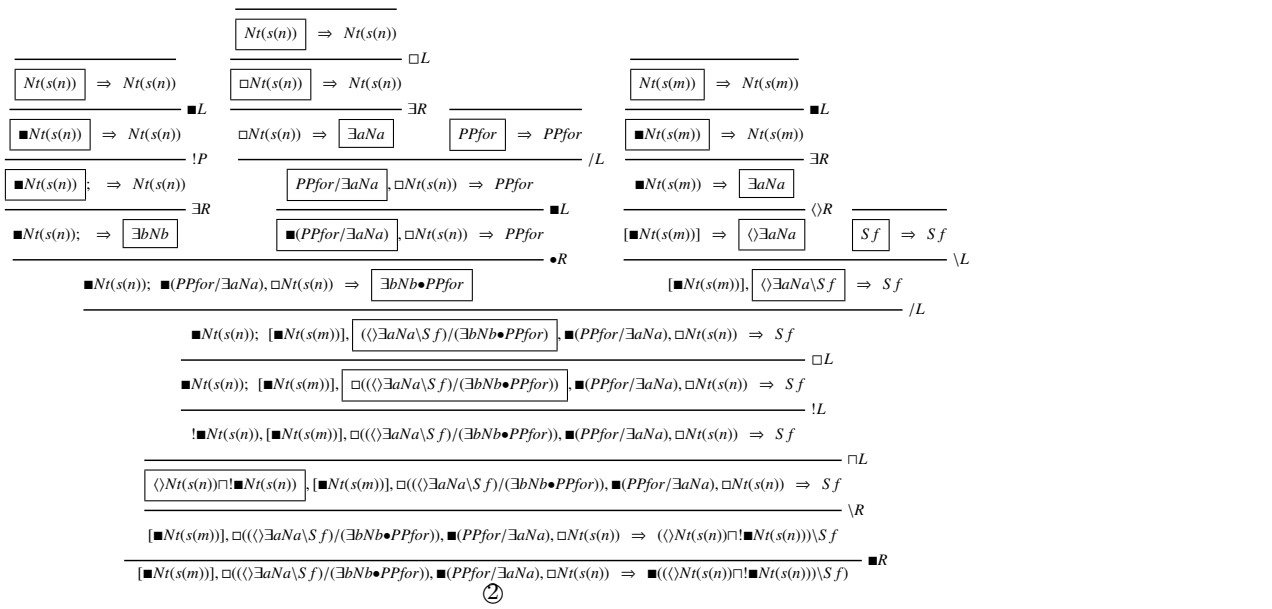
(25) $\Box CNs(n) : mountain, [[\blacksquare\forall n(Nt(n)/CNn) : \iota, \Box(CNs(n)/PPof) : \hat{}\lambda A((\check{}of\ A)\ \check{}painting),$
$\Box((\forall n(CNn\backslash CNn)/\blacksquare\exists bNb)\&(PPof/\exists aNa)) : \hat{}(\check{}of, \lambda BB), \blacksquare\forall n\forall m((Nt(n)\uparrow Nt(m))\downarrow$
$([]^{-1}[]^{-1}(CNm\backslash CNm)/\blacksquare((\langle\rangle Nt(n) \sqcap !\blacksquare Nt(n))\backslash S\,f))) : \lambda C\lambda D\lambda E\lambda F[(E\ F) \wedge (D\ (C\ F))],$
$\Box(\forall n(CNn\backslash CNn)/\exists aNa) : \hat{}\lambda G\lambda H((\check{}by\ G)\ H), \blacksquare Nt(s(m)) : c, [\blacksquare Nt(s(m)) : j],$
$\Box((\langle\rangle\exists aNa\backslash S\,f)/(\exists bNb \bullet PPfor)) : \hat{}\lambda I\lambda J(Past\ (((\check{}sell\ \pi_2 I)\ \pi_1 I)\ J)),$
$\blacksquare(PPfor/\exists aNa) : \lambda KK, \Box Nt(s(n)) : tenmilliondollars]] \Rightarrow CNs(n)$

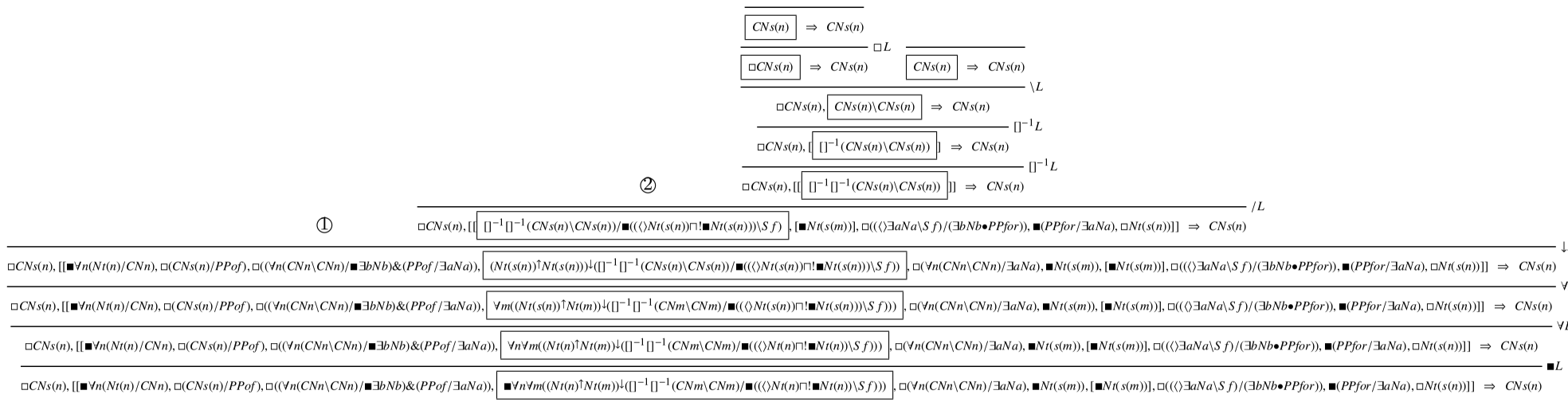
There is the derivation:

Figure 9: Derivation of **dog which Mary saw today**

Figure 10: Derivation of **John slept before Mary did**






CNs(n) ⇒ CNs(n)
□L
□CNs(n) ⇒ CNs(n)
CNs(n) ⇒ CNs(n)
\L
□CNs(n), CNs(n)\CNs(n) ⇒ CNs(n)
[]^-1L
□CNs(n), [[]^-1(CNs(n)\CNs(n))] ⇒ CNs(n)
[]^-1L
□CNs(n), [[[]^-1[]^-1(CNs(n)\CNs(n))]] ⇒ CNs(n)
②
/L
①
□CNs(n), [[[]^-1[]^-1(CNs(n)\CNs(n))/■((⟨⟩Nt(s(n))⊓!■Nt(s(n)))\S f)], [■Nt(s(m))], □(((⟨⟩∃aNa\S f)/(∃bNb•PPfor)), ■(PPfor/∃aNa), □Nt(s(n))]] ⇒ CNs(n)
↓L
□CNs(n), [[■∀n(Nt(n)/CNn), □(CNs(n)/PPof), □((∀n(CNn\CNn)/■∃bNb)&(PPof/∃aNa)), (Nt(s(n))↑Nt(s(n)))↓([]^-1[]^-1(CNs(n)\CNs(n))/■((⟨⟩Nt(s(n))⊓!■Nt(s(n)))\S f)) , □(∀n(CNn\CNn)/∃aNa), ■Nt(s(m)), [■Nt(s(m))], □(((⟨⟩∃aNa\S f)/(∃bNb•PPfor)), ■(PPfor/∃aNa), □Nt(s(n))]] ⇒ CNs(n)
∀L
□CNs(n), [[■∀n(Nt(n)/CNn), □(CNs(n)/PPof), □((∀n(CNn\CNn)/■∃bNb)&(PPof/∃aNa)), ∀m((Nt(s(n))↑Nt(m))↓([]^-1[]^-1(CNm\CNm)/■(((⟨⟩Nt(s(n))⊓!■Nt(s(n)))\S f))) , □(∀n(CNn\CNn)/∃aNa), ■Nt(s(m)), [■Nt(s(m))], □(((⟨⟩∃aNa\S f)/(∃bNb•PPfor)), ■(PPfor/∃aNa), □Nt(s(n))]] ⇒ CNs(n)
∀L
□CNs(n), [[■∀n(Nt(n)/CNn), □(CNs(n)/PPof), □((∀n(CNn\CNn)/■∃bNb)&(PPof/∃aNa)), ∀n∀m((Nt(n)↑Nt(m))↓([]^-1[]^-1(CNm\CNm)/■(((⟨⟩Nt(n)⊓!■Nt(n)))\S f))) , □(∀n(CNn\CNn)/∃aNa), ■Nt(s(m)), [■Nt(s(m))], □(((⟨⟩∃aNa\S f)/(∃bNb•PPfor)), ■(PPfor/∃aNa), □Nt(s(n))]] ⇒ CNs(n)
■L
□CNs(n), [[■∀n∀m((Nt(n)↑Nt(m))↓([]^-1[]^-1(CNm\CNm)/■(((⟨⟩Nt(n)⊓!■Nt(n)))\S f))) , □(∀n(CNn\CNn)/∃aNa), ■Nt(s(m)), [■Nt(s(m))], □(((⟨⟩∃aNa\S f)/(∃bNb•PPfor)), ■(PPfor/∃aNa), □Nt(s(n))]] ⇒ CNs(n)

This delivers semantics:

(26) $\lambda D[(\check{}\mathit{mountain}\ D) \wedge (\mathit{Past}\ (((\check{}\mathit{sell}\ \check{}\mathit{tenmilliondollars})\ (\iota\ ((\check{}\mathit{by}\ c)\ ((\check{}\mathit{of}\ D)\ \check{}\mathit{painting}))))\ j))]$

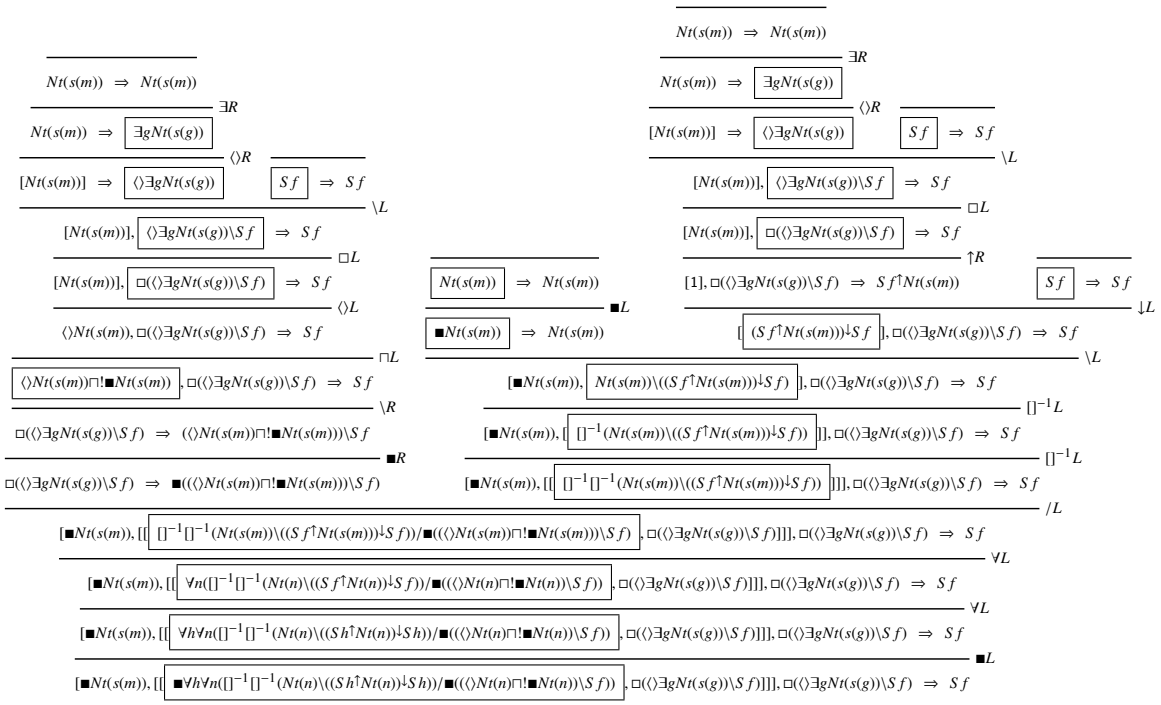
In appositive relativisation a relative clause, marked off by a prosodic phrase, modifies a full noun phrase:

(27) (tdc(67)) **[john**+[[**who**+**jogs**]]]+**sneezed** : $S\,f$

The lexical lookup yields:

(28) $[\blacksquare Nt(s(m)) : j, [[\blacksquare\forall h\forall n([]^{-1}[]^{-1}(Nt(n)\backslash((S\,h\uparrow Nt(n))\downarrow S\,h))/\blacksquare(((\langle\rangle Nt(n)\sqcap !\blacksquare Nt(n))\backslash S\,f)) :$
$\lambda A\lambda B\lambda C[(A\ B)\wedge(C\ B)], \square(\langle\rangle\exists gNt(s(g))\backslash S\,f) : \hat{}\lambda D(\mathit{Pres}\ (\check{}\mathit{jog}\ D))]]], \square(\langle\rangle\exists gNt(s(g))\backslash S\,f) :$
$\hat{}\lambda E(\mathit{Past}\ (\check{}\mathit{sneeze}\ E))\ \Rightarrow\ S\,f$

There is the derivation:



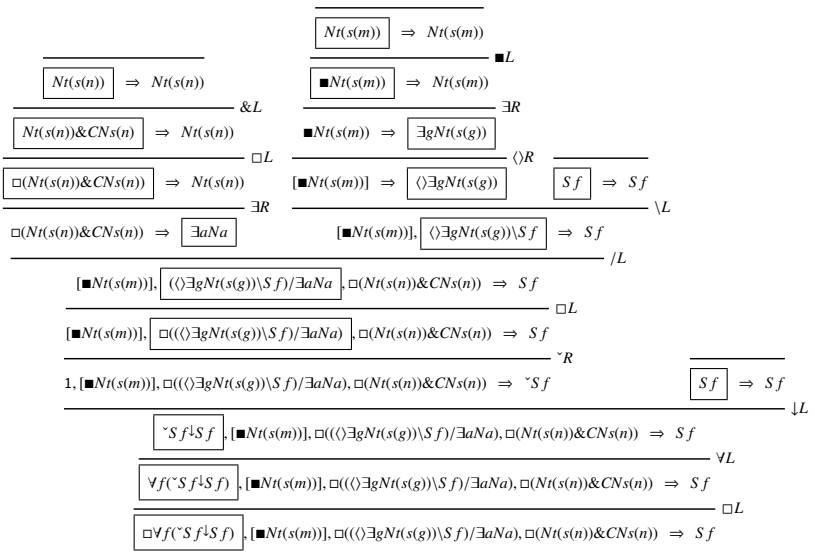
This delivers semantics:

(29) $[(\mathit{Pres}\ (\check{}\mathit{jog}\ j)) \wedge (\mathit{Past}\ (\check{}\mathit{sneeze}\ j))]$

There follow four placements of a parenthetical adverbial within a sentence all derived from the same types and all assigning the same semantics. First:

(30) (tdc(70a)) **fortunately**+[**john**]+**has**+**perseverance** : $S\,f$

(31) $\square\forall f(\check{}S\,f\downarrow S\,f) : \mathit{fortunately}, [\blacksquare Nt(s(m)) : j], \square((\langle\rangle\exists gNt(s(g))\backslash S\,f)/\exists aNa) :$
$\hat{}\lambda A\lambda B(\mathit{Pres}\ ((\check{}\mathit{have}\ A)\ B)), \square(Nt(s(n))\&CNs(n)) : \hat{}((\mathit{gen}\ \check{}\mathit{perseverance}), \check{}\mathit{perseverance})$
$\Rightarrow\ S\,f$

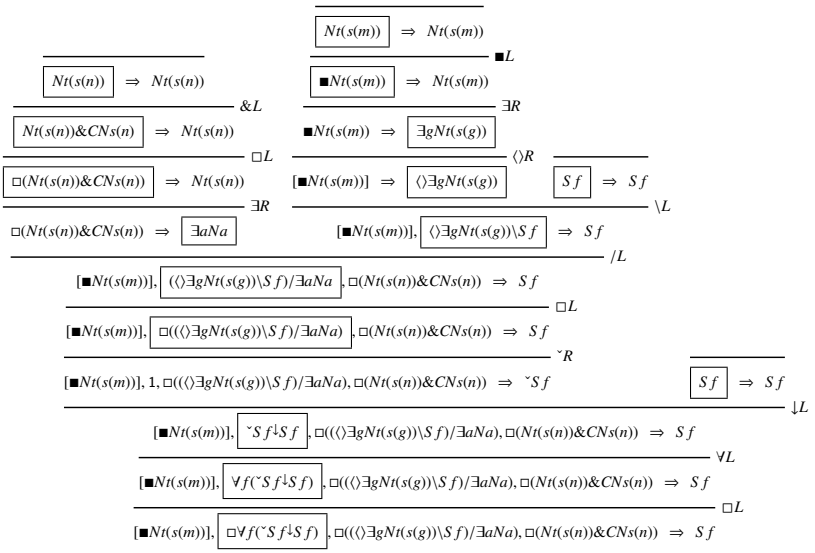


(32) (ˇ*fortunately* (*Pres* ((ˇ*have* (*gen* ˇ*perseverance*)) *j*)))

Second:

(33) (tdc(70b)) [**john**]+**fortunately**+**has**+**perseverance** : Sf

(34) $[\blacksquare Nt(s(m)) : j], \Box\forall f(\check{}Sf{\downarrow}Sf) : \textit{fortunately}, \Box((\langle\rangle\exists gNt(s(g))\backslash Sf)/\exists aNa) :$
$\hat{}\lambda A\lambda B(Pres\ ((\check{}have\ A)\ B)), \Box(Nt(s(n))\&CNs(n)) : \hat{}((gen\ \check{}perseverance), \check{}perseverance)$
$\Rightarrow Sf$

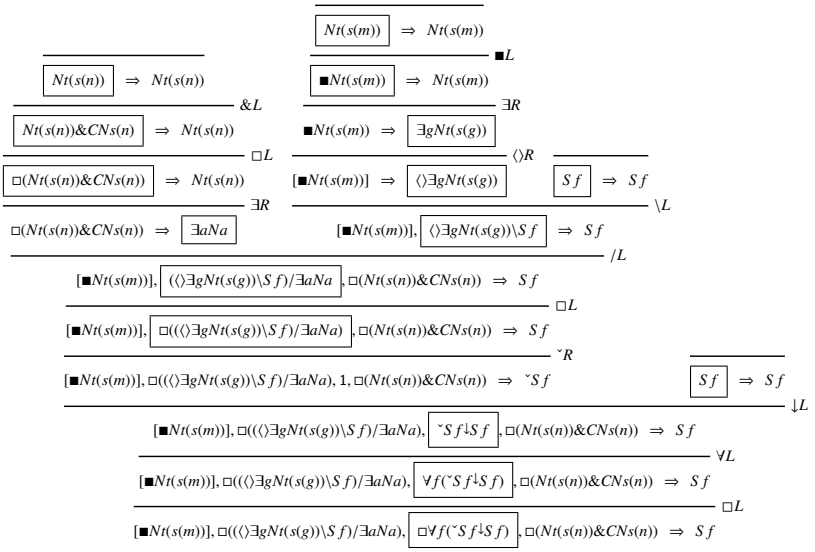


(35) (ˇ*fortunately* (*Pres* ((ˇ*have* (*gen* ˇ*perseverance*)) *j*)))

Third:

(36) (tdc(70c)) [**john**]+**has**+**fortunately**+**perseverance** : Sf

(37) $[\blacksquare Nt(s(m)) : j], \square((\langle\rangle \exists gNt(s(g)) \backslash S\, f)/\exists aNa) : \hat{}\,\lambda A \lambda B(Pres\ ((\check{}\,have\ A)\ B)), \square \forall f(\check{}\,S\, f \downarrow S\, f) :$
fortunately, $\square(Nt(s(n)) \& CNs(n)) : \hat{}\,((gen\ \check{}\,perseverance), \check{}\,perseverance) \Rightarrow S\, f$

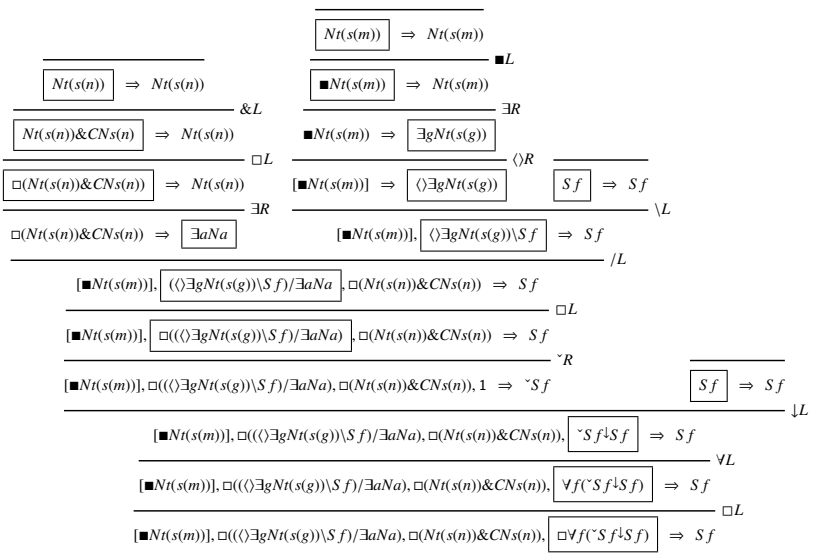


(38) (ˇ*fortunately* (*Pres* ((ˇ*have* (*gen* ˇ*perseverance*)) *j*)))

And fourth:

(39) (tdc(70d)) **[john]+has+perseverance+fortunately** : $S\, f$

(40) $[\blacksquare Nt(s(m)) : j], \square((\langle\rangle \exists gNt(s(g)) \backslash S\, f)/\exists aNa) : \hat{}\,\lambda A \lambda B(Pres\ ((\check{}\,have\ A)\ B)), \square(Nt(s(n)) \& CNs(n)) :$
$\hat{}\,((gen\ \check{}\,perseverance), \check{}\,perseverance), \square \forall f(\check{}\,S\, f \downarrow S\, f) : fortunately \Rightarrow S\, f$



(41) (ˇ*fortunately* (*Pres* ((ˇ*have* (*gen* ˇ*perseverance*)) *j*)))

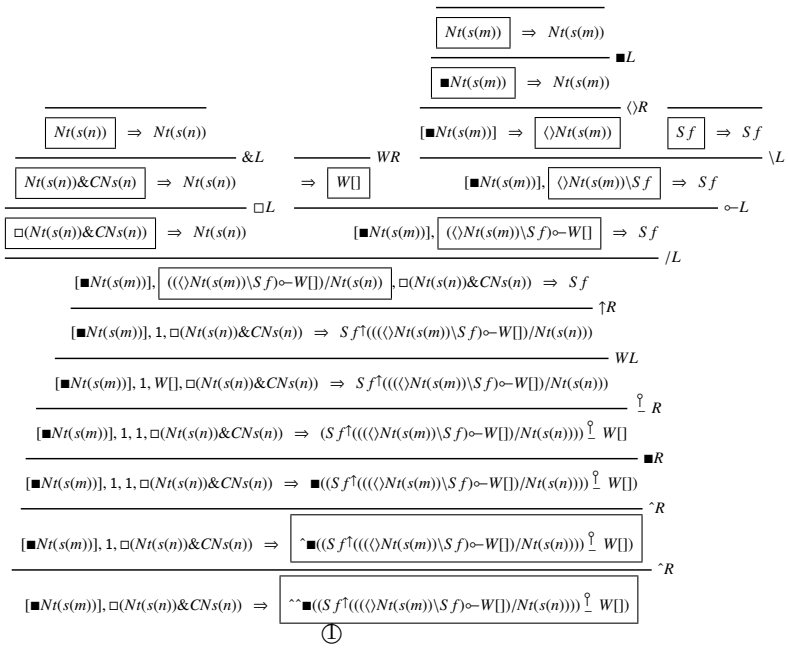
The next example is gapping:

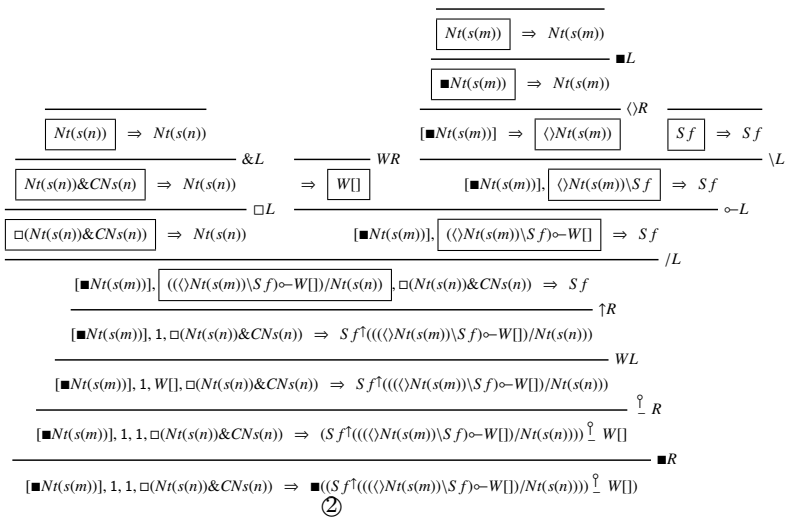
(42) (tdc(73)) [[[**john**]+**studies**+**logic**+**and**+[**charles**]+**phonetics**]] : Sf

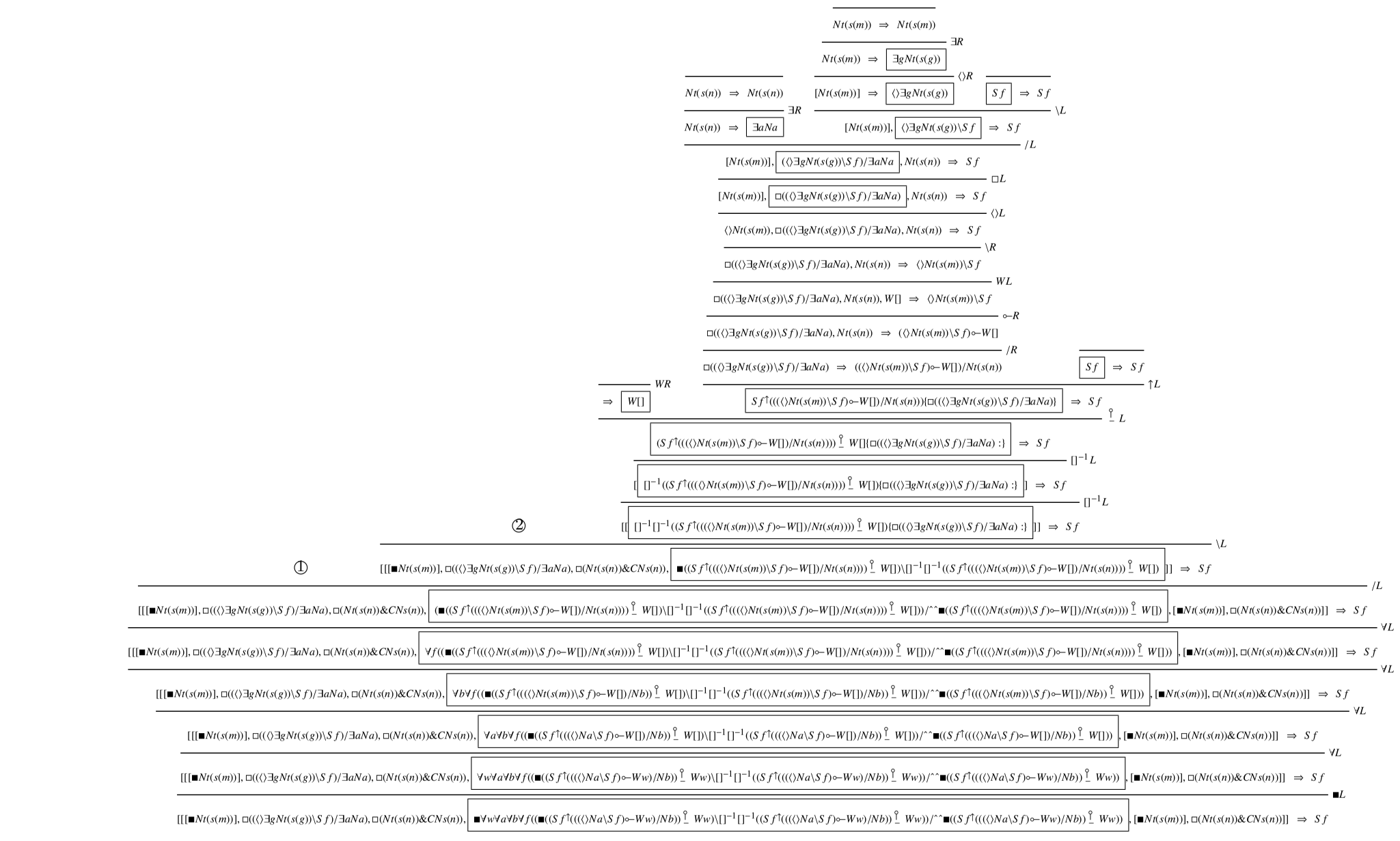
(Note the double brackets for the coordinate structure strong island.) The treatment of the example, however, is modified according to Morrill and Valentín (2014b) in view of the observations of Kubota and Levine (2012). Lexical lookup yields:

(43) $[[[\blacksquare Nt(s(m)) : j], \square((\langle\rangle \exists g Nt(s(g)) \backslash Sf)/\exists a Na) : \hat{}\lambda A \lambda B(Pres\ ((\check{}study\ A)\ B)),$
$\square(Nt(s(n))\&CNs(n)) : \hat{}((gen\ \check{}logic), \check{}logic), \blacksquare\forall w \forall a \forall b \forall f((\blacksquare((Sf\uparrow((((\langle\rangle Na \backslash Sf) \circ\!\!- Ww)/Nb)) \overset{\circ}{\underline{!}} Ww) \backslash []^{-1}[]^{-1}((Sf\uparrow((((\langle\rangle Na \backslash Sf) \circ\!\!- Ww)/Nb)) \overset{\circ}{\underline{!}} Ww))/\hat{}\hat{}\blacksquare((Sf\uparrow((((\langle\rangle Na \backslash Sf) \circ\!\!- Ww)/Nb)) \overset{\circ}{\underline{!}} Ww)) : \lambda C \lambda D \lambda E[(D\ E) \wedge (C\ E)], [\blacksquare Nt(s(m)) : c], \square(Nt(s(n))\&CNs(n)) :$
$\hat{}((gen\ \check{}phonetics), \check{}phonetics)]] \Rightarrow Sf$

There is the derivation:







This delivers semantics:

(44) $[(Pres\ ((\check{}study\ (gen\ \check{}logic))\ j)) \wedge (Pres\ ((\check{}study\ (gen\ \check{}phonetics))\ c))]$

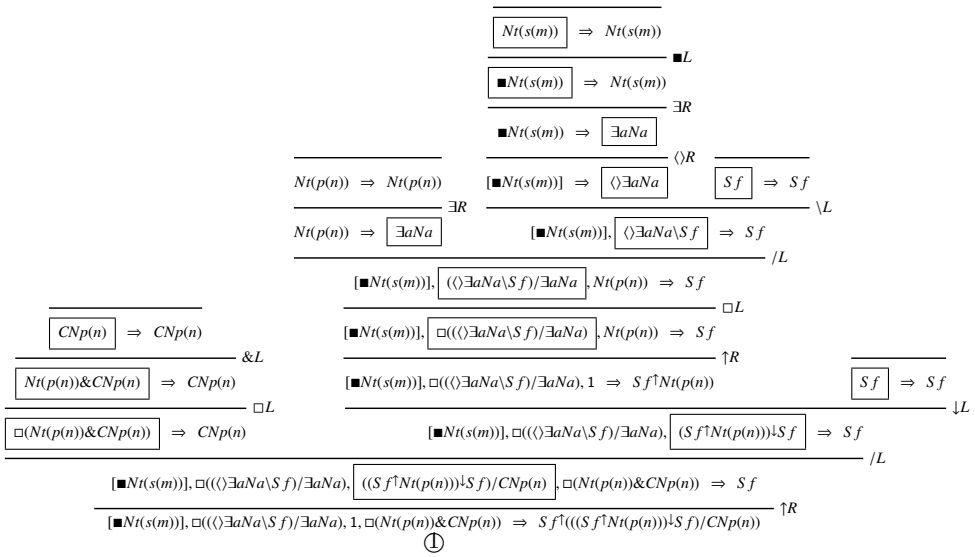
Example tdc(75) contains comparative subdeletion:

(45) (tdc(75)) **[john]+ate+more+donuts+than+[mary]+bought+bagels** : $S\,f$

Lexical lookup yields:

(46) $[\blacksquare Nt(s(m)) : j], \Box((\langle\rangle \exists aNa \backslash S\,f)/\exists aNa) : \hat{}\lambda A \lambda B(Past\ ((\check{}eat\ A)\ B)),$
$\blacksquare \forall h \forall g \forall f((S\,f\uparrow(((S\,h\uparrow Nt(p(g)))\downarrow S\,h)/CNp(g)))\downarrow(S\,f/\hat{}(CPthan\uparrow\blacksquare(((S\,h\uparrow Nt(p(g)))\downarrow S\,h)/CNp(g))))) :$
$\lambda C \lambda D[|\lambda E(C\ \lambda F \lambda G[(F\ E) \wedge (G\ E)])| > |\lambda H\check{}(D\ \lambda I \lambda J[(I\ H) \wedge (J\ H)])|], \Box(Nt(p(n)) \& CNp(n)) :$
$\hat{}((gen\ \check{}donuts), \check{}donuts), \blacksquare(CPthan/\Box S\,f) : \lambda K K, [\blacksquare Nt(s(f)) : m], \Box((\langle\rangle \exists aNa \backslash S\,f)/\exists aNa) :$
$\hat{}\lambda L \lambda M(Past\ ((\check{}buy\ L)\ M)), \Box(Nt(p(n)) \& CNp(n)) : \hat{}((gen\ \check{}bagels), \check{}bagels) \Rightarrow S\,f$

There is the derivation:



This delivers semantics:

(47) $[|\lambda C[(\check{}donuts\ C) \wedge (Past\ ((\check{}eat\ C)\ j))]| > |\lambda F[(\check{}bagels\ F) \wedge (Past\ ((\check{}buy\ F)\ m))]|]$

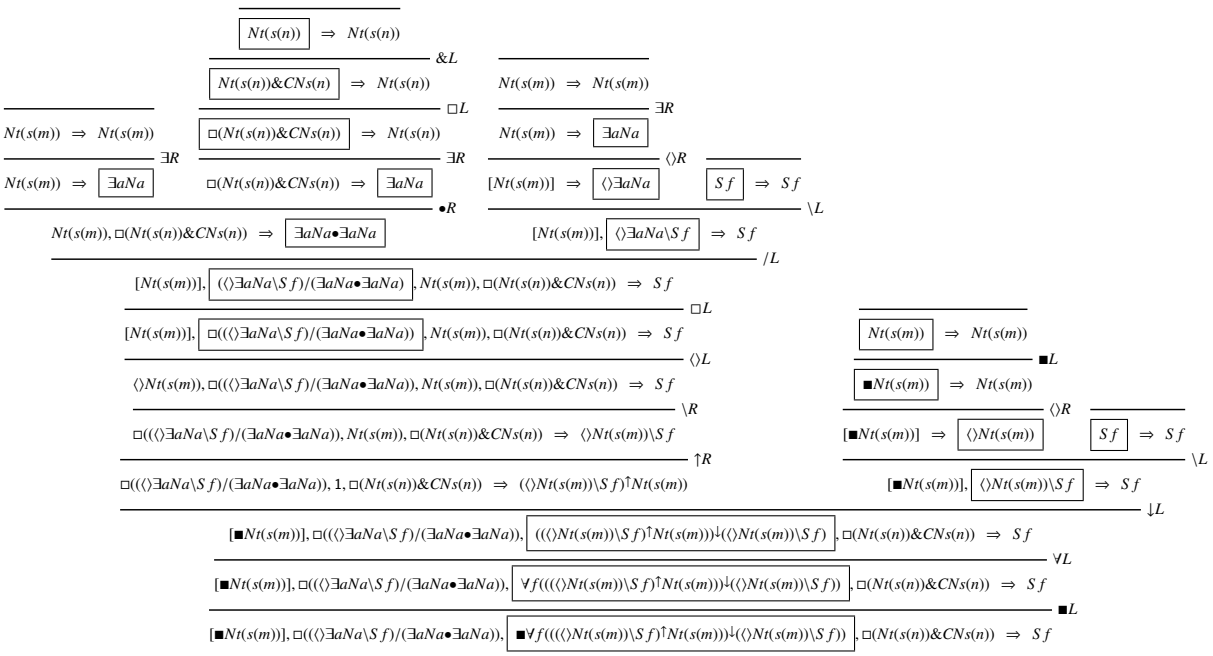
Finally, there is the medial reflexivisation:

(48) (tdc(86a)) **[john]+bought+himself+coffee** : $S\,f$

The lexical lookup yields:

(49) $[\blacksquare Nt(s(m)) : j], \Box((\langle\rangle \exists aNa \backslash S\,f)/(\exists aNa \bullet \exists aNa)) : \hat{}\,\lambda A \lambda B(Past\ (((\check{}\,buy\ \pi_1 A)\ \pi_2 A)\ B)),$
$\blacksquare \forall f((((\langle\rangle Nt(s(m)) \backslash S\,f) \uparrow Nt(s(m))) \downarrow (\langle\rangle Nt(s(m)) \backslash S\,f)) : \lambda C \lambda D((C\,D)\,D), \Box(Nt(s(n)) \& CNs(n)) :$
$\hat{}\,((gen\ \check{}\,coffee), \check{}\,coffee) \ \Rightarrow\ S\,f$

There is the derivation:



This delivers semantics:

(50) $(Past\ (((\check{}\,buy\ j)\ (gen\ \check{}\,coffee))\ j))$

5. Conclusions

It is our view that computational implementation of wide and deep empirical analysis is not only a desideratum, but also necessary, to verify the correct interaction of the myriad factors of language. With this paper we aim to set the pace on this methodology and we invite others to follow.

References

Andreoli, J. M. 1992. Logic programming with focusing in linear logic. Journal of Logic and Computation 2(3):297–347.

Dowty, David R., Robert E. Wall, and Stanley Peters. 1981. Introduction to Montague Semantics, vol. 11 of Synthese Language Library. Dordrecht: D. Reidel.

Kubota, Yusuke and Robert Levine. 2012. Gapping as like-category coordination. In D. Bechet and A. Dikovsky, eds., Logical Aspects of Computational Linguistics, vol.

7351 of Lecture Notes in Computer Science, 135–150. Springer Berlin Heidelberg. ISBN 978-3-642-31261-8.

Moortgat, Michael. 1997. Categorial Type Logics. In J. van Benthem and A. ter Meulen, eds., Handbook of Logic and Language, 93–177. Amsterdam and Cambridge, Massachusetts: Elsevier Science B.V. and the MIT Press.

Morrill, Glyn. 2011a. Logic Programming of the Displacement Calculus. In S. Pogodalla and J.-P. Prost, eds., Proceedings of Logical Aspects of Computational Linguistics 2011, LACL'11, Montpellier, No. LNAI 6736 in Springer Lecture Notes in AI, 175–189. Berlin: Springer.

Morrill, Glyn. 2012. CatLog: A Categorial Parser/Theorem-Prover. In LACL 2012 System Demonstrations, Logical Aspects of Computational Linguistics 2012, 13–16. Nantes.

Morrill, Glyn. 2014. A Categorial Type Logic. In M. M. Claudia Casadio, Bob Coeke and P. Scott, eds., Categories and Types in Logic, Language and Physics: Essays Dedicated to Jim Lambek on the Occasion of His 90th Birthday, vol. 8222 of LNCS, FoLLI Publications in Logic, Language and Information, 331–352. Berlin: Springer.

Morrill, Glyn and Oriol Valentín. 2010. Displacement Calculus. Linguistic Analysis 36(1–4):167–192. Special issue Festschrift for Joachim Lambek, http://arxiv.org/abs/1004.4181.

Morrill, Glyn and Oriol Valentín. 2014a. Displacement Logic for Anaphora. Journal of Computing and System Science 80:390–409. http://dx.doi.org/10.1016/j.jcss.2013.05.006.

Morrill, Glyn and Oriol Valentín. 2014b. Semantically Inactive Multiplicatives and Words as Types. In N. Asher and S. Soloviev, eds., Proceedings of Logical Aspects of Computational Linguistics, LACL'14, Toulouse, No. 8535 in LNCS, FoLLI Publications on Logic, Language and Information, 149–162. Berlin: Springer.

Morrill, Glyn, Oriol Valentín, and Mario Fadda. 2011. The Displacement Calculus. Journal of Logic, Language and Information 20(1):1–48. Doi 10.1007/s10849-010-9129-2.

Morrill, Glyn V. 1994. Type Logical Grammar: Categorial Logic of Signs. Dordrecht: Kluwer Academic Publishers.

Morrill, Glyn V. 2011b. Categorial Grammar: Logical Syntax, Semantics, and Processing. New York and Oxford: Oxford University Press.

Valentín, Oriol. 2012. Theory of Discontinuous Lambek Calculus. Ph.D. thesis, Universitat Autònoma de Barcelona, Barcelona.

Valentín, Oriol, Daniel Serret, and Glyn Morrill. 2013. A Count Invariant for Lambek Calculus with Additives and Bracket Modalities. In G. Morrill and M.-J. Nederhof, eds., Proceedings of Formal Grammar 2012 and 2013, vol. 8036 of Springer LNCS, FoLLI Publications in Logic, Language and Information, 263–276. Berlin: Springer.

van Benthem, J. 1991. Language in Action: Categories, Lambdas, and Dynamic Logic. No. 130 in Studies in Logic and the Foundations of Mathematics. Amsterdam: North-Holland. Revised student edition printed in 1995 by the MIT Press.

Appendix: CatLog output for Dutch verb raising and cross-serial dependency

kan : $(NA\backslash S\,i)\downarrow(NA\backslash S\,f) : \lambda B\lambda C((isable\ (B\ C))\ C)$
las : $NA\backslash(Nt(s(B))\backslash S\,f) : read$
wil : $(NA\backslash S\,i)\downarrow(NA\backslash S\,f) : \lambda B\lambda C((want\ (B\ C))\ C)$
kan : $Q/\hat{}(S\,f\uparrow((NA\backslash S\,i)\downarrow(NA\backslash S\,f))) : \lambda B(B\ \lambda C\lambda D((isable\ (C\ D))\ D))$
las : $Q/\hat{}(S\,f\uparrow(NA\backslash(Nt(s(B))\backslash S\,f))) : \lambda C(C\ read)$
wil : $Q/\hat{}(S\,f\uparrow((NA\backslash S\,i)\downarrow(NA\backslash S\,f))) : \lambda B(B\ \lambda C\lambda D((want\ (C\ D))\ D))$
alles : $(SA\uparrow Nt(s(n)))\downarrow SA : \lambda B\forall C[(thing\ C)\rightarrow(B\ C)]$
boeken : $Np(n) : books$
cecilia : $Nt(s(f)) : c$
de : $Nt(s(A))/CNA : the$
helpen : $\triangleright^{-1}((NA\backslash S\,i)\downarrow(NB\backslash(NA\backslash S\,i))) : \lambda C\lambda D((help\ (C\ D))\ D)$
henk : $Nt(s(m)) : h$
jan : $Nt(s(m)) : j$
kunnen : $\triangleright^{-1}((NA\backslash S\,i)\downarrow(NA\backslash S\,i)) : \lambda B\lambda C((isable\ (B\ C))\ C)$
lezen : $\triangleright^{-1}(NA\backslash(NB\backslash S\,i)) : read$
nijlpaarden : $CNp(n) : hippos$
voeren : $\triangleright^{-1}(NA\backslash(NB\backslash S\,i)) : feed$
zag : $(Nt(s(A))\backslash S\,i)\downarrow(NB\backslash(Nt(s(A))\backslash S\,f)) : \lambda C\lambda D((saw\ (C\ D))\ D)$

(d(1)) **jan+boeken+las** : $S\,f$

$Nt(s(m)) : j, Np(n) : books, NA\backslash(Nt(s(B))\backslash S\,f) : read \Rightarrow S\,f$

$$\dfrac{\dfrac{}{Np(n)\Rightarrow Np(n)}\quad\dfrac{\dfrac{}{Nt(s(m))\Rightarrow Nt(s(m))}\quad\dfrac{}{\boxed{S\,f}\Rightarrow S\,f}\backslash L}{Nt(s(m)),\boxed{Nt(s(m))\backslash S\,f}\Rightarrow S\,f}\backslash L}{Nt(s(m)),Np(n),\boxed{Np(n)\backslash(Nt(s(m))\backslash S\,f)}\Rightarrow S\,f}$$

((*read books*) *j*)

$Nt(s(m)) : j, Np(n) : books, Q/\hat{}(S\,f\uparrow(NA\backslash(Nt(s(B))\backslash S\,f))) : \lambda C(C\ read) \Rightarrow S\,f$

(d(2)) **jan+boeken+kan+lezen** : $S\,f$

$Nt(s(m)) : j, Np(n) : books, (NA\backslash S\,i)\downarrow(NA\backslash S\,f) : \lambda B\lambda C((isable\ (B\ C))\ C), \triangleright^{-1}(ND\backslash(NE\backslash S\,i)) : read \Rightarrow S\,f$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Np(n)\Rightarrow Np(n)}\quad\dfrac{\dfrac{}{Nt(s(m))\Rightarrow Nt(s(m))}\quad\dfrac{}{\boxed{Si\{1\}}\Rightarrow Si}\backslash L}{Nt(s(m)),\boxed{Nt(s(m))\backslash Si\{1\}}\Rightarrow Si}\backslash L}{Nt(s(m)),Np(n),\boxed{Np(n)\backslash(Nt(s(m))\backslash Si)\{1\}}\Rightarrow Si}\triangleright^{-1}L}{Nt(s(m)),Np(n),1,\boxed{\triangleright^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))}\Rightarrow Si}\backslash R}{Np(n),1,\triangleright^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))\Rightarrow Nt(s(m))\backslash Si}\quad\dfrac{\dfrac{}{Nt(s(m))\Rightarrow Nt(s(m))}\quad\dfrac{}{\boxed{S\,f}\Rightarrow S\,f}\backslash L}{Nt(s(m)),\boxed{Nt(s(m))\backslash S\,f}\Rightarrow S\,f}\backslash L}{Nt(s(m)),Np(n),\boxed{(Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash S\,f)},\triangleright^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))\Rightarrow S\,f}\downarrow L$$

((*isable* ((*read books*) *j*)) *j*)

$Nt(s(m)) : j, Np(n) : books, Q/\hat{}(S\,f\uparrow((NA\backslash S\,i)\downarrow(NA\backslash S\,f))) : \lambda B(B\ \lambda C\lambda D((isable\ (C\ D))\ D)), \triangleright^{-1}(NE\backslash(NF\backslash S\,i)) : read \Rightarrow S\,f$

(d(3)) **jan+boeken+wil+kunnen+lezen** : $S\,f$

$Nt(s(m)) : j, Np(n) : books, (NA\backslash S\,i)\downarrow(NA\backslash S\,f) : \lambda B\lambda C((want\ (B\ C))\ C), \triangleright^{-1}((ND\backslash S\,i)\downarrow(ND\backslash S\,i)) : \lambda E\lambda F((isable\ (E\ F))\ F), \triangleright^{-1}(NG\backslash(NH\backslash S\,i)) : read \Rightarrow S\,f$

$$\dfrac{\dfrac{\dfrac{}{Np(n)\ \Rightarrow\ Np(n)}\quad\dfrac{\dfrac{}{Nt(s(m))\ \Rightarrow\ Nt(s(m))}\quad\dfrac{}{\boxed{Si\{1\}}\ \Rightarrow\ Si}}{Nt(s(m)),\boxed{Nt(s(m))\backslash Si\{1\}}\ \Rightarrow\ Si}\backslash L}{Nt(s(m)),Np(n),\boxed{Np(n)\backslash(Nt(s(m))\backslash Si)\{1\}}\ \Rightarrow\ Si}\backslash L}{Nt(s(m)),Np(n),1,\boxed{\triangleright^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))}\ \Rightarrow\ Si}\triangleright^{-1}L\quad\dfrac{\dfrac{}{Nt(s(m))\ \Rightarrow\ Nt(s(m))}\quad\dfrac{}{\boxed{Si\{1\}}\ \Rightarrow\ Si}}{Nt(s(m)),\boxed{Nt(s(m))\backslash Si\{1\}}\ \Rightarrow\ Si}\backslash L}{Np(n),1,\triangleright^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Nt(s(m))\backslash Si}\backslash R\qquad\qquad}{Nt(s(m)),Np(n),\boxed{(Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash Si)\{1\}},\triangleright^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Si}\downarrow L}{Nt(s(m)),Np(n),1,\boxed{\triangleright^{-1}((Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash Si))},\triangleright^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Si}\triangleright^{-1}L\quad\dfrac{\dfrac{}{Nt(s(m))\ \Rightarrow\ Nt(s(m))}\quad\dfrac{}{\boxed{Sf}\ \Rightarrow\ Sf}}{Nt(s(m)),\boxed{Nt(s(m))\backslash Sf}\ \Rightarrow\ Sf}\backslash L}{Np(n),1,\triangleright^{-1}((Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash Si)),\triangleright^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Nt(s(m))\backslash Si}\backslash R\qquad\qquad}{Nt(s(m)),Np(n),\boxed{(Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash Sf)},\triangleright^{-1}((Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash Si)),\triangleright^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Sf}\downarrow L$$

((*want* ((*isable* ((*read books*) *j*)) *j*)) *j*)

$Nt(s(m)) : j, Np(n) : books, Q/\hat{}(Sf\uparrow((NA\backslash Si)\downarrow(NA\backslash Sf))) : \lambda B(B\ \lambda C\lambda D((want\ (C\ D))\ D)), \triangleright^{-1}((NE\backslash Si)\downarrow(NE\backslash Si)) : \lambda F\lambda G((isable\ (F\ G))\ G), \triangleright^{-1}(NH\backslash(NI\backslash Si)) : read\ \Rightarrow\ Sf$

(d(4)) **jan+alles+las** : Sf

$Nt(s(m)) : j, (SA\uparrow Nt(s(n)))\downarrow SA : \lambda B\forall C[(thing\ C) \rightarrow (B\ C)], ND\backslash(Nt(s(E))\backslash Sf) : read\ \Rightarrow\ Sf$

$$\dfrac{\dfrac{\dfrac{}{Nt(s(n))\ \Rightarrow\ Nt(s(n))}\quad\dfrac{\dfrac{}{Nt(s(m))\ \Rightarrow\ Nt(s(m))}\quad\dfrac{}{\boxed{Sf}\ \Rightarrow\ Sf}}{Nt(s(m)),\boxed{Nt(s(m))\backslash Sf}\ \Rightarrow\ Sf}\backslash L}{Nt(s(m)),Nt(s(n)),\boxed{Nt(s(n))\backslash(Nt(s(m))\backslash Sf)}\ \Rightarrow\ Sf}\backslash L}{Nt(s(m)),1,Nt(s(n))\backslash(Nt(s(m))\backslash Sf)\ \Rightarrow\ Sf\uparrow Nt(s(n))}\uparrow R\qquad\dfrac{}{\boxed{Sf}\ \Rightarrow\ Sf}}{Nt(s(m)),\boxed{(Sf\uparrow Nt(s(n)))\downarrow Sf},Nt(s(n))\backslash(Nt(s(m))\backslash Sf)\ \Rightarrow\ Sf}\downarrow L$$

$\forall B[(thing\ B) \rightarrow ((read\ B)\ j)]$

$Nt(s(m)) : j, (SA\uparrow Nt(s(n)))\downarrow SA : \lambda B\forall C[(thing\ C) \rightarrow (B\ C)], Q/\hat{}(Sf\uparrow(ND\backslash(Nt(s(E))\backslash Sf))) : \lambda F(F\ read)\ \Rightarrow\ Sf$

(d(5)) **jan+alles+kan+lezen** : Sf

$Nt(s(m)) : j, (SA\uparrow Nt(s(n)))\downarrow SA : \lambda B\forall C[(thing\ C) \rightarrow (B\ C)], (ND\backslash Si)\downarrow(ND\backslash Sf) : \lambda E\lambda F((isable\ (E\ F))\ F), \triangleright^{-1}(NG\backslash(NH\backslash Si)) : read\ \Rightarrow\ Sf$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{Nt(s(n))\ \Rightarrow\ Nt(s(n))}\quad\dfrac{\dfrac{}{Nt(s(m))\ \Rightarrow\ Nt(s(m))}\quad\dfrac{}{\boxed{Si\{1\}}\ \Rightarrow\ Si}}{Nt(s(m)),\boxed{Nt(s(m))\backslash Si\{1\}}\ \Rightarrow\ Si}\backslash L}{Nt(s(m)),Nt(s(n)),\boxed{Nt(s(n))\backslash(Nt(s(m))\backslash Si)\{1\}}\ \Rightarrow\ Si}\backslash L}{Nt(s(m)),Nt(s(n)),1,\boxed{\triangleright^{-1}(Nt(s(n))\backslash(Nt(s(m))\backslash Si))}\ \Rightarrow\ Si}\triangleright^{-1}L\quad\dfrac{\dfrac{}{Nt(s(m))\ \Rightarrow\ Nt(s(m))}\quad\dfrac{}{\boxed{Sf}\ \Rightarrow\ Sf}}{Nt(s(m)),\boxed{Nt(s(m))\backslash Sf}\ \Rightarrow\ Sf}\backslash L}{Nt(s(n)),1,\triangleright^{-1}(Nt(s(n))\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Nt(s(m))\backslash Si}\backslash R\qquad\qquad}{Nt(s(m)),Nt(s(n)),\boxed{(Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash Sf)},\triangleright^{-1}(Nt(s(n))\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Sf}\uparrow R\qquad\qquad}{Nt(s(m)),1,(Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash Sf),\triangleright^{-1}(Nt(s(n))\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Sf\uparrow Nt(s(n))}\quad\dfrac{}{\boxed{Sf}\ \Rightarrow\ Sf}}{Nt(s(m)),\boxed{(Sf\uparrow Nt(s(n)))\downarrow Sf},(Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash Sf),\triangleright^{-1}(Nt(s(n))\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Sf}\downarrow L$$

$\forall B[(thing\ B) \rightarrow ((isable\ ((read\ B)\ j))\ j)]$

$Nt(s(m)) : j, (SA\uparrow Nt(s(n)))\downarrow SA : \lambda B\forall C[(thing\ C) \rightarrow (B\ C)], Q/\hat{}(Sf\uparrow((ND\backslash Si)\downarrow(ND\backslash Sf))) : \lambda E(E\ \lambda F\lambda G((isable\ (F\ G))\ G)), \triangleright^{-1}(NH\backslash(NI\backslash Si)) : read\ \Rightarrow\ Sf$

(d(6)) **jan+cecilia+henk+de+nijlpaarden+zag+helpen+voeren** : Sf

$Nt(s(m)) : j, Nt(s(f)) : c, Nt(s(m)) : h, Nt(s(A))/CNA : the, CNp(n) : hippos, (Nt(s(B))\backslash Si)\downarrow(NC\backslash(Nt(s(B))\backslash Sf)) : \lambda D\lambda E((saw\ (D\ E))\ E), \triangleright^{-1}((NF\backslash Si)\downarrow(NG\backslash(NF\backslash Si))) :$

$\lambda H \lambda I((help\ (H\ I))\ I), \triangleright^{-1}(NJ\backslash(NK\backslash Si)) : feed \;\Rightarrow\; Sf$

$$\dfrac{CNp(n) \Rightarrow CNp(n) \qquad \boxed{Nt(s(p(n)))} \Rightarrow Nt(s(p(n)))}{\boxed{Nt(s(p(n)))/CNp(n)}, CNp(n) \Rightarrow Nt(s(p(n)))}\ /L$$

$$\dfrac{Nt(s(m)) \Rightarrow Nt(s(m)) \qquad \boxed{Si\{1\}} \Rightarrow Si}{Nt(s(m)), \boxed{Nt(s(m))\backslash Si\{1\}} \Rightarrow Si}\ \backslash L$$

$$\dfrac{\boxed{Nt(s(p(n)))/CNp(n)}, CNp(n) \Rightarrow Nt(s(p(n))) \qquad Nt(s(m)), \boxed{Nt(s(m))\backslash Si\{1\}} \Rightarrow Si}{Nt(s(m)), Nt(s(p(n)))/CNp(n), CNp(n), \boxed{Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si)\{1\}} \Rightarrow Si}\ \backslash L$$

$$\dfrac{Nt(s(m)), Nt(s(p(n)))/CNp(n), CNp(n), \boxed{Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si)\{1\}} \Rightarrow Si}{Nt(s(m)), Nt(s(p(n)))/CNp(n), CNp(n), 1, \boxed{\triangleright^{-1}(Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si))} \Rightarrow Si}\ \triangleright^{-1}L$$

$$\dfrac{Nt(s(m)), Nt(s(p(n)))/CNp(n), CNp(n), 1, \boxed{\triangleright^{-1}(Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si))} \Rightarrow Si}{Nt(s(p(n)))/CNp(n), CNp(n), 1, \triangleright^{-1}(Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si)) \Rightarrow Nt(s(m))\backslash Si}\ \backslash R$$

$$\dfrac{Nt(s(m)) \Rightarrow Nt(s(m)) \qquad \boxed{Si\{1\}} \Rightarrow Si}{Nt(s(m)), \boxed{Nt(s(m))\backslash Si\{1\}} \Rightarrow Si}\ \backslash L$$

$$\dfrac{Nt(s(m)) \Rightarrow Nt(s(m)) \qquad Nt(s(m)), \boxed{Nt(s(m))\backslash Si\{1\}} \Rightarrow Si}{Nt(s(m)), Nt(s(m)), \boxed{Nt(s(m))\backslash(Nt(s(m))\backslash Si)\{1\}} \Rightarrow Si}\ \backslash L$$

$$\dfrac{Nt(s(p(n)))/CNp(n), CNp(n), 1, \triangleright^{-1}(Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si)) \Rightarrow Nt(s(m))\backslash Si \qquad Nt(s(m)), Nt(s(m)), \boxed{Nt(s(m))\backslash(Nt(s(m))\backslash Si)\{1\}} \Rightarrow Si}{Nt(s(m)), Nt(s(m)), Nt(s(p(n)))/CNp(n), CNp(n), \boxed{(Nt(s(m))\backslash Si){\downarrow}(Nt(s(m))\backslash(Nt(s(m))\backslash Si))\{1\}}, \triangleright^{-1}(Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si)) \Rightarrow Si}\ {\downarrow}L$$

$$\dfrac{Nt(s(m)), Nt(s(m)), Nt(s(p(n)))/CNp(n), CNp(n), \boxed{(Nt(s(m))\backslash Si){\downarrow}(Nt(s(m))\backslash(Nt(s(m))\backslash Si))\{1\}}, \triangleright^{-1}(Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si)) \Rightarrow Si}{Nt(s(m)), Nt(s(m)), Nt(s(p(n)))/CNp(n), CNp(n), 1, \boxed{\triangleright^{-1}((Nt(s(m))\backslash Si){\downarrow}(Nt(s(m))\backslash(Nt(s(m))\backslash Si)))}, \triangleright^{-1}(Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si)) \Rightarrow Si}\ \triangleright^{-1}L$$

$$\dfrac{Nt(s(m)), Nt(s(m)), Nt(s(p(n)))/CNp(n), CNp(n), 1, \boxed{\triangleright^{-1}((Nt(s(m))\backslash Si){\downarrow}(Nt(s(m))\backslash(Nt(s(m))\backslash Si)))}, \triangleright^{-1}(Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si)) \Rightarrow Si}{Nt(s(m)), Nt(s(p(n)))/CNp(n), CNp(n), 1, \triangleright^{-1}((Nt(s(m))\backslash Si){\downarrow}(Nt(s(m))\backslash(Nt(s(m))\backslash Si))), \triangleright^{-1}(Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si)) \Rightarrow Nt(s(m))\backslash Si}\ \backslash R$$

$$\dfrac{Nt(s(m)) \Rightarrow Nt(s(m)) \qquad \boxed{Sf} \Rightarrow Sf}{Nt(s(m)), \boxed{Nt(s(m))\backslash Sf} \Rightarrow Sf}\ \backslash L$$

$$\dfrac{Nt(s(f)) \Rightarrow Nt(s(f)) \qquad Nt(s(m)), \boxed{Nt(s(m))\backslash Sf} \Rightarrow Sf}{Nt(s(m)), Nt(s(f)), \boxed{Nt(s(f))\backslash(Nt(s(m))\backslash Sf)} \Rightarrow Sf}\ \backslash L$$

$$\dfrac{Nt(s(m)), Nt(s(p(n)))/CNp(n), CNp(n), 1, \triangleright^{-1}((Nt(s(m))\backslash Si){\downarrow}(Nt(s(m))\backslash(Nt(s(m))\backslash Si))), \triangleright^{-1}(Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si)) \Rightarrow Nt(s(m))\backslash Si \qquad Nt(s(m)), Nt(s(f)), \boxed{Nt(s(f))\backslash(Nt(s(m))\backslash Sf)} \Rightarrow Sf}{Nt(s(m)), Nt(s(f)), Nt(s(m)), Nt(s(p(n)))/CNp(n), CNp(n), \boxed{(Nt(s(m))\backslash Si){\downarrow}(Nt(s(f))\backslash(Nt(s(m))\backslash Sf))}, \triangleright^{-1}((Nt(s(m))\backslash Si){\downarrow}(Nt(s(m))\backslash(Nt(s(m))\backslash Si))), \triangleright^{-1}(Nt(s(p(n)))\backslash(Nt(s(m))\backslash Si)) \Rightarrow Sf}\ {\downarrow}L$$

$(((saw\ (((help\ ((feed\ (the\ hippos))\ h))\ h)\ c))\ c)\ j)$

(d(7)) **wil+jan+boeken+lezen** : Q

$(NA\backslash Si)\downarrow(NA\backslash Sf) : \lambda B\lambda C((want\ (B\ C))\ C), Nt(s(m)) : j, Np(n) : books, \rhd^{-1}(ND\backslash(NE\backslash Si)) : read \ \Rightarrow\ Q$

$Q/\hat{}(Sf\uparrow((NA\backslash Si)\downarrow(NA\backslash Sf))) : \lambda B(B\ \lambda C\lambda D((want\ (C\ D))\ D)), Nt(s(m)) : j, Np(n) : books, \rhd^{-1}(NE\backslash(NF\backslash Si)) : read \ \Rightarrow\ Q$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Nt(s(m))\ \Rightarrow\ Nt(s(m))}\quad\dfrac{}{\boxed{Si\{1\}}\ \Rightarrow\ Si}}{Np(n)\ \Rightarrow\ Np(n)\qquad Nt(s(m)), \boxed{Nt(s(m))\backslash Si\{1\}}\ \Rightarrow\ Si}\backslash L}{Nt(s(m)), Np(n), \boxed{Np(n)\backslash(Nt(s(m))\backslash Si)\{1\}}\ \Rightarrow\ Si}\backslash L}{Nt(s(m)), Np(n), 1, \boxed{\rhd^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))}\ \Rightarrow\ Si}\rhd^{-1}L \quad \dfrac{}{Nt(s(m))\ \Rightarrow\ Nt(s(m))}\quad\dfrac{}{\boxed{Sf}\ \Rightarrow\ Sf}}{Np(n), 1, \rhd^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Nt(s(m))\backslash Si}\backslash R \quad \dfrac{}{Nt(s(m)), \boxed{Nt(s(m))\backslash Sf}\ \Rightarrow\ Sf}\backslash L}{Nt(s(m)), Np(n), \boxed{(Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash Sf)}, \rhd^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Sf}\downarrow L}{Nt(s(m)), Np(n), 1, \rhd^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Sf\uparrow((Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash Sf))}\uparrow R}{Nt(s(m)), Np(n), \rhd^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ \boxed{\hat{}(Sf\uparrow((Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash Sf)))}}\hat{}R \quad \dfrac{}{\boxed{Q}\ \Rightarrow\ Q}}{\boxed{Q/\hat{}(Sf\uparrow((Nt(s(m))\backslash Si)\downarrow(Nt(s(m))\backslash Sf)))}, Nt(s(m)), Np(n), \rhd^{-1}(Np(n)\backslash(Nt(s(m))\backslash Si))\ \Rightarrow\ Q}/L$$

$((want\ ((read\ books)\ j))\ j)$

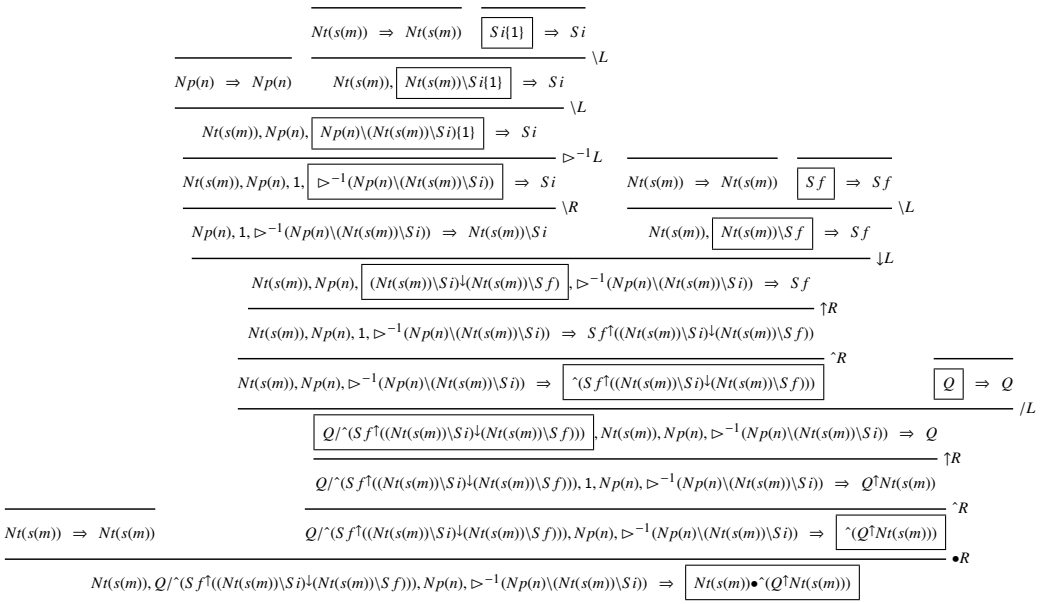
(d(8)) **jan+wil+boeken+lezen** : $Nt(s(m))\bullet\hat{}(Q\uparrow Nt(s(m)))$

$Nt(s(m)) : j, (NA\backslash Si)\downarrow(NA\backslash Sf) : \lambda B\lambda C((want\ (B\ C))\ C), Np(n) : books, \rhd^{-1}(ND\backslash(NE\backslash Si)) : read \ \Rightarrow\ Nt(s(m))\bullet\hat{}(Q\uparrow Nt(s(m)))$

$Nt(s(m)) : j, Q/\hat{}(Sf\uparrow((NA\backslash Si)\downarrow(NA\backslash Sf))) : \lambda B(B\ \lambda C\lambda D((want\ (C\ D))\ D)), Np(n) : books, \rhd^{-1}(NE\backslash(NF\backslash Si)) : read \ \Rightarrow\ Nt(s(m))\bullet\hat{}(Q\uparrow Nt(s(m)))$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Np(n)\ \Rightarrow\ Np(n)}\quad\dfrac{}{\boxed{Si\{1\}}\ \Rightarrow\ Si}}{Nt(s(m))\ \Rightarrow\ Nt(s(m))\qquad Np(n), \boxed{Np(n)\backslash Si\{1\}}\ \Rightarrow\ Si}\backslash L}{Np(n), Nt(s(m)), \boxed{Nt(s(m))\backslash(Np(n)\backslash Si)\{1\}}\ \Rightarrow\ Si}\backslash L}{Np(n), Nt(s(m)), 1, \boxed{\rhd^{-1}(Nt(s(m))\backslash(Np(n)\backslash Si))}\ \Rightarrow\ Si}\rhd^{-1}L \quad \dfrac{}{Np(n)\ \Rightarrow\ Np(n)}\quad\dfrac{}{\boxed{Sf}\ \Rightarrow\ Sf}}{Nt(s(m)), 1, \rhd^{-1}(Nt(s(m))\backslash(Np(n)\backslash Si))\ \Rightarrow\ Np(n)\backslash Si}\backslash R \quad \dfrac{}{Np(n), \boxed{Np(n)\backslash Sf}\ \Rightarrow\ Sf}\backslash L}{Np(n), Nt(s(m)), \boxed{(Np(n)\backslash Si)\downarrow(Np(n)\backslash Sf)}, \rhd^{-1}(Nt(s(m))\backslash(Np(n)\backslash Si))\ \Rightarrow\ Sf}\downarrow L}{Np(n), Nt(s(m)), 1, \rhd^{-1}(Nt(s(m))\backslash(Np(n)\backslash Si))\ \Rightarrow\ Sf\uparrow((Np(n)\backslash Si)\downarrow(Np(n)\backslash Sf))}\uparrow R}{Np(n), Nt(s(m)), \rhd^{-1}(Nt(s(m))\backslash(Np(n)\backslash Si))\ \Rightarrow\ \boxed{\hat{}(Sf\uparrow((Np(n)\backslash Si)\downarrow(Np(n)\backslash Sf)))}}\hat{}R \quad \dfrac{}{\boxed{Q}\ \Rightarrow\ Q}}{\boxed{Q/\hat{}(Sf\uparrow((Np(n)\backslash Si)\downarrow(Np(n)\backslash Sf)))}, Np(n), Nt(s(m)), \rhd^{-1}(Nt(s(m))\backslash(Np(n)\backslash Si))\ \Rightarrow\ Q}/L}{Q/\hat{}(Sf\uparrow((Np(n)\backslash Si)\downarrow(Np(n)\backslash Sf))), Np(n), 1, \rhd^{-1}(Nt(s(m))\backslash(Np(n)\backslash Si))\ \Rightarrow\ Q\uparrow Nt(s(m))}\uparrow R}{Q/\hat{}(Sf\uparrow((Np(n)\backslash Si)\downarrow(Np(n)\backslash Sf))), Np(n), \rhd^{-1}(Nt(s(m))\backslash(Np(n)\backslash Si))\ \Rightarrow\ \boxed{\hat{}(Q\uparrow Nt(s(m)))}}\hat{}R \quad \dfrac{}{Nt(s(m))\ \Rightarrow\ Nt(s(m))}}{Nt(s(m)), Q/\hat{}(Sf\uparrow((Np(n)\backslash Si)\downarrow(Np(n)\backslash Sf))), Np(n), \rhd^{-1}(Nt(s(m))\backslash(Np(n)\backslash Si))\ \Rightarrow\ \boxed{Nt(s(m))\bullet\hat{}(Q\uparrow Nt(s(m)))}}\bullet R$$

$(j, \lambda A((want\ ((read\ A)\ books))\ books))$



$(j, \lambda A((want\ ((read\ books)\ A))\ A))$

Coordination in Linear Categorial Grammar with Phenogrammatical Subtyping*

Carl Pollard
Ohio State University

Chris Worth
Ohio State University

Abstract

The left and right implications (\ and /) of Lambek grammars are well-equipped to analyze functor coordination, since they provide a way to encode the notion of like directionality with respect to argument position. However, they suffer difficulty in analyzing medial extraction and quantifier raising (QR), necessitating the addition of other connectives, exemplified by frameworks such as displacement calculus, multimodal type-logical grammar, and NL_λ. By contrast, 'curryesque frameworks' such as abstract categorial grammar, lambda grammar, and linear categorial grammar (LCG), with the single nondirectional implication $\multimap$, exhibit opposite characteristics. By separating the abstract combinatorics (tectogrammar) from their concrete representation (phenogrammar), they treat extraction and QR easily, at the expense of a straightforward account of coordination. One solution, adopted by hybrid type-logical grammar (HTLG), is to augment a curryesque framework with Lambek-style directional implications. However, this strategy gives rise to new problems, as well as abandoning the sharp distinction between phenogrammar and tectogrammar pioneered by Curry. In this paper, we discuss these issues and propose a different approach to coordination that uses only the nondirectional $\multimap$ in the tectogrammar but augments the *phenogrammar* with separation-style subtyping in the manner of Lambek and Scott's higher order categorical logic. This allows the definition of directionally sensitive phenogrammatical functor subtypes which are the images of certain linear combinators called *phenominators*, restoring the ability to distinguish between leftward and rightward-looking functors, among others. The resulting framework is called LCG_ϕ (LCG with phenogrammatical subtyping). The paper includes a comparison of the empirical consequences of HTLG and LCG_ϕ with respect to functor coordination.

Keywords: subtype, phenominator, coordination, lambda calculus, linear logic, higher order logic

1 Introduction

Famously, categorial formalisms—such as Lambek grammar (L, Lambek 1958) and combinatory categorial grammar (CCG, Steedman (1985) *inter multa alia*)—whose sole connectives are the directional implications / and \ are well-equipped to analyze functor coordination. This is because, generally speaking, coordinated functors are constrained to 'seek their arguments in the same direction'. So as long as the coordinators are assigned

*We are grateful to Manjuan Duan, Yusuke Kubota, Bob Levine, Gerald Penn, the CG2015 committee, and three anonymous reviewers for their comments. Any errors or misunderstandings rest solely on the shoulders of the authors.

the schematic lexical type $(X\backslash X)/X$, coordinated functors are guaranteed to observe that constraint: the two argument occurrences of X have to be instantiated by one and the same type (both $A\backslash B$ or else both B/A, for fixed values of A and B). For example, even though the right-node-raising remnant type S/NP and the 'VP' type NP\S both take NP arguments, return S values, and correspond to properties in the semantics, they don't count as the same type; the directionality of functors is part and parcel of their categorial identity.

On the other hand, these same formalisms are notoriously ill-equipped to handle medial cases of extraction and quantifier raising (QR) because they lack the resources to express the notion of an B that has a A missing from some position or other—possibly a nonperipheral one—somewhere inside it. Indeed, there is an abundance of formalisms that extend L with additional connectives which, among other things, provide the greater expressivity required to handle such cases, such as multimodal type-logical grammar (MMTLG, Moortgat 1997), displacement calculus (D, Morrill et al. 2011), and NL_λ (Barker and Shan, 2014).

Apart from the family of extended Lambek grammars, which can be reasonably considered to constitute the main stream of categorial grammar research, there is a second stream of categorial thought which takes its inspiration from a programmatic—and in places cryptic—paper by Curry (1961), itself a revised and extended version of a lecture presented in 1948 to the 'comparative literature group' at (then) Pennsylvania State College. In that paper, Curry insisted on drawing a distinction between 'two different levels of grammar' : **tectogrammatics** (also known as as abstract syntax, tectogrammar, or simply tecto), and **phenogrammatics** (also known as concrete syntax, phenogrammar, or simply pheno). According to Curry, tectogrammatics was concerned with how phrases combine, as determined by the (tecto) types to which they belonged. Curry's basic types were **sams** and **tettles** (essentially, NP and S respectively), and the nonbasic, or **functor** types were formed with the help of a single **functionality primitive** (in more up-to-date parlance, a type constructor) F, such that a functor of type $\mathsf{F}XY$ combined with an X argument to form a Y. Phenogrammatics, on the other hand, was concerned not with abstract functor-argument structure, but rather with the **expressions** used to represent them. In the case of basic types, these expressions would be strings of symbols (where a word is thought of as a single symbol). But for functors, the corresponding expressions were to be *ways of combining expressions.* These functorial expressions couldn't be described merely in terms of symbols, but also required ancillary devices—Curry used blanks with numerical subscripts—to indicate where the argument expressions were to be inserted in the result expression (though he noted that '[a] functor may, conceivably, so modify its arguments that even the notations involving blanks are inadequate to describe it'). In his concluding remarks, Curry commented on Lambek's (1958)

> recently proposed ... calculus of grammatical structure based on two kinds of functionality notions expressed by slant lines. Thus N/S would mean a functor forming a noun from a sentential argument on its right, while N\S would mean a functor forming a sentence from a nominal argument on the left. This is a classification of expressions in a concatenative grammar rather than a classification of functors .... If we use 'f' as an abbreviation for the expression, then Lambek's " 'f' is an N/S" would mean the same as " 'f_{-1}' is an FSN" whereas his " 'f is an N\S" would mean the same as my " '$\_{-1}f$ is an FNS". Thus Lambek's conception has an admixture of phenogrammatics. Moreover, it seems to break down completely with respect to functors which are not either prefixes or suffixes.

The second, 'curryesque', stream of categorial grammar research begins with a kind of grammar proposed by Oehrle (1994). Leaving semantics aside (it is easy to add it back in, but we won't need to do that in this paper), the essence of Oehrle's proposal can be summarized by the following two points:

1. Curry's tectogrammatical types can be identified with formulas of implicative linear logic, and his functionality primitive F with the linear implication $\multimap$; and

2. Curry's expressions (including the complicated functorial ones that he couldn't describe with blanks) can be identified with terms of the linear lambda calculus, with constants of type s (string) in place of symbols.

Thus, in Oehrle's system, what Curry would have written as $\mathsf{F}(\mathsf{F}\ \mathrm{N}\ \mathrm{S})\mathrm{S}$ becomes $(\mathrm{NP} \multimap \mathrm{S}) \multimap \mathrm{S}$; and what Curry would have written as $\__1\mathrm{ate}\__2$ becomes $\lambda_{st}.s \cdot \mathrm{ate} \cdot t$.

Oehrle did not pursue this line of work, but it was subsequently picked up by others, most notably de Groote (2001) and Muskens (2001, 2003, 2007), who elaborated it into the frameworks of abstract categorial grammar (ACG) and lambda grammar (λG) respectively. As far as we know, the λG framework was not further developed after 2007. ACG, on the other hand, has been intensively investigated since its inception, but the emphasis in this line of work has been on the mathematical and computational properties of the formalism. Unlike the situation with respect to Lambek grammar and its extensions, there has been little consideration, to say nothing of appreciation, of the potential for curryesque grammars to serve as the basis for detailed linguistic analysis and theorization. The present paper, which seeks a curryesque analysis of coordination, is part of a larger project aimed at exploiting that potential by developing a practical linguistic research framework, linear categorial grammar (LCG), that maintains a strict separation between phenogrammatics and tectogrammatics (Duan 2015; Martin 2013, 2015; Martin and Pollard 2014; Mihaliček 2012; Mihaliček and Pollard 2012; Pollard 2015; Pollard and Smith 2012; Smith 2010; Worth 2014).[1]

As we noted, L copes admirably with coordination but falters on medial extraction and QR. For curryesque grammars, though, the situation is just the reverse. Medial extraction and QR are unproblematic because the nondirectional implicative type $A \multimap B$ captures precisely the notion of a B missing an A. Thus, e.g., a quantificational noun phrase (QP) such as *someone* or a fronted NP such as *bagels* with a contrastive-topicalization intonation can both be analyzed, tectogrammatically, as $(\mathrm{NP} \multimap \mathrm{S}) \multimap \mathrm{S}$; the difference between quantifier scoping and topicalization that mainstream generative grammar analyzes in terms of covert vs. overt movement is succinctly captured at the pheno level ($\lambda_f.f$ someone vs. $\lambda_f.\mathrm{bagels} \cdot (f\ \mathbf{e})$).[2] But, as noted by e.g. Kubota (2010) and Moot (2014), coordination of signs whose pheno components are functions rather than strings is, on the face of it, unanalyzable, because the absence of directional implications entails that there is no straightforward (tecto) type assignment for the coordinators that imposes a like-directionality constraint on the conjuncts.

One way to cope with the inability of curryesque CG to analyze coordination is to simply combine it with L, as advocated in a series of recent papers by Kubota and Levine (2012, 2013, 2015; Kubota 2015). The resulting framework, aptly dubbed hybrid type-logical categorial grammar (HTLCG, here shortened to HTLG) by its creators, employs

[1] In some of these works, the emerging framework is referred to as LinCG, LG (linear grammar), or PTDCG (pheno-tecto-differentiated categorial grammar).

[2] Here f is a variable of type $\mathrm{s} \to \mathrm{s}$ (functions from strings to strings), $\cdot$ denotes concatenation (an infixed constant of type $\mathrm{s} \to \mathrm{s} \to \mathrm{s}$), and $\mathbf{e}$ is a string constant corresponding to the null string.

both nondirectional[3] and directional implications, as well as Oehrle-style phenogrammatical lambda terms. The use of pheno terms sets HTLG apart from the other extended-Lambek formalisms; at the same time, the use of directional slashes, with their 'admixture of phenogrammatics', amounts to an abandonment, if not a repudiation, of the curryesque program. It could well turn out that Kubota and Levine are on the right track, that the idea of a grammar architecture based on a sharp bifurcation of abstract vs. concrete syntax was simply wrong from the start. But we think it is illuminating to put some effort into trying to salvage that idea.

In this paper, we will propose an alternative to HTLG which preserves the nondirectional character of the tecto component, by *making the phenotypes more fine-grained*. More specifically, we will classify pheno functors by articulating the usual phenogrammatical functional types into *subtypes* in terms of how, concretely, they manipulate the pheno components of their arguments. In Curry's original notation, this means that, e.g. *Kim likes* and *likes Sandy*, while sharing the tectotype F NP S, are of different *phenogrammatical* subtypes, because the former is a prefix and the latter a suffix. Likewise, *someone* and (topicalized) *bagels* share the tectotype F (F NP S) S, but are of different phenogrammatical subtypes because the former *inserts* a string (here, someone) into the blank of the second argument, while the latter inserts the null string into the blank of the second argument and prepends a string (here, bagels) to the result. All of this can be expressed more precisely and succinctly with linear lambda terms, and, luckily for us, Lambek and Scott (1986) have conveniently defined a scheme of subtyping for higher-order logic that simplifies the task of articulating phenotypes into fine-grained subtypes.

The rest of the paper is structured as follows. Section 2 sketches the formalism of linear categorial grammar (LCG). Section 3 introduces the formal machinery of *phenominators* and uses them to clarify the nature of LCG's problem with coordination. In section 4, we use phenominators to extend LCG with *fine-grained* phenotypes; the extended framework is called LCG_ϕ. Section 5 develops a lexical coordination schema in LCG_ϕ that analyzes a wide range of coordinations where the phenos of the disjuncts are not strings.[4] Section 6 provides a detailed analysis of a seemingly (but not really) problematic case, namely *argument cluster coordination*. Section 7 discusses relevant aspects of HTLG and compares its analysis of coordination to our own. Section 8 concludes by noting some remaining issues to be addressed in future work.

2 The LCG framework

Like ACG and λG, LCG is inspired by Curry's (1961) insistence on distinguishing pheno and tecto dimensions of syntax, and follows Oehrle (1994) in using linear logic rather than Lambek calculus or one of its elaborations in the tecto component. (Of course LCG signs also have a semantic component, omitted in this paper.) Also following Oehrle, LCG allows as phenos of signs not just strings but also (possibly higher-order) functions over strings, which enable straightforward analyses not only of quantifier scope and medial extraction (as already shown by Oehrle (1994) and Muskens (2007) respectively) but also, *inter alia*, correlational comparatives (Smith, 2010), second-position clitics (Mihaliček, 2012), *wh*-in situ and Baker ambiguities (Mihaliček and Pollard, 2012), parasitic scope (Pollard and Smith, 2012), alternative questions (Duan, 2015), and supplemental updates (Martin, 2015). Unlike ACG, LCG makes no use of tecto terms, so an LCG sign consists of a

[3] In HTLG papers, $A \multimap B$ is written $B|A$; in this paper, we stick with $\multimap$.

[4] Throughout, for the sake of expository simplicity, we limit attention to coordination of two signs with the same tectotype.

typed pheno term, a tectotype (linear logic formula), and a typed semantic term (omitted here). Also unlike ACG, the grammar architecture is parallel rather than bifunctorial (i.e. there are no (categorical) functors from tecto to pheno and semantics respectively), so the formalism does not constrain which combinations of phenotype, tectotype, and semantic type can co-occur in a sign.

A **linear categorial grammar (LCG)** has two kinds of axioms: **traces** and **lexical entries**, and two inference rule schemata: **modus ponens (MP)** and **hypothetical proof (HP)**. Traces are multidimensional counterparts of the usual logical axiom schema $B \vdash B$, of the form:

$$p : A; B \vdash p : A; B$$

while lexical entries[5] are null-context nonlogical axioms such as the following[6]:

$$\vdash \text{pedro} : \text{s}; \text{NP}$$

$$\vdash \lambda_s.s \cdot \text{brayed} : \text{s} \to \text{s}; \text{VP}$$

$$\vdash \lambda_{st}.t \cdot \text{beat} \cdot s : \text{s} \to \text{s} \to \text{s}; \text{TV}$$

$$\vdash \lambda_{stu}.u \cdot \text{gave} \cdot s \cdot t : \text{s} \to \text{s} \to \text{s} \to \text{s}; \text{DV}$$

$$\vdash \lambda_f.f \text{ someone} : (\text{s} \to \text{s}) \to \text{s}; \text{QP}$$

$$\vdash \lambda_f.\text{who} \cdot (f\ \mathbf{e}) : (\text{s} \to \text{s}) \to \text{s}; \text{WP}$$[7]

MP is $\multimap$-elimination in the tecto component and function application in the other components; HP is $\multimap$-introduction in the tecto component and lambda abstraction in the other components:

$$\frac{\Gamma \vdash f; A \multimap B \qquad \Delta \vdash a; A}{\Gamma, \Delta \vdash f\ a; B}\ \text{MP}$$

$$\frac{\Gamma, p; A \vdash a; B}{\Gamma \vdash \lambda_p.a; A \multimap B}\ \text{HP}$$

LCG analysis trees are in Gentzen-sequent-style natural deduction format, with contexts in sequents being (possibly empty) finite sets of hypothetical signs (i.e. their pheno and semantic terms are variables), which can be thought of as specifying the unbound traces at each node. More specifically, an **analysis tree** is an ordered tree with every node labelled by a sequent, such that (1) every leaf node is labelled by an axiom, and (2) for every nonleaf node, there is an inference rule whose conclusion labels that node and whose premiss(es) label(s) that node's daughter(s). (So in an analysis trees, all binary branching nodes are instances of MP whose left daughter is the major premiss, and all unary branching nodes are instances of HP.) A sign is **generated** by a grammar provided there is an analysis tree whose root is labelled by a null-context sequent with that sign as its statement, e.g. the following analysis tree shows that the sign $\text{pedro} \cdot \text{beat} \cdot \text{chiquita}; \text{S}$ is generated by the current grammar

[5]For the sake of familiarity, we follow the widespread convention in categorial grammar of treating the first postverbal complement (if any) as the first argument of a verb lexical entry and the subject as the last argument. This choice has no theoretical consequences; in fact,all other argument orders are derivable.

[6]We adopt the following abbreviations for tectotypes: $\text{VP} =_{\text{def}} \text{NP} \multimap \text{S}$; $\text{TV} =_{\text{def}} \text{NP} \multimap \text{VP} = \text{NP} \multimap \text{NP} \multimap \text{S}$; $\text{DV} =_{\text{def}} \text{NP} \multimap \text{TV} = \text{NP} \multimap \text{NP} \multimap \text{NP} \multimap \text{S}$; $\text{QP} =_{\text{def}} \text{VP} \multimap \text{S} = (\text{NP} \multimap \text{S}) \multimap \text{S}$ (the raised type for quantificational noun phrases); $\bar{\text{Q}}$ (embedded interrogative clause); and $\text{WP} =_{\text{def}} \text{VP} \multimap \bar{\text{Q}} = (\text{NP} \multimap \text{S}) \multimap \bar{\text{Q}}$ (the type for 'overtly moved' *wh*-phrases in embedded *wh*-interrogative clauses). Types of pheno terms are often omitted if they can be inferred from the types of the constants and variables.

[7]We consider only *embedded wh*-questions, in order to sidestep irrelevant complications arising from *wh*-auxiliary inversion.

$$\dfrac{\dfrac{\vdash \lambda_{st}.t \cdot \text{beat} \cdot s; \text{TV} \qquad \vdash \text{chiquita}; \text{NP}}{\vdash \lambda_t.t \cdot \text{beat} \cdot \text{chiquita}; \text{VP}} \qquad \vdash \text{pedro}; \text{NP}}{\vdash \text{pedro} \cdot \text{beat} \cdot \text{chiquita}; \text{S}}$$

3 LCG's problem with coordination

LCG pheno terms (hereafter, phenos) are terms in the higher order theory with one basic type m (other than the truth value type t supplied by the underlying logic). Using the standard encoding of the theory of monoids, we model strings with the type $\text{s} =_{\text{def}} \text{m} \to \text{m}$. Metavariables over variables of type s are s, t, u, and v; and metavariables over variables of type $\text{s} \to \text{s}$ are f and g. We write e (null string) as an abbreviation for $\lambda_m.m : \text{s}$, and $\cdot$ (concatenation, written infix) as an abbreviation for $\lambda_{stm}.s\ (t\ m) : \text{s} \to \text{s} \to \text{s}$; hence the monoid axioms (that concatenation is associative and that the null string is a two-sided identity for concatenation) are derivable. Additionally, we have nonlogical constants of type s that represent (length-one strings of) phonological words, e.g. pedro, brayed, etc. By an **LCG phenotype**, we mean either s or else a functional type over s. In LCG as it stands, phenos of signs are always terms with an LCG phenotype.

To explain why coordination is problematic for LCG, we must define some terms. First, a closed pheno term is a **(pheno) combinator** iff it contains no nonlogical constants.[8] A combinator is **linear** iff every lambda binds exactly one variable instance. An A-**phenominator** is a linear combinator whose type is of the form $\text{s} \to A$. In an interpretation I, we think of each string constant as denoting a string of phonological words of length one, and each A-phenominator denotes an injective function from the set of strings into the set of phenos $I(A)$. For each phenominator ϕ and each string s, we call s the **string support** of $(\phi\ \mathsf{s})$. Abbreviations for some much-used phenominators are:

$\mathsf{n} =_{\text{def}} \lambda_v.v$ (name)

$\mathsf{i} =_{\text{def}} \lambda_{vs}.s \cdot v$ (intransitive verb)

$\mathsf{r} =_{\text{def}} \lambda_{vs}.v \cdot s$ (right node raising)

$\mathsf{t} =_{\text{def}} \lambda_{vst}.t \cdot v \cdot s$ (transitive verb)

$\mathsf{d} =_{\text{def}} \lambda_{vstu}.u \cdot v \cdot s \cdot t$ (ditransitive verb)

$\mathsf{q} =_{\text{def}} \lambda_{vf}.f\ v$ (quantificational noun phrase)

$\mathsf{w} =_{\text{def}} \lambda_{vf}.v \cdot f\ \mathsf{e}$ (wh-noun phrase)

Note, for example, that in the lexical entries above, the pheno for a name results from applying n to a string, the pheno for an intransitive verb results from applying i to a string, etc. We can make such facts explicit by rewriting lexical entries with their pheno terms in pre-β-reduced forms:

$\vdash \mathsf{n}$ pedro : s; NP

$\vdash \mathsf{i}$ brayed : $\text{s} \to$ s; VP

$\vdash \mathsf{t}$ beat : $\text{s} \to \text{s} \to$ s; TV

$\vdash \mathsf{d}$ gave : $\text{s} \to \text{s} \to \text{s} \to$ s; DV

$\vdash \mathsf{q}$ someone : $(\text{s} \to \text{s}) \to$ s; QP

[8] According to this definition, a pheno combinator could conceivably contain *logical* constants, e.g. $=$, $\exists$, $\neg$, etc., though in practice, LCG pheno terms have not made recourse to these. But in LCG_ϕ, introduced in section 4, pheno terms must be allowed to contain $\uparrow$-terms that denote canonical injections, and these are defined in terms of $\exists$ and $=$.

$\vdash \mathsf{w}$ who : $(\mathrm{s} \to \mathrm{s}) \to \mathrm{s}; \mathrm{WP}$

Note also that phenos with the same LCG phenotype can lie in the image of different phenominators; for example, the phenos of *someone* and *who* are both of LCG phenotype $(\mathrm{s} \to \mathrm{s}) \to \mathrm{s}$, but *someone* is in the image of q while *who* is in the image of w. For another example: the pheno of the constituent *pedro beat* in the analysis tree above has LCG phenotype $\mathrm{s} \to \mathrm{s}$, the same as an intransitive verb; but the phenominator in whose image it lies is not i but rather r. The relevance of these observations is that *constituents with nonstring phenos which lie in the image of different phenominators cannot in general be coordinated*.[9] For example, VP-coordination (both conjunct phenos in the image of i) and right-node raising (both conjuncts in the image of r) are fine, but the mixed cases are not:

(1) a. Chiquita [brayed and grazed].
b. [Pedro beat, and Maria patted], Chiquita.
c. *Chiquita [brayed and Maria patted].
d. *[brayed and Maria patted] Chiquita.

LCG as it stands cannot block such ungrammatical coordinations because the conjunct signs' LCG phenotypes are insensitive to which phenominators' images the respective phenos lie within. We propose to solve this problem by making use, in a highly constrained way, of *separation-style subtyping*, explained in the following section. The basic idea is to provide a mechanism for identifying *fine-grained* phenotypes corresponding to the images of phenominators. For example, an intransitive verb pheno will have the fine-grained phenotype $(\mathrm{s} \to \mathrm{s})_{\mathsf{i}}$, the subtype of $(\mathrm{s} \to \mathrm{s})$ consisting of phenos of the form $(\mathsf{i}\ s)$ for some string s, whereas a conjunct in a right-node raising sentence will have a pheno of the fine-grained phenotype $(\mathrm{s} \to \mathrm{s})_{\mathsf{r}}$ consisting of phenos of the form $(\mathsf{r}\ s)$. Then the lexical entry for *and* will require of its conjunct arguments that they have the same fine-grained phenotype.

4 LCG with Fine-Grained Phenotypes (LCG_ϕ)

To add fine-grained phenotypes to LCG, we make use of a scheme of **separation-style subtyping** for higher-order logic (HOL), essentially following Lambek and Scott (1986) (though our notation is somewhat different). The basic idea is to define subtypes by analogy with the way that the Axiom of Separation is used in set theory to define the subset of a given set determined by by a given predicate. To be more precise, we can define the set of types in a higher-order theory as follows:

a. The basic types supplied by HOL, called the **logical** basic types, namely t (the type of truth values) and T (the unit type), are types.

b. The additional basic types specified by the theory, called the **nonlogical** basic types, are types.

c. If A and B are types, then so is $A \to B$.

d. If A is a type and P an A-predicate (i.e. a term of type $A \to \mathrm{t}$), then $[x : A | P\ x]$ is a type, called the **subtype** of A **determined by** P.

[9] However, such coordinations are sometimes judged acceptable when the resulting coordination itself becomes an argument rather than a functor, e.g. *donkey that loves Maria but Pedro mistreats*, where the coordinate structure becomes an argument to the relativizer *that*. For now we lack an analysis of such cases.

e. Nothing else is a type.

Moreover, there is a term constructor ker (read 'kernel') such that, for any A-predicate P, ker_P is a term of type $[x : A|P\ x] \to A$; and there are axiom schemas which say of these terms that they denote injective functions. In an interpretation, $[x : A|P\ x]$ denotes the subset of the set denoted by A whose characteristic function is the function denoted by P, and ker_P denotes the canonical embedding of the subset into the superset.

We now specialize to the case where the higher-order theory in question has the single nonlogical basic type m, and s (the string type), $\cdot$ (concatenation), and $\mathbf{e}$ (null string) are defined as before. As before, we define an A-**phenominator** to be a linear combinator of type $\mathrm{s} \to A$. For any A-phenominator ϕ, we denote by P_ϕ the A-predicate $\lambda_{p:A}.\exists_{s:\mathrm{s}}.p = (\phi\ s)$. Also, we denote by A_ϕ the subtype $[p : A|P_\phi\ p]$; and we denote by $\uparrow_\phi$ (read 'up sub ϕ') the canonical embedding $\mathsf{ker}_{P_\phi} : A_\phi \to A$.

Then. from among the set of types defined as immediately above, we single out a set of types called the **LCG$_\phi$ phenotypes** as follows:

a. s is an LCG$_\phi$ phenotype.

b. If A and B are LCG$_\phi$ phenotypes, then so is $A \to B$.

c. If A is an LCG$_\phi$ phenotype and ϕ an A-phenominator, then A_ϕ is an LCG$_\phi$ phenotype.

d. Nothing else is an LCG$_\phi$ phenotype.

By a **fine-grained phenotype**, we mean an LCG$_\phi$ phenotype of the form A_ϕ. As a special case, s itself is identified with the fine-grained phenotype $\mathrm{s}_{\lambda_s.s}$. If A_ϕ is a fine-grained phenotype and $a : A$ a term such that $\vdash P_\phi\ a$ (where the $\vdash$ here means provability in the higher-order pheno theory), then we denote by $(\downarrow_\phi\ \ a)$ (read 'down sub ϕ of a') the unique element of A_ϕ whose embedded image in A is a. As a consequence of these definitions and conventions:

a. $\vdash \forall_{x:A_\phi}.(\downarrow_\phi\ \ (\uparrow_\phi\ \ p)) = p$

b. $\vdash \forall_{p:A}.(P_\phi\ p) \to (\uparrow_\phi\ \ (\downarrow_\phi\ \ p)) = p$

To integrate the enriched system of phenotypes into the grammatical framework, we first have to extend the schemata MP and HP so that the types of the pheno terms are allowed to range over LCG$_\phi$-phenotypes. And second, we need to "teach" LCG about fine-grained phenotypes, by adding rules for $\uparrow$ and $\downarrow$:

$$\frac{\Gamma \vdash a : A_\phi; B}{\Gamma \vdash (\uparrow_\phi a) : A; B}\uparrow$$

$$\frac{\Gamma \vdash a : A; B \qquad \Gamma' \vdash P_\phi\ a}{\Gamma \vdash (\downarrow_\phi a) : A_\phi; B}\downarrow$$

In the second premiss of the $\downarrow$ rule, Γ' is the projection of Γ onto the pheno component, and $\vdash$ is provability in the higher-order pheno theory, not grammatical derivability; this premiss can be thought of as a side condition on the first premiss.

Then our solution to LCG's coordination problem will be a lexical entry schema for *and* of the following form:[10]

$$\vdash \lambda_{p:A_\phi}.\lambda_{q:A_\phi}.\downarrow_\phi\ \ (\phi\ ((\mathsf{say}_{A_\phi}\ p) \cdot \mathrm{and} \cdot (\mathsf{say}_{A_\phi}\ q))) : A_\phi \to A_\phi \to A_\phi; B \multimap B \multimap B$$

[10]The simple-minded tectotype used here won't generalize to conjuncts with unlike tectotypes. For that, as argued by Worth (in progress), we need the tectotype $\bigcirc B \multimap \bigcirc B \multimap \bigcirc B$, where $\bigcirc$ is a strong monad.

schematized over fine-grained phenotypes A_ϕ and arbitrary tectotypes B. Here say_{A_ϕ}, as explained in the following section, is a function which, when applied to a pheno with a fine-grained phenotype, extracts the pheno's string support. This allows coordination only for conjuncts with the same fine-grained phenotype, i.e. whose phenos lie in the image of the same phenominator ϕ. In English prose: the pheno of the coordination of two conjuncts with fine-grained phenotype A_ϕ is obtained by first applying the phenominator ϕ to the string obtained by concatenating together (1) the string support of the first conjunct, (2) *and*, and (3) the string support of the second conjunct; and then taking the preimage of the result with respect to the canonical embedding of A_ϕ into A.

5 Coordination in LCG_ϕ

To complete the formulation of the coordination schematic lexical entry in the preceding section, we first have to define the say function, or, more precisely, the family of say_A functions, where A ranges over a certain set of LCG_ϕ-phenotypes called (unsurprisingly) the *sayable* phenotypes. The basic intuition here is: to 'say' a pheno functor, supply it with "vacuous' arguments and if the result is a string, that's how you say it; but if the result is not a string, 'say' the result. Eventually this process will bottom out at a string. In order to do this, we will need to be amply supplied with 'vacuous' terms, called *vacuities*, each of which plays the role with respect to the type it belongs to that is played by the null string $\mathbf{e}$ with respect to the type s. Some delicacy is called for here, since not all phenotypes *have* vacuities; those that do are called *vacuable*. So we need to say what a vacuable phenotype is.

To that end, we first define the set $\{\mathrm{s}_n | n \leq 0\}$ of **left-associative** phenotypes[11] as follows:

a. $\mathrm{s}_0 =_{\mathrm{def}} \mathrm{s}$

b. $\mathrm{s}_{-n-1} =_{\mathrm{def}} (\mathrm{s}_{-n}) \to s$ $(n \geq 0)$

Thus, e.g., we have:

$\mathrm{s}_3 =_{\mathrm{def}} \mathrm{s} \to \mathrm{s} \to \mathrm{s} \to \mathrm{s}$

$\mathrm{s}_2 =_{\mathrm{def}} \mathrm{s} \to \mathrm{s} \to \mathrm{s}$

$\mathrm{s}_1 =_{\mathrm{def}} \mathrm{s} \to \mathrm{s}$

$\mathrm{s}_0 =_{\mathrm{def}} \mathrm{s}$

$\mathrm{s}_{-1} =_{\mathrm{def}} \mathrm{s} \to \mathrm{s}$

$\mathrm{s}_{-2} =_{\mathrm{def}} (\mathrm{s} \to \mathrm{s}) \to \mathrm{s}$

$\mathrm{s}_{-3} =_{\mathrm{def}} ((\mathrm{s} \to \mathrm{s}) \to \mathrm{s}) \to \mathrm{s}$

Then we define a phenotype to be **vacuable** iff it is either left-associative or fine-grained.

The set of vacuities vac_A, schematized by the vacuable types, is defined as follows.

a. For $A = B_\phi$ fine-grained, $\mathsf{vac}_A =_{\mathrm{def}\downarrow_\phi} (\phi\ \mathbf{e})$.

b. For A left-associative:

i. $\mathsf{vac}_{\mathrm{s}_0} =_{\mathrm{def}} \mathbf{e}$

ii. $\mathsf{vac}_{\mathrm{s}_{-1}} =_{\mathrm{def}} \lambda_s.s$

[11] Note that the left-associative phenotypes are abbreviated by s with a nonpositive numerical subscript. Positive subscripts are reserved for **right-associative types**, i.e. $\mathrm{s}_{n+1} =_{\mathrm{def}} s \to \mathrm{s}_n$ for all $n \geq 0$.

iii. $\mathsf{vac}_{\mathrm{s}_{-n-2}} =_{\text{def}} \lambda_{h:\mathrm{s}_{-n-1}}.h\ \mathsf{vac}_{\mathrm{s}_{-n}}$

For example, as the reader can easily verify:

$\mathsf{vac}_{\mathrm{s}} = \mathbf{e}$

$\mathsf{vac}_{\mathrm{s}\to\mathrm{s}} = \lambda_s.s$

$\mathsf{vac}_{(\mathrm{s}\to\mathrm{s})\to\mathrm{s}} = \lambda_f.f\ \mathbf{e}$

$\mathsf{vac}_{((\mathrm{s}\to\mathrm{s})\to\mathrm{s})\to\mathrm{s}} = \lambda_{p:(\mathrm{s}\to\mathrm{s})\to\mathrm{s}}.p\ \lambda_s.s$

Then the **sayable** phenotypes are defined as follows:

a. s is sayable.

b. Any fine-grained phenotype A_ϕ is sayable if A is sayable.

c. If A is vacuable and B is sayable, then $A \to B$ is sayable.

c. Nothing else is sayable.

For example, the codomain types for all the phenominators listed above are easily verified to be sayable:

$\mathsf{n} : \mathrm{s} \to \mathrm{s}$

$\mathsf{i}, \mathsf{r} : \mathrm{s} \to \mathrm{s}_1$

$\mathsf{t} : \mathrm{s} \to \mathrm{s}_2$

$\mathsf{d} : \mathrm{s} \to \mathrm{s}_3$

$\mathsf{q}, \mathsf{w} : \mathrm{s} \to \mathrm{s}_{-2}$

Finally, the say_A functions are defined as follows:

a. $\mathsf{say}_{\mathrm{s}} =_{\text{def}} \lambda_s.s$

b. $\mathsf{say}_{A_\phi} =_{\text{def}} \lambda_{p:A_\phi}.\ \mathsf{say}_A\ \ (\uparrow_\phi\ \ p)$, for A_ϕ fine-grained and A sayable.

c. $\mathsf{say}_{A\to B} =_{\text{def}} \lambda_{h:A\to B}.\ \mathsf{say}_B\ \ (h\ \mathsf{vac}_A)$, for A vacuable and B sayable.

For example, as again can be easily verified, the say functions for the phenominator target types listed above are as follows:

$\mathsf{say}_{\mathrm{s}} = \lambda_s.s$

$\mathsf{say}_{\mathrm{s}_1} = \lambda_f.f\ \mathbf{e}$

$\mathsf{say}_{\mathrm{s}_2} = \lambda_{p:\mathrm{s}_2}.p\ \mathbf{e}\ \mathbf{e}$

$\mathsf{say}_{\mathrm{s}_3} = \lambda_{p:\mathrm{s}_3}.p\ \mathbf{e}\ \mathbf{e}\ \mathbf{e}$

$\mathsf{say}_{\mathrm{s}_{-2}} = \lambda_{p:\mathrm{s}_{-2}}.p\ (\lambda_s.s)$

It can be shown to be a consequence of the definitions that for any sayable type A, A-phenominator ϕ, and string s:

$$\vdash \mathsf{say}_A\ (\phi\ \mathsf{s}) = \mathsf{s}$$

i.e. say_A is a left inverse for every A-phenominator; and that for every $n \geq 0$,

$$\vdash \mathsf{say}_{\mathrm{s}_{-n}} = \mathsf{vac}_{\mathrm{s}_{-n-1}}$$

i.e. the say function for each left-associative phenotype is the same as the vac function of the next-lowest order.

We conclude this section by listing illustrative instances of the coordination schema, repeated here:

$$\vdash \lambda_{p:A_\phi}.\lambda_{q:A_\phi}.\downarrow_\phi\ (\phi\ ((\mathsf{say}_{A_\phi}\ p)\cdot \mathrm{and}\cdot (\mathsf{say}_{A_\phi}\ q))) : A_\phi \to A_\phi \to A_\phi; B \multimap B \multimap B$$

Here A ranges over sayable phenotypes, ϕ over A-phenominators, and B over arbitrary tectotypes. Then by suitably instantiating A, ϕ, and B, we obtain, inter multa alia, the following seven coordination rules (after lambda conversion, and with explicit phenotyping omitted):

a. NP coordination: $B = \mathrm{NP}$, $A = \mathrm{s}$, $\phi = \mathsf{n}$
   $\vdash \lambda_{st}.s\cdot \mathrm{and}\cdot t; \mathrm{NP} \multimap \mathrm{NP} \multimap \mathrm{NP}$

b. VP coordination: $B = \mathrm{VP}$, $A = \mathrm{s}_1$, $\phi = \mathsf{i}$
   $\vdash \lambda_{p:(\mathrm{s}_1)_\mathsf{i}}.\lambda_{q:(\mathrm{s}_1)_\mathsf{i}}.\downarrow_\mathsf{i}\ (\lambda_s.s\cdot(\uparrow_\mathsf{i}\ p\ \mathbf{e})\cdot \mathrm{and}\cdot(\uparrow_\mathsf{i}\ q\ \mathbf{e})); \mathrm{VP} \multimap \mathrm{VP} \multimap \mathrm{VP}$

c. Right node raising: $B = \mathrm{VP}$, $A = \mathrm{s}_1$, $\phi = \mathsf{r}$
   $\vdash \lambda_{p:(\mathrm{s}_1)_\mathsf{r}}.\lambda_{q:(\mathrm{s}_1)_\mathsf{r}}.\downarrow_\mathsf{r}\ (\lambda_s.(\uparrow_\mathsf{r}\ p\ \mathbf{e})\cdot \mathrm{and}\cdot(\uparrow_\mathsf{r}\ q\ \mathbf{e})\cdot s); \mathrm{VP} \multimap \mathrm{VP} \multimap \mathrm{VP}$

d. Transitive coordination: $B = \mathrm{TV}$, $A = \mathrm{s}_2$, $\phi = \mathsf{t}$
   $\vdash \lambda_{p:(\mathrm{s}_2)_\mathsf{t}}.\lambda_{q:(\mathrm{s}_2)_\mathsf{t}}.\downarrow_\mathsf{t}\ (\lambda_{st}.t\cdot(\uparrow_\mathsf{t}\ p\ \mathbf{e}\ \mathbf{e})\cdot \mathrm{and}\cdot(\uparrow_\mathsf{t}\ q\ \mathbf{e}\ \mathbf{e})\cdot s); \mathrm{TV} \multimap \mathrm{TV} \multimap \mathrm{TV}$

e. Ditransitive coordination: $B = \mathrm{DV}$, $A = \mathrm{s}_3$, $\phi = \mathsf{d}$
   $\vdash \lambda_{p:(\mathrm{s}_3)_\mathsf{d}}.\lambda_{q:(\mathrm{s}_3)_\mathsf{d}}.\downarrow_\mathsf{d}\ (\lambda_{stu}.u\cdot(\uparrow_\mathsf{d}\ p\ \mathbf{e}\ \mathbf{e}\ \mathbf{e})\cdot \mathrm{and}\cdot(\uparrow_\mathsf{d}\ q\ \mathbf{e}\ \mathbf{e}\ \mathbf{e})\cdot s\cdot t); \mathrm{DV} \multimap \mathrm{DV} \multimap \mathrm{DV}$

f. Quantificational NP coordination: $B = \mathrm{QP}$; $A = \mathrm{s}_{-2}$; $\phi = \mathsf{q}$
   $\vdash \lambda_{p:(\mathrm{s}_{-2})_\mathsf{q}}.(\lambda_{q:(\mathrm{s}_{-2})_\mathsf{q}}.\downarrow_\mathsf{q}\ \lambda_f.f\ ((\uparrow_\mathsf{q}\ p\ (\lambda_s.s))\cdot \mathrm{and}\cdot(\uparrow_\mathsf{q}\ q\ (\lambda_s.s)))); \mathrm{QP} \multimap \mathrm{QP} \multimap \mathrm{QP}$

g. *Wh*-NP coordination: $B = \mathrm{WP}$; $A = \mathrm{s}_{-2}$; $\phi = \mathsf{w}$
   $\vdash \lambda_{p:(\mathrm{s}_{-2})_\mathsf{w}}.\lambda_{q:(\mathrm{s}_{-2})_\mathsf{w}}.\downarrow_\mathsf{w}\ (\lambda_f.(\uparrow_\mathsf{w}\ p\ (\lambda_s.s))\cdot \mathrm{and}\cdot(\uparrow_\mathsf{w}\ q\ (\lambda_s.s)))\cdot(f\ \mathbf{e})); \mathrm{WP} \multimap \mathrm{WP} \multimap \mathrm{WP}$

6 Argument Cluster Coordination

6.1 Overview of the analysis

So-called *argument cluster coordination* (henceforth ACC) is occasioned when each conjunct of a coordinate structure appears to be a sequence of complements of the same verb. as in the following (where *Fido* and *Rover* are understood to be dogs):

(2) Kim gave [[Sandy Fido] and [Leslie Rover]].

This and similar examples posed a challenge for phrase-structural frameworks, since the expressions *Sandy Fido* and *Leslie Rover* do not form constituents in the traditional sense. Famously, categorial grammars with directional implications successfully analyzed such examples by treating the conjuncts as having category DV\VP, so that the coordinate structure, also of that type, forms a VP by combining with a DV to its left. In Dowty's (1988) CCG analysis, which cannot appeal to hypothetical proof, the argument clusters are formed by giving the indirect and direct objects the raised types DV\TV and TV\VP respectively and then combining them by right-to-left function composition (Dowty, 1988). Here VP, TV, and DV are defined not in terms of $\multimap$ as in LCG, but rather as the directional categories NP\S, (NP\S)/NP, and ((NP\S)/NP)/NP respectively. In the case of Lambek grammar (see e.g. Carpenter 1997), the clusters can be formed in a more straightforward fashion, by combining a hypothetical DV with its two

(unraised) NP objects and then withdrawing the hypothesis. Either way, directionality is centrally involved, because the complement clusters end up with the (directional) category DV\VP = (((NP\S)/NP)/NP)\(NP\S).

As a consequence, as with other cases of functor coordination, ACC is beyond the reach of curryesque frameworks unless they are augmented with some mechanism for making reference to direction of combination. In the case of HTLG, the Lambek-style analysis can be imported wholesale (as it is in Kubota and Levine 2013), since HTLG is an extension of Lambek grammar. In LCG_ϕ, though, where the tectotypes are all nondirectional, this path is not open to us. Instead, we will show that a complement cluster like *Sandy Fido* is not just any old DV $\multimap$ VP, but moreover has the crucial property that its phenogrammatical component lies in the image of a certain phenominator a, to be defined immediately below. It will follow from this that ACC can be straightforwardly analyzed by an instance of the coordination schema already proposed.

The a phenominator is more complex than any phenominator we have previously seen, since in ACC constructions, we must make crucial reference to the fact that the verb whose arguments the construction itself is comprised of must have a particular shape, namely, that of a ditransitive, a fact which is encoded in the subtype of the higher-order functional argument. We define a as follows:

$$\mathsf{a} =_{\text{def}} \lambda_{v:\text{s}}.\lambda_{p:(\text{s}_3)_\mathsf{d}}.\lambda_{s:\text{s}}.s \cdot (\mathsf{say}_{(\text{s}_3)_\mathsf{d}}\ p) \cdot v \text{ (argument cluster coordination)}$$

As an immediate consequence of the definition of say, we have

$$\mathsf{a} = \lambda_{v:\text{s}}.\lambda_{p:(\text{s}_3)_\mathsf{d}}.\lambda_{s:\text{s}}.s \cdot (\uparrow_\mathsf{d} p\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) \cdot v$$

6.2 Some lemmas for ACC

We find it both spatially and presentationally useful to prove lemmas for intuitively identifiable subparts of the full proof, which we then provide with numerical indices to be inserted in the full proof. We make use of this convention here for the two conjuncts in the coordinate structure (L1 and L2), the insertion of the lexical entry for *and* (L6), and for the subtyping of the verb (L5).

6.2.1 Formation of the conjunct clusters

First, we derive the desired structure for the first conjunct, the cluster *sandy fido*. The essential strategy is to hypothesize a ditransitive verb, apply it to its two arguments, withdraw the hypothesis, and then subtype according to the result, which will be under the argument cluster coordination phenominator a. Here we see the first practical instance of a phenominator which itself makes reference to fine-grained types, since a invokes the type $(\text{s}_3)_\mathsf{d}$. This will be essential, since we will need the eventual coordinate structure to select for a ditransitive verb specifically, and not a general ternary string function.

L1
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{h : (\text{s}_3)_\mathsf{d};\text{DV} \vdash h : (\text{s}_3)_\mathsf{d};\text{DV}}{h : (\text{s}_3)_\mathsf{d};\text{DV} \vdash \uparrow_\mathsf{d} h : \text{s}_3;\text{DV}}\uparrow \qquad \vdash \text{sandy} : \text{s};\text{NP}}{h : (\text{s}_3)_\mathsf{d};\text{DV} \vdash \uparrow_\mathsf{d} h\ \text{sandy} : \text{s}_2;\text{TV}} \qquad \vdash \text{fido} : \text{s};\text{NP}}{h : (\text{s}_3)_\mathsf{d};\text{DV} \vdash \uparrow_\mathsf{d} h\ \text{sandy fido} : \text{s}_1;\text{VP}}}{\vdash \lambda_{h:(\text{s}_3)_\mathsf{d}}.\uparrow_\mathsf{d} h\ \text{sandy fido} : (\text{s}_3)_\mathsf{d} \to \text{s}_1;\text{DV} \multimap \text{VP}} \qquad \vdash P_\mathsf{a}\ (\lambda_{h:(\text{s}_3)_\mathsf{d}}.\uparrow_\mathsf{d} h\ \text{sandy fido})}{\vdash \downarrow_\mathsf{a} (\lambda_{h:(\text{s}_3)_\mathsf{d}}.\uparrow_\mathsf{d} h\ \text{sandy fido}) : ((\text{s}_3)_\mathsf{d} \to \text{s}_1)_\mathsf{a};\text{DV} \multimap \text{VP}}\downarrow$$

L2 $\vdash \downarrow_\mathsf{a} (\lambda_{h:(\text{s}_3)_\mathsf{d}}.\uparrow_\mathsf{d} h\ \text{leslie rover}) : ((\text{s}_3)_\mathsf{d} \to \text{s}_1)_\mathsf{a};\text{DV} \multimap \text{VP}$

The formation of the second conjunct, the cluster *leslie rover*, proceeds exactly as the derivation given in L1, changing the names of the constants sandy and fido to leslie and rover, respectively.

6.2.2 Subtyping justification for the conjuncts

In order to coordinate the clusters in question, we need to show that they do in fact lie in the image of the relevant phenominator, namely a. In particular, we want to show that

$$\text{L3} \quad \vdash P_{\mathsf{a}}\ (\lambda_{h:(\mathrm{s}_3)_{\mathsf{d}}}.\ \uparrow_{\mathsf{d}} h \text{ sandy fido})$$

Suppose that we have $h : (\mathrm{s}_3)_{\mathsf{d}}$. Now, since $h = (\downarrow_{\mathsf{d}} (\uparrow_{\mathsf{d}} h))$, it is easy to see that $\uparrow_{\mathsf{d}} h$ is in the image of d. To put it another way, P_{d} holds of $\uparrow_{\mathsf{d}} h$. So by the definition of P_{d},

$$\vdash \exists_{v:\mathrm{s}}.\ \uparrow_{\mathsf{d}} h = \mathsf{d}\ v.$$

So we can ask, what is v? By definition, it is the string support of $\uparrow_{\mathsf{d}} h$, and we can therefore obtain it by taking $\mathsf{say}_{\mathrm{s}_3}(\uparrow_{\mathsf{d}} h)$. After working through the definition of say, we are left with the support $(\uparrow_{\mathsf{d}} h\ \mathsf{e}\ \mathsf{e}\ \mathsf{e})$. So $v = (\uparrow_{\mathsf{d}} h\ \mathsf{e}\ \mathsf{e}\ \mathsf{e})$, and thus

$$\vdash \mathsf{d}\ v = \mathsf{d}\ (\uparrow_{\mathsf{d}}\ h\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) = \lambda_{stu:\mathrm{s}}.u \cdot (\uparrow_{\mathsf{d}} h\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) \cdot s \cdot t.$$

Now, if we apply the injection of h to the two arguments we wish to create the cluster from, we can see that

$$\vdash \uparrow_{\mathsf{d}} h \text{ sandy fido} = \lambda_{stu:\mathrm{s}}.u \cdot (\uparrow_{\mathsf{d}} h\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) \cdot s \cdot t \text{ sandy fido}.$$

After reduction, we are left with the following:

$$\vdash \uparrow_{\mathsf{d}} h \text{ sandy fido} = \lambda_{u:\mathrm{s}}.u \cdot (\uparrow_{\mathsf{d}} h\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) \cdot \text{sandy} \cdot \text{fido}.$$

Furthermore, by binding our original hypothesis h in each side, we see that

$$\vdash \lambda_{h:(\mathrm{s}_3)_{\mathsf{d}}}.\ \uparrow_{\mathsf{d}} h \text{ sandy fido} = \lambda_{h:(\mathrm{s}_3)_{\mathsf{d}}}.\lambda_{u:\mathrm{s}}.u \cdot (\uparrow_{\mathsf{d}} h\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) \cdot \text{sandy} \cdot \text{fido}.$$

So now we are in a position to ask the question we were originally curious about with respect to ACC subtyping: does P_{a} hold of $\lambda_{h:(\mathrm{s}_3)_{\mathsf{d}}}.\ \uparrow_{\mathsf{d}} h$ sandy fido? As a consequence of the previous line of reasoning, this is the same as asking whether P_{a} holds of $\lambda_{h:(\mathrm{s}_3)_{\mathsf{d}}}.\lambda_{u:\mathrm{s}}.u \cdot (\uparrow_{\mathsf{d}} h\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) \cdot \text{sandy} \cdot \text{fido}$.
By examining the definition of P more precisely, we can see that we are asking about the truth of the following:

$$\vdash \exists_{v:\mathrm{s}}.\ \lambda_{h:(\mathrm{s}_3)_{\mathsf{d}}}.\lambda_{u:\mathrm{s}}.u \cdot (\uparrow_{\mathsf{d}} h\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) \cdot \text{sandy} \cdot \text{fido} = \mathsf{a}\ v.$$

Since a is defined as $\lambda_{v:\mathrm{s}}.\lambda_{p:(\mathrm{s}_3)_{\mathsf{d}}}.\lambda_{s:\mathrm{s}}.s \cdot (\uparrow_{\mathsf{d}} p\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) \cdot v$, this amounts to asking whether

$$\vdash \exists_{v:\mathrm{s}}.\ \lambda_{h:(\mathrm{s}_3)_{\mathsf{d}}}.\lambda_{u:\mathrm{s}}.u \cdot (\uparrow_{\mathsf{d}} h\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) \cdot \text{sandy} \cdot \text{fido} = \lambda_{p:(\mathrm{s}_3)_{\mathsf{d}}}.\lambda_{s:\mathrm{s}}.s \cdot (\uparrow_{\mathsf{d}} p\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) \cdot v.$$

After α-conversion, the answer is yes, with $v = \text{sandy} \cdot \text{fido}$, so

$$\vdash P_{\mathsf{a}}\ (\lambda_{h:(\mathrm{s}_3)_{\mathsf{d}}}.\ \uparrow_{\mathsf{d}} h \text{ sandy fido}) \quad \square$$

and the subtyping is justified.

$$\text{L4} \quad \vdash P_{\mathsf{a}}\ (\lambda_{h:(\mathrm{s}_3)_{\mathsf{d}}}.\ \uparrow_{\mathsf{d}} h \text{ leslie rover})$$

As in L3, mutatis mutandis.

6.2.3 Subtyping of *gave*

Finally, we derive the subtyping of the ditransitive verb *gave* as being under image of the d phenominator:

$$\text{L5}\quad \frac{\vdash \mathsf{d}\ \text{gave} : \mathrm{s}_3; \mathrm{DV} \qquad \vdash P_{\mathsf{d}}\ (\mathsf{d}\ \text{gave})}{\vdash \downarrow_{\mathsf{d}}\ (\mathsf{d}\ \text{gave}) : (\mathrm{s}_3)_{\mathsf{d}}; \mathrm{DV}}\downarrow$$

As for the justification of the second premiss, it follows immediately from the definition of P that $\vdash P_{\mathsf{d}}\ (\mathsf{d}\ \text{gave})$.

6.2.4 Instantiation of the *and* lexical entry

Owing to the complexity of the structure of the conjunct clusters, the instantiation of our coordination lexical entry is similarly complex, with $B = \mathrm{DV} \multimap \mathrm{VP}$, $A = (\mathrm{s}_3)_{\mathsf{d}} \to \mathrm{s}_1$, and $\phi = \mathsf{a}$:

$$\begin{array}{ll} \vdash & \lambda_{pq:((\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1)_{\mathsf{a}}}.\downarrow_{\mathsf{a}}\ (\mathsf{a}\ (\mathsf{say}_{(\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1}\ p)\cdot \mathrm{and} \cdot (\mathsf{say}_{(\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1}\ q)) \\ : & ((\mathrm{s}_3)_{\mathsf{d}} \to \mathrm{s}_1)_{\mathsf{a}} \to ((\mathrm{s}_3)_{\mathsf{d}} \to \mathrm{s}_1)_{\mathsf{a}} \to ((\mathrm{s}_3)_{\mathsf{d}} \to \mathrm{s}_1)_{\mathsf{a}} \\ ; & (\mathrm{DV} \multimap \mathrm{VP}) \multimap (\mathrm{DV} \multimap \mathrm{VP}) \multimap (\mathrm{DV} \multimap \mathrm{VP}) \end{array}$$

It is useful to see exactly how the recursive definition for say works out for the type of the conjuncts in question. Perhaps the most central insight to be gleaned from this comes from asking the question of how exactly one "says" a vacuous verb. The answer is that while you can't "say" the verb itself, but you could "say" whatever arguments it has in the order that the verb specifies. This is accomplished by taking the phenominator associated with the verb in question (here, d), and applying it to the empty string, in order to create a function that places the verbal string arguments in the correct relative location.

$$\begin{array}{ll} & \mathsf{say}_{((\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1)_{\mathsf{a}}} \\ = & \lambda_{p:((\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1)_{\mathsf{a}}}.\mathsf{say}_{(\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1}(\uparrow_{\mathsf{a}} p) \\ = & \lambda_{p:((\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1)_{\mathsf{a}}}.\mathsf{say}_{\mathrm{s}_1}(\uparrow_{\mathsf{a}} p\ \mathsf{vac}_{(\mathrm{s}_3)_{\mathsf{d}}}) \\ = & \lambda_{p:((\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1)_{\mathsf{a}}}.\mathsf{say}_{\mathrm{s}_1}(\uparrow_{\mathsf{a}} p\ (\downarrow_{\mathsf{d}}\ (\mathsf{d}\ \mathsf{e}))) \\ = & \lambda_{p:((\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1)_{\mathsf{a}}}.\mathsf{say}_{\mathrm{s}_1}(\uparrow_{\mathsf{a}} p\ (\downarrow_{\mathsf{d}}\ (\lambda_{vstu}.u\cdot v\cdot s\cdot t\ \mathsf{e}))) \\ = & \lambda_{p:((\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1)_{\mathsf{a}}}.\mathsf{say}_{\mathrm{s}_1}(\uparrow_{\mathsf{a}} p\ (\downarrow_{\mathsf{d}}\ \lambda_{stu}.u\cdot \mathsf{e}\cdot s\cdot t)) \\ = & \lambda_{p:((\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1)_{\mathsf{a}}}.\mathsf{say}_{\mathrm{s}_1}(\uparrow_{\mathsf{a}} p\ (\downarrow_{\mathsf{d}}\ \lambda_{stu}.t\cdot s\cdot u)) \\ = & \lambda_{p:((\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1)_{\mathsf{a}}}.\mathsf{say}_{\mathrm{s}}(\uparrow_{\mathsf{a}} p\ (\downarrow_{\mathsf{d}}\ \lambda_{stu}.u\cdot s\cdot t)\ \mathsf{vac}_{\mathrm{s}}) \\ = & \lambda_{p:((\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1)_{\mathsf{a}}}.\mathsf{say}_{\mathrm{s}}(\uparrow_{\mathsf{a}} p\ (\downarrow_{\mathsf{d}}\ \lambda_{stu}.u\cdot s\cdot t)\ \mathsf{e}) \\ = & \lambda_{p:((\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1)_{\mathsf{a}}}.\lambda_{s:\mathrm{s}}.s\ (\uparrow_{\mathsf{a}} p\ (\downarrow_{\mathsf{d}}\ \lambda_{stu}.u\cdot s\cdot t)\ \mathsf{e}) \\ = & \lambda_{p:((\mathrm{s}_3)_{\mathsf{d}}\to\mathrm{s}_1)_{\mathsf{a}}}.(\uparrow_{\mathsf{a}} p\ (\downarrow_{\mathsf{d}}\ \lambda_{stu}.u\cdot s\cdot t)\ \mathsf{e}) \end{array}$$

It is easy to see, then, that the lexical entry for ACC *and* boils down to the following:

$$\begin{array}{lll} \text{L6} & \vdash & \lambda_{pq}.\downarrow_{\mathsf{a}}\ (\mathsf{a}\ ((\uparrow_{\mathsf{a}} p\ (\downarrow_{\mathsf{d}}\ \lambda_{stu}.u\cdot s\cdot t)\ \mathsf{e})\cdot \mathrm{and} \cdot (\uparrow_{\mathsf{a}} q\ (\downarrow_{\mathsf{d}}\ \lambda_{stu}.u\cdot s\cdot t)\ \mathsf{e}))) \\ & : & ((\mathrm{s}_3)_{\mathsf{d}} \to \mathrm{s}_1)_{\mathsf{a}} \to ((\mathrm{s}_3)_{\mathsf{d}} \to \mathrm{s}_1)_{\mathsf{a}} \to ((\mathrm{s}_3)_{\mathsf{d}} \to \mathrm{s}_1)_{\mathsf{a}} \\ & ; & (\mathrm{DV} \multimap \mathrm{VP}) \multimap (\mathrm{DV} \multimap \mathrm{VP}) \multimap (\mathrm{DV} \multimap \mathrm{VP}) \end{array}$$

6.3 The completed ACC derivation

For the sake of brevity, we stipulate that the conjunct variable q has type $((\mathrm{s}_3)_\mathsf{d} \to \mathrm{s}_1)_\mathsf{a}$, and we will omit the type annotation from the derivation below. We will sometimes add line breaks to the signs below to improve legibility, and we suggest that the reader read : as "has phenotype" and ; as "has tectotype". The reader should note the inference following the inclusion of L2, wherein the phenominator a appears in its explicit form, applied to the completed coordinate structure, rather than in its previous abbreviated form.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\begin{array}{l}\vdash \lambda_q.\downarrow_\mathsf{a} (\mathsf{a}\ (\text{sandy} \cdot \text{fido} \cdot \text{and} \cdot (\uparrow_\mathsf{a} q\ (\downarrow_\mathsf{d} \lambda_{stu}.u \cdot s \cdot t)\ \mathsf{e})))\\ : ((\mathrm{s}_3)_\mathsf{d} \to \mathrm{s}_1)_\mathsf{a} \to ((\mathrm{s}_3)_\mathsf{d} \to \mathrm{s}_1)_\mathsf{a}\\ ; (\mathrm{DV} \multimap \mathrm{VP}) \multimap (\mathrm{DV} \multimap \mathrm{VP})\end{array}}\ \text{L6 L1}}{\begin{array}{l}\vdash \downarrow_\mathsf{a} (\lambda_{p:(\mathrm{s}_3)_\mathsf{d}}.\lambda_s.s \cdot (\uparrow_\mathsf{d} p\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) \cdot \text{sandy} \cdot \text{fido} \cdot \text{and} \cdot \text{leslie} \cdot \text{rover})\\ : ((\mathrm{s}_3)_\mathsf{d} \to \mathrm{s}_1)_\mathsf{a}; \mathrm{DV} \multimap \mathrm{VP}\end{array}\ \text{L2}}{\begin{array}{l}\vdash \lambda_{p:(\mathrm{s}_3)_\mathsf{d}}.\lambda_s.s \cdot (\uparrow_\mathsf{d} p\ \mathsf{e}\ \mathsf{e}\ \mathsf{e}) \cdot \text{sandy} \cdot \text{fido} \cdot \text{and} \cdot \text{leslie} \cdot \text{rover}\\ : (\mathrm{s}_3)_\mathsf{d} \to \mathrm{s}_1; \mathrm{DV} \multimap \mathrm{VP}\end{array}\ \uparrow}{\vdash \lambda_s.s \cdot \text{gave} \cdot \text{sandy} \cdot \text{fido} \cdot \text{and} \cdot \text{leslie} \cdot \text{rover} : \mathrm{s}_1; \mathrm{VP}}\ \text{L5} \qquad \vdash \text{kim} : \mathrm{s}; \mathrm{NP}}{\vdash \text{kim} \cdot \text{gave} \cdot \text{sandy} \cdot \text{fido} \cdot \text{and} \cdot \text{leslie} \cdot \text{rover} : \mathrm{s}; \mathrm{S}}$$

In summary: we first apply the sign for *and* to each conjunct. This results in an expression "missing" both its ditransitive verb (a ternary function over strings which is under the image of the d phenominator) and its NP subject (a string). This entire expression is under the a phenominator, meaning that to make use of it, we pass it through the $\uparrow$ rule, embedding it within its supertype. Now we can apply it first to the ditransitive verb *gave* and the NP *Kim*, in order to obtain the complete expression *Kim gave Sandy Fido and Leslie Rover*, which is asserted to be a phenogrammatical string, and a sentence in the tectogrammar, as was desired.

7 Comparison with Hybrid Type Logical Grammar

As mentioned above, Kubota and Levine's HTLG framework is an extension of Lambek grammar that resembles the curryesque frameworks in making use of pheno terms. Here we compare HTLG's analysis of coordination with our own. To facilitate the comparison, we reformulate HTLG from Kubota and Levine's Prawitz-style natural deduction presentation into a sequent-style natural deduction presentation.

The HTLG tectotypes are recursively, but not freely, generated over a set of basic tectotypes as follows:

a. Every basic tectotype is a Lambek tectotype.

b. If A and B are Lambek tectotypes, so are $A\backslash B$ and B/A.

c. Lambek tectotypes are HTLG tectotypes.

d. If A and B are HTLG tectotypes, so is $A \multimap B$.

Assuming (as we will) that the two frameworks use the same basic tectotypes, the LCG tectotypes are a proper subset of the HTLG tectotypes, namely the ones that make no use of the directional implications / and \ in their construction; we call these the **linear** tectotypes. Conversely, the Lambek tectotypes are the ones that make no use of the linear

implication $\multimap$ in their construction. From these considerations it follows that the basic tectotypes are both linear tectotypes and Lambek tectotypes.

Significantly, in an HTLG sign, there can never be an occurrence of $\multimap$ under a directional implication. For example, $\bar{\mathrm{Q}}/(\mathrm{NP} \multimap \mathrm{S})$ is not an HTLG tectotype, even though to the uninitiated it might appear to be the natural tectotype assignment for interrogative *who*. Relatedly, in HTLG, a sign with a Lambek tectotype always has a string pheno component, while a sign with a nonbasic linear tectotype always has a functional pheno component.

The rules of HTLG are just the rules of LCG plus the following elimination and introduction rules for the directional implications:

$$\frac{\Gamma \vdash a; A \qquad \Delta \vdash b; A\backslash B}{\Gamma, \Delta \vdash a \cdot b; B}\ \backslash\mathrm{E}$$

$$\frac{\Gamma, s; A \vdash s \cdot a; B}{\Gamma \vdash a; A\backslash B}\ \backslash\mathrm{I}$$

$$\frac{\Gamma \vdash b; B/A \qquad \Delta \vdash a; A}{\Gamma, \Delta \vdash b \cdot a; B}\ /\mathrm{E}$$

$$\frac{\Gamma, s; A \vdash a \cdot s; B}{\Gamma \vdash a; B/A}\ /\mathrm{I}$$

One complication that arises in trying to compare HTLG with LCG_ϕ is that, given an HTLG sign with a nonbasic Lambek tectotype (and therefore a string pheno), there is a canonical way to use the grammar rules to derive a sign with a nonbasic linear tectotype (and therefore a functional pheno). HTLG derivations of this kind are collectively referred to as *unslanting*. Such expressions are obviously easier to compare with their LCG_ϕ counterparts in unslanted form. But they can only be coordinated in *slanted* form. This is because the lexical schema for *and* is

$$\vdash \mathrm{and}; (A\backslash A)/A$$

where, crucially, the tectotype metavariable A ranges over *Lambek* types. From this it follows that, in order to coordinate signs whose tectotypes are not Lambek types, first the grammar rules have to be used to *slant* the signs into ones with Lambek tectotypes. This is not always possible, and when it *is* possible, typically there are multiple, nonequivalent ways to do it; that is, it is possible to produce multiple signs with distinct Lambek tectotypes from a single sign with a non-Lambek tectotype.

An expository complication arises in connection with the tectotype abbreviations VP, TV, and DV, since we have used two different sets of conventions for interpreting them (one for LCG/LCG_ϕ and one for Lambek categorial grammar). To avoid ambiguity, in this section we use these abbreviations only with their Lambek interpretations.

In the simplest case, namely non-functors (e.g. NPs), coordination in HTLG and LCG_ϕ differ only in the respect that the HTLG *and* is phenogrammatically a string that is combined with its conjunct tecto-arguments by concatenation, while the LCG_ϕ *and* is phenogrammatically a function that linearizes the conjunct phenos by taking them as pheno-arguments:

HTLG: $\vdash \mathrm{and}; (\mathrm{NP}\backslash\mathrm{NP})/\mathrm{NP}$

LCG_ϕ: $\vdash \lambda_{st}.s \cdot \mathrm{and} \cdot t; \mathrm{NP} \multimap \mathrm{NP} \multimap \mathrm{NP}$

This theory-internal technical difference has no empirical consequences.[12]

Next we consider simple cases of functor coordination where the conjuncts themselves select arguments with basic tectotypes, such as VP, S/NP (right node raising), TV, and DV:

$\vdash$ brayed; VP

$\vdash$ pedro $\cdot$ beat; S/NP

$\vdash$ beat; TV

$\vdash$ gave; DV

In such cases, the LCG_ϕ counterparts of the HTLG conjuncts are the same as their HTLG canonical unslanted versions, and they have to have $\downarrow_\phi$ (where ϕ is the appropriate phenominator, in the present cases i, r, t, or d respectively) applied to them before being taken as arguments by the appropriate instance of the LCG_ϕ *and*-schema. Such cases appear straightforward because it is easy to see by inspection what the appropriate phenominator must be (just replace the string support, which is the maximal subterm consisting of a concatenated sequence of string constants, by a string variable and then abstract on that variable).

$$\dfrac{\dfrac{\dfrac{\vdash \lambda_{st:\mathrm{s}}.t \cdot \text{beat} \cdot s : \mathrm{s}_2; \text{NP} \multimap \text{NP} \multimap \text{S} \qquad v : \mathrm{s}; \text{NP} \vdash v : \mathrm{s}; \text{NP}}{v : \mathrm{s}; \text{NP} \vdash \lambda_{t:\mathrm{s}}.t \cdot \text{beat} \cdot v : \mathrm{s}_1; \text{NP} \multimap \text{S}} \qquad \vdash \text{pedro} : \mathrm{s}; \text{NP}}{v : \mathrm{s}; \text{NP} \vdash \text{pedro} \cdot \text{beat} \cdot v : \mathrm{s}_1; \text{S}}}{\vdash \lambda_{v:\mathrm{s}}.\text{pedro} \cdot \text{beat} \cdot v : \mathrm{s}_1; \text{NP} \multimap \text{S}} \qquad \vdash P_{\mathsf{r}}\ (\lambda_{v:\mathrm{s}}.\text{pedro} \cdot \text{beat} \cdot v)}{\vdash \downarrow_{\mathsf{r}}\ (\lambda_{v:\mathrm{s}}.\text{pedro} \cdot \text{beat} \cdot v) : (\mathrm{s}_1)_{\mathsf{r}}; \text{NP} \multimap \text{S}}\ \downarrow$$

A more complicated situation is one where the conjuncts to be unslanted themselves take a functor argument, such as ACC. Overall, the HTLG and LCG_ϕ analyses are still relatable in essentially the same way as the cases discussed immediately above, but the connection is harder to see at a glance because it is harder to grasp by visual inspection that the appropriate phenominator for an LCG_ϕ ACC conjunct like

$$\lambda_{h:(\mathrm{s}_3)_{\mathsf{d}}}.\uparrow_{\mathsf{d}} h\ \text{sandy fido} : (\mathrm{s}_3)_{\mathsf{d}} \to \mathrm{s}_1; \text{DV} \multimap \text{VP}$$

is the a phenominator. In fact this points to a weakness in the LCG_ϕ coordination analysis as it stands. What is at issue here is whether there is an algorithm that inputs an LCG_ϕ pheno term and decides (a) whether it is under the image of a phenominator, and (b) if so, which one. In the present case, we found the right phenominator for ACC conjuncts with the help of an experimental algorithm that generates a phenominator from a Lambek type, but what we need is one that works for terms of an arbitrary LCG_ϕ phenotype.

Other issues are raised in the case of signs with nonbasic linear types that can be slanted but not in a unique way. For example, in LCG_ϕ, any two QPs such as

$\vdash \mathsf{q}$ someone : s_{-2}; QP

and

$\vdash \mathsf{q}$ something : s_{-2}; QP

[12]Here the LCG_ϕ lexical entry for *and* has been written to combine first with the left conjunct, but it could just as well have been done the other way; in fact the right-conjunct-first version is derivable.

can be conjoined 'right off the shelf' once $\downarrow_{\mathsf{q}}$ has been applied to them. In HTLG though, they cannot be conjoined until they have been slanted (and slanted the same way); but there are many ways in HTLG of slanting a QP, depending on where it occurs and what it scopes over. For example, a subject QP that scopes over the local S should slant to S/(NP\S), and a QP object of transitive which scopes over the local S should slant to (S/NP)\S; but it is unclear what to do if the QP in question is nonperipheral, or scopes to a superordinate S, or both. We have not as yet seen an example for which we are unable to find an appropriate slanting, but we do not know whether there is a fully general algorithm for slanting QPs to be conjoined.

Finally, we briefly consider coordination of expressions for which, seemingly, no appropriate HTLG slanting exists. The simplest example of this we know of is *wh*-expressions 'displaced' to the left periphery of a finite clause, e.g.

$\vdash \mathsf{w}\ \text{who} : \mathrm{s}_{-2}; \mathrm{WP}$

$\vdash \mathsf{w}\ \text{what} : \mathrm{s}_{-2}; \mathrm{WP}$

In LCG_{ϕ}, sentences like

(3) I wonder [who and what] Kim showed to Fido.

are analyzed unproblematically, because the *wh*-conjuncts are simply conjoined in the usual way courtesy of the w phenominator, without any consideration of where the 'displaced' coordinate structure occurs, or where the gap that it is related to occurs. In HTLG by contrast, it is not currently clear how to analyze such examples.

8 Residual Issues

We have sketched the outlines of LCG_{ϕ}, a kind of categorial grammar that maintains Curry's sharp distinction between tectogrammar and phenogrammar, while elaborating the latter with fine-grained phenotypes that allow for the encoding of functor directionality. Thus it improves on previous 'curryesque' frameworks (such as Oehrle 1994, ACG, lambda grammar, and LCG simpliciter) in providing a straightforward theory of functor coordination: functors can be coordinated only if they are in the image of the same phenominator.

Much remains to be done. On the empirical side, there are constructions yet to be analyzed that are generally considered under the rubric of coordination, but where the conjuncts are not in the image of a phenominator, such as gapping (*Kim saw Fido, and Sandy Rover*) and conjoined scopes of '*wh*-movement' (*This is the farmer who Maria loves but Chiquita thinks is sadistic*). This last class of examples is problematic not simply because the conjuncts can fail to be under a phenominator, but also because the presence of multiple gaps related to a single 'displaced' expression (so-called ATB extraction) poses a challenge to the underlyingly linear character of the grammar. This issue arises too in the case of 'parasitic gap' constructions (*reports that Sandy filed without reading*, *which rebel leader did former supporters of assassinate?*).

Moreover, there are constructions other than coordination whose analysis seems to require reference to directionality. For example, in English, *that*-less subject relatives are impossible if the relatived position is the subject of the relative clause, but fine if it is the subject of an embedded clause ($*$ *This is the donkey kicked Pedro* vs. *This is the donkey Maria thinks kicked Pedro*). Evidently, an NP $\multimap$ S can be a relative clause, but not if it is, phenogrammatically, an $(s_1)_{\mathrm{i}}$; that is, the gap cannot be on the left periphery.

Also to be considered are applications of fine-grained phenotyping to empirical domains other than directionality. One of these is the analysis of tune. For example, we could posit a theory of focus that involves in the phenogrammar the alignment a pitch accent with a particular expression. If the expression in question is a string, then one can employ a representation that encodes the fact that the string is accent-bearing. Analyzing constructions within which the focused element is an expression we would represent as a functor is more difficult. Since the focus target is no longer a string, but a (potentially higher order) function over strings, our task in the phenogrammar is to ensure that the requisite **sub**strings of the functor bear an accent, while maintaining the overall linearization structure it expresses. The function say seems well-suited to the former task, and phenominators to the latter.

In this paper, we ignored the tectogrammatics of coordination, considering only binary coordination of conjuncts with like tectotypes. As Worth (in progress) argues, the extension of the analysis proposed here beyond these simple cases requires the introduction of disjunction and strong monad (or equivalent) types in the tectogrammar. In analyzing constructions like *evil and a tyrant*, disjunction types enable us to assign the same tectotype to the conjuncts, and monad types ensure that the resulting construction is itself sensibly typed. Relatedly, there is the issue of *scope* of coordination exemplified by amiguities such as *Bob thinks that Julie or Julia submitted the grade change forms, but he doesn't know which/but I don't remember which.* Could it be that the coordination monad is, or is closely related to, the continuation monad?

More work is needed on the formal underpinnings of LCG_ϕ. For example, we suspect that the machinery of say and vac functions can be streamlined by taking better advantage of what is known (but alas, not by us) about linear lambda terms. Of particular urgency is the decidability issue that arises from the need to ascertain, for a given sign, whether its pheno is in the image of a phenominator, and if so, which one. Development of an algorithm to calculate the phenominator for a given Lambek tectotype is under way, and while we have had some preliminary success, the correctness of the algorithm is an open question.

We would like to have a clearer understanding of the relationship between LCG_ϕ and HTLG. What would it mean to 'embed' the former into the latter or the latter into the former? For example, intuitively one senses that the two frameworks analyze ACC 'the same' way but analyze conjoined QPs 'different ways', but can we make those intuitions precise?

Finally, there is the question of whether the 'curryesque' program (of sharply distinguishing phenogrammar from tectogrammar) is on the right track. Mainstream categorial grammar gets by without this distinction. HTLG *makes* the distinction in terms of formal architecture, but the presence of the directional implications with their 'admixture of phenogrammatics' seems to muddle it. If one is going to employ the directional implications anyway, then why not just dispense with a distinct phenogrammatical component and work in one of the extended Lambek formalisms instead? LCG_ϕ maintains the sharp pheno/tecto distinction, but theoretical purity is not worth much if it is too hard to extend the empirical coverage or if the decidability issue cannot be resolved favorably. Clearly, there is plenty left to do.

References

Barker, Chris and Chung-chieh Shan. 2014. *Continuations and natural language*, vol. 53 of *Oxford Studies in Theoretical Linguistics*. Oxford University Press.

Carpenter, Bob. 1997. *Type-Logical Semantics*. Bradford books. MIT Press.

Curry, Haskell B. 1961. Some Logical Aspects of Grammatical Structure. In R. O. Jakobson, ed., *Structure of Language and its Mathematical Aspects*, 56–68. American Mathematical Society.

de Groote, Philippe. 2001. Towards Abstract Categorial Grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, 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*, vol. 32 of *Studies in Linguistics and Philosophy*, 153–197. Springer Netherlands.

Duan, Manjuan. 2015. A categorial analysis of Chinese alternative questions. This volume.

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

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. 2013. Coordination in hybrid type-logical categorial grammar. *Ohio State University Working Papers in Lingustics* 60:21–50.

Kubota, Yusuke and Robert Levine. 2015. 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.

Lambek, Joachim. 1958. The mathematics of sentence structure. *American mathematical monthly* 154–170.

Lambek, Joachim and Philip J. Scott. 1986. *Introduction to Higher Order Categorical Logic*. Cambridge University Press, Cambridge.

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

Martin, Scott. 2015. A unidimensional syntax-semantics interface for supplements. This volume.

Martin, Scott and Carl Pollard. 2014. A dynamic categorial grammar. In G. Morrill, R. Muskens, R. Osswald, and F. Richter, eds., *Formal Grammar*, vol. 8612 of *Lecture Notes in Computer Science*, 138–154. Springer Berlin Heidelberg.

Mihaliček, Vedrana. 2012. *Serbo-Croatian Word Order: A Logical Approach*. Ph.D. thesis, The Ohio State University.

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.

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

Moot, Richard. 2014. Hybrid type-logical grammars, first-order linear logic, and the descriptive inadequacy of lambda grammars. Unpublished manuscript.

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

Muskens, Reinhard. 2001. Lambda grammars and the syntax-semantics interface. In *Proceedings of the Thirteenth Amsterdam Colloquium*, 150–155.

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. 1994. Term-Labeled Categorial Type Systems. *Linguistics and Philosophy* 17:633–678.

Pollard, Carl. 2015. Agnostic hyperintensional semantics. *Synthese* 192:535–562.

Pollard, Carl and E. Allyn Smith. 2012. A unified analysis of *the same*, phrasal comparatives, and superlatives. In A. Chereches, ed., *Proceedings of SALT 22*, 307–325. eLanguage.

Smith, E. Allyn. 2010. *Correlational Comparison in English*. Ph.D. thesis, The Ohio State University.

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

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.

Worth, Chris. In progress. *English Coordination in Linear Categorial Grammar*. Ph.D. thesis, The Ohio State University.

Right-Node Wrapping: A Combinatory Account*

Alex Warstadt
Brown University

Abstract

This paper shows that a new pattern of non-canonical coordination (NCC) going under the rubric Right-Node Wrapping (RNW) follows as a natural consequence of adequate accounts of coordination and discontinuous constituency. It formalizes a combinatory variant of Categorial Grammar (CG) which permits a highly limited notion of discontinuous constituency that nonetheless leads to empirical successes. It also demonstrates that the coordination data captured by Dowty's (1988) continuous combinatory CG can be accounted for straightforwardly in the present system with minor extensions. Crucially, I show that RNW data can be largely accounted for by just these independently motivated analyses. In RNW coordinations, the pivot expression shared by the conjuncts is followed by some additional expression interpreted as part of the rightmost conjunct alone (Wilder, 1999; Whitman, 2009). Due to empirical similarities between RNW and NCC, any adequate account of NCC should naturally predict RNW. However, previous like-category coordination analyses of NCC in CG can only generate peripheral pivots, thereby failing on RNW, while the present account succeeds by assigning the right conjunct discontinuous constituency. Previously unknown patterns of RNW are also considered and shown to follow from the analysis. The present formulation of discontinuity is compared favorably to existing discontinuity-based analyses of RNW in multimodal Type Logical Grammar (TLG) (Whitman, 2009; Kubota, 2014, Ms.). Finally, it is argued that the variety of discontinuity found in RNW cannot be subsumed under more general notions of discontinuity in TLG (e.g. Kubota and Levine, forthcoming; Morrill and Valentín, 2012).

Keywords: Right-Node Wrapping, discontinuity, Discontinuous Combinatory Categorial Grammar, coordination, wrap, Type-Logical Grammar

1 Introduction

Categorial Grammar[1] (CG) is celebrated for its elegant accounts of non-canonical constituent coordination (NCC) (e.g. Steedman, 1985; Dowty, 1988) and discontinuous constituency by *wrapping* (e.g. Bach, 1979; Moortgat, 1988). Only recently, however, have novel empirical discoveries—namely the Right-Node Wrapping (RNW) phenomenon (Wilder, 1999; Whitman, 2009)—focused research on the intersection of these domains.[2]

For the present discussion, RNW is defined as any coordination (non-canonical or otherwise) in which the rightmost conjunct is a discontinuous constituent surrounding

*This paper owes much to the comments of three anonymous reviewers for CG2015, to the suggestions of the faculty and students in the CLPS department at Brown University, and most of all to the invaluable guidance of Polly Jacobson.

[1]In this paper 'CG' refers to both the combinatory and type-logical approaches, for which I use 'CCG' and 'TLG' when relevant. 'CCG' is not restricted to Steedman's (e.g. 2000) theory of the same name.

[2]Of course Gapping represents a long-known type of discontinuous coordination, but as I argue in §4.5 this is not properly a *wrap* phenomenon.

some expression shared by all the conjuncts (henceforth the 'pivot'). Consider (1): the left conjunct (A) and the right conjunct (B, B′) both take the pivot (C) as argument. Note that an alternative analysis in which the conjuncts are A and B and the pivot C-B′ is not possible, as B′ is not a semantic argument of A. (In this and in other examples, conjuncts are shown in square brackets, discontinuous constituents underlined, and pivots in boldface.)

(1) John should [fetch] and [give **the book** to Mary]. (Wilder, 1999)
A B C B′

As I explore in §4, RNW is far less restricted than this single example suggests, and even richer than previously acknowledged in the literature. This paper defends the position that (nearly) all these new data can be gotten for free from adequate, independently motivated analyses of NCC and discontinuity.

2 Discontinuous CCG

Combinatory Categorial Grammars (CCGs) are extensions of the categorial systems of Ajdukiewicz (1935) and Bar-Hillel (1953) which postulate a number of primitive combinatory rules that operate on the phonology, syntax, and semantics of expressions. Following Jacobson (2014), all expressions, including lexical entries, take the form of triples of phonology, syntactic category, and semantics. In the following I notate an expression α with phonology $[\alpha]$, category A, and semantics α' as follows: $\alpha : \langle [\alpha] ; A ; \alpha' \rangle$. For expository purposes, I will refer to phonological strings with English orthography and omit angle brackets and the semantic dimension whenever possible.

The discontinuous CCG (DCCG) adopted in this paper pairs a new prosodic calculus with the categorial syntax using a set of combinatory rules. In the prosodic calculus there are four operations on strings O_i which build larger strings from their constituent parts. Each operation on strings is paired with a category constructor $/_i$ in the syntax that shares its subscript. As in many other CG approaches, the semantics is directly compositional, built up in parallel with the syntax.

The most basic of combinatory rules is 'function application' (**FA**), which corresponds to the slash elimination rules in TLG. Like the other combinatory rules, **FA** has effects in each dimension: in the syntactic dimension the main slash of the functor is eliminated, in the semantic dimension the functor is applied to the argument, and in the phonological dimension the functor and argument expressions are taken as first- and second-argument respectively to the operation on strings corresponding to the functor's main slash. The stipulation that the functor is always the *first* argument into the operation on strings becomes a central point of analysis to follow.

(2) **FA**
Given expressions $\alpha : \langle [\alpha] ; A/_i B ; \alpha' \rangle$ and $\beta : \langle [\beta] ; B ; \beta' \rangle$, infer an expression $\gamma : \langle [O_i(\alpha, \beta)] ; A ; \alpha'(\beta') \rangle$. (Jacobson, 2014a)

The prosodic calculus operates on strings of two sorts. The continuous sort L^0 includes strings with no infixation point (i.e. discontinuity), while the discontinuous sort L^1 includes strings with exactly one infixation point (denoted as in Jacobson (2014a) with the vertical bar '$|$'). It contains two modes of operations on strings: a continuous and a discontinuous mode, each of which includes two operations defined polymorphically over input strings of different sorts. This way, the operations of the continuous mode are still defined over

discontinuous arguments.

The operations in the continuous mode, *left-* and *right-concatenation*—O_L and O_R—are identical (but with their arguments reversed), except in the $L^1 \times L^1$ condition. In this case the convention is that only the discontinuity in the first argument is preserved in the resulting string.

(3) For strings $w,\ x,\ y,\ z\ \in\ L^0$:

$$O_R : \begin{cases} O_R(x,\ y) = xy \\ O_R(x|y,\ z) = x|yz \\ O_R(x,\ y|z) = xy|z \\ O_R(w|x,\ y|z) = w|xyz \end{cases} \qquad O_L : \begin{cases} O_L(x,\ y) = yx & L^0 \times L^0 \to L^0 \\ O_L(x|y,\ z) = zx|y & L^1 \times L^0 \to L^1 \\ O_L(x,\ y|z) = y|zx & L^0 \times L^1 \to L^1 \\ O_L(w|x,\ y|z) = yzw|x & L^1 \times L^1 \to L^1 \end{cases}$$

The convention for 'passing up' infixation points is simple: all infixation points in the arguments are preserved in the output unless both arguments have an infixation point, in which case only the functor's remains.

The pair of operations in the discontinuous mode is *wrap*—O_W—and *infixation*—O_I. As with concatenation, they are defined polymorphically, though this time for all strings $x, /y$, it holds that $O_I(x,\ y)\ =\ O_W(y,\ x)$; for this reason I omit the O_I definitions. Put generally, the argument which ends up discontinuous in the resulting string must have an infixation point. In the resulting expression, the infix string occupies the place of the infixation point, so the number of infixation points in the output is always one less than that of the arguments together.

(4) For strings $w,\ x,\ y,\ z\ \in\ L^0$:

$$O_W : \begin{cases} O_W(x,\ y) = N/A & L^0 \times L^0 \to N/A \\ O_W(x|y,\ z) = xzy & L^1 \times L^0 \to L^0 \\ O_W(x,\ y|z) = N/A & L^0 \times L^1 \to N/A \\ O_W(w|x,\ y|z) = wy|zx & L^1 \times L^1 \to L^1 \end{cases}$$

2.1 introducing discontinuity

The operations on strings introduced so far only preserve discontinuities or eliminate them. For English, then, I propose an additional unary operation on strings O_{IR} which equips a continuous string with an infixation point on its right edge.

(5) For strings $x,\ y\ \in\ L^0$:

$$O_{IR} : \begin{cases} O_{IR}(x) = x| & L^0 \to L^1 \\ O_{IR}(x|y) = N/A & L^1 \to N/A \end{cases}$$

Accordingly there is a unary combinatory rule **I** (mnemonic for 'infixation' and 'identity') which is paired with O_{IR}. **I** applies O_{IR} to the phonology of the input, shifts the innermost $/_R$ of the input's category to a $/_W$, and applies the combinator 'I' (the identity function) to the semantics. Significantly, **I** is defined so that the introduction of infixation points is restricted to 'syntactic words' of certain categories.

Such a claim, of course, requires a well-defined notion of word. For the time being, I follow Hoeksema (1984), Zwicky (1992), and Moortgat and Oehrle (1994) in supposing that the prosodic sort of an expression should be marked in the expression's syntactic category. In the notation of Hoeksema, X^0 contains all expressions in X of the 'word' sort, and X^1 those expressions of the 'phrase' sort. A property of the Hoeksema and the Moortgat and

Oehrle accounts, and one which I adopt, is that any part of a complex category may be marked with its sort. For example, in a morphologically complex N^0 'electric organ player', the subexpression 'electric organ' can have the category $(N^0/_R N^0)^1$—i.e. it is a member of the phrase sort that is a function from a word to another word. To eliminate notational clutter, these details will be omitted when irrelevant.

This restriction is accomplished by defining **I** over members of the word sort alone. **I** is further restricted to functions which after taking some non-zero number of arguments return either an $S/_L NP$ or an N. Such categories include transitive verbs, ditransitive verbs, attributive adjectives, and predicative adjectives. (This is the meaning of the $X\$$, which is recursively defined as the set of expressions α such that $\alpha \in X$ or $\alpha \in B/_i C$ for all B and for all $C \in X\$$. $X\$_i$ is some particular member of $X\$$.)

(6) **I**
Given an expression $\alpha : \langle [\alpha] ; (X/_R Y)\$_i^0 ; \mathrm{a} \rangle$, infer an expression $\beta : \langle [O_{IR}(\alpha)] ; (X/_W Y)\$_i^0 ; a \rangle$ where $X \in \{S/_L NP, N\}$.

2.2 some empirical motivation

So far, DCCG can already account for considerable discontinuity data. Ditransitives like *give* are stored in the lexicon with the category $(VP/_R NP)/_R PP$. By stipulating only the continuous category in the lexicon, we obtain the two word orders associated with the ditransitive by the optional application of **I**: the heavy-NP shift *V-PP-NP* order (7a) is gotten from the lexical category of *give*, and the canonical *V-NP-PP* order (7b) is gotten from applying **I** to *give*.

(7) a.

$$\dfrac{\dfrac{\overset{\text{gave}}{((VP/_R NP)/_R PP)^0} \quad \overset{\text{to Bach}}{PP}}{\text{gave to Bach: } VP/_R NP}\ \mathbf{FA} \quad \overset{\text{the oboe}}{NP}}{\text{gave to Bach the oboe: } VP}\ \mathbf{FA}$$

b.

$$\dfrac{\dfrac{\dfrac{\overset{\text{gave}}{((VP/_R NP)/_R PP)^0}}{\text{gave}|\text{: } ((VP/_W NP)/_R PP)^0}\ \mathbf{I} \quad \overset{\text{to Bach}}{PP}}{\text{gave}|\text{to Bach: } VP/_W NP}\ \mathbf{FA} \quad \overset{\text{the oboe}}{NP}}{\text{gave the oboe to Bach: } VP}\ \mathbf{FA}$$

Jacobson (2014a) also proposes this constituency for ditransitives to explain the weak-crossover effects in (8). In terms of Jacobson's (1999) variable-free account of pronoun binding, the aforementioned binding asymmetries are explained by virtue of the verb taking its direct object after its indirect object (the same asymmetry is observed in the heavy-NP shift version, suggesting the same constituency). Alternately, in a configurational account of binding (which is in many ways an epiphenomenon of the variable-free account), the direct object c-commands the indirect object.

(8) a. Bach gave no score$_i$ to its$_i$ owner after Friday.
b. *Bach gave her$_i$ score to no soprano$_i$ after Friday.

Note also that the combinator **I** is formalized such that it may apply to transitive verbs, though in the simplest case this step does not alter the relative order of the verb and its object (9). Such cases will henceforth be referred to as 'vacuous wrap'. It is not the case however that all applications of **I** to transitive verbs results in vacuous wrap. This mechanism is responsible for DCCG's account of the verb-particle alternation which productively modify and form both continuous and discontinuous constituents with transitive verbs (10), modulo some aspectual restrictions.

(9) a.

$$\dfrac{\dfrac{\text{compose} : VP/_R NP \qquad \text{a fugue} : NP}{O_R(\text{compose, a fugue}) : VP}\, O_R}{\text{compose a fugue} : VP}\, \textbf{FA}$$

b.

$$\dfrac{\dfrac{\dfrac{\text{compose} : VP/_R NP}{\text{compose}| : VP/_W NP}\, \textbf{I} \qquad \text{a fugue} : NP}{O_W(\text{compose}|, \text{a fugue}) : VP}\, \textbf{FA}}{\text{compose a fugue} : VP}\, O_W$$

Productive particles like *out* can be supposed to belong to $VP\$^0_i/_L VP\$^0_i$, i.e. they modify verbs with any argument structure as long as they are of the *word* sort, and give back a member of the *word* sort of the same category. Applying **I** to the verb-particle pair gives the continuous particle-verb alternate (10a), while applying **I** to the verb alone gives the discontinuous one (10b).[3]

(10) a. Bach copied/wrote/transcribed/distributed out the parts.
b. Bach copied/wrote/transcribed/distributed the parts out.

The ditransitive particle-verb data (11a)-(11b) discussed in Emonds (1976) falls out by the same mechanism,[4] while (11c) is not generated, as a discontinuous expression like a particle-verb pair can only *wrap* around one argument—O_W in (4) 'fills in' the infixation point.

(11) a. Bach sent out the parts to the singers.
b. Bach sent the parts out to the singers.
c. *Bach sent the parts to the singers out.

The interaction between heavy-NP shift and particle alternation (12) has not been discussed previously in the syntactic literature as far as I know. Strikingly, these data are entirely predicted by the present account.

(12) a. Bach sent out to the singers the parts (that he edited yesterday).
b. *Bach sent to the singers out the parts (that he edited yesterday).
c. *Bach sent to the singers the parts (that he edited yesterday) out.

Recall that the heavy-NP shift word order occurs only when **I** is never applied to the verb. Therefore, only the continuous particle-verb pair (12a) is possible, and the discontinuous ones (12b)-(12c) are correctly predicted ungrammatical.

[3]Idiosyncratic (i.e. non-productive) particle-verb pairs (e.g. look up, take on, throw out) show the same syntactic alternation as productive pairs. Under the assumption that such idiomatic expressions must be specified in the lexicon, it is necessary that each pair has two phonologies—e.g. *look*$|$ *up* and *look up*$|$. Luckily, this assumption is flawed. Rather we can postulate idiomatic lexical entries $look_2$ and up_2 which have the syntax of non-idiomatic *look* and *up*—therefore the alternation in the syntax follows as with the productive pairs. The semantics of up_2, though, is partial function defined only over $look_2$. This analysis is supported by evidence noted by Jacobson (1987) which shows the semantics of even idiosyncratic particles must be partly compositional:

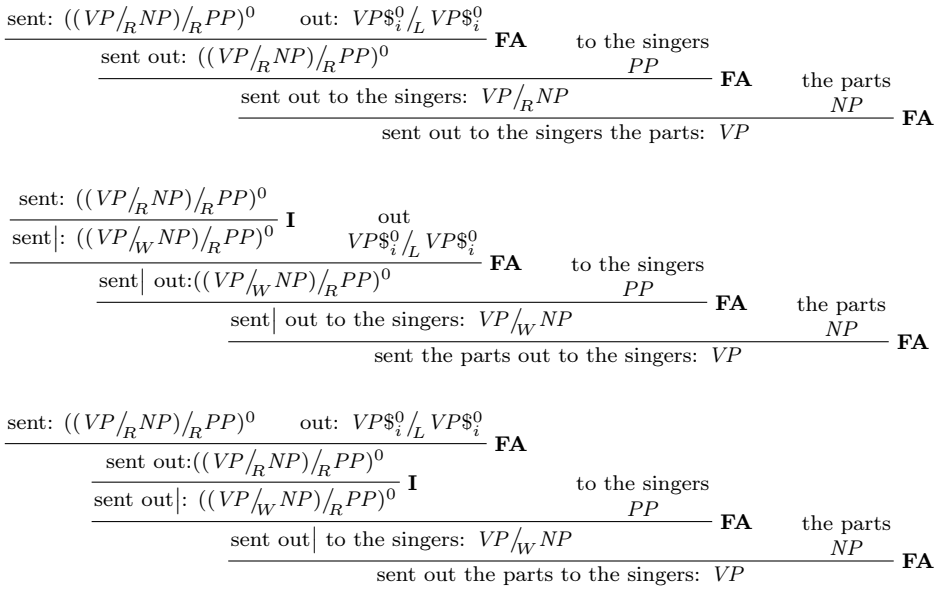
(i) John looked [this word up in the dictionary] and [that word up in the thesaurus].

[4]Emonds points out data similar to (11) concerning the interaction of ditransitives, particles, and the passive which do not clearly follow in the present analysis:

(i) a. The singers were sent out the parts.
b. *The singers were sent the parts out.

An analysis of (11b), however, requires an analysis of the passive, which is far outside the scope of this paper. I must, then, leave these data as an open question.

(13)



Additionally, **I** predicts the existence of attributive adjectives which take their arguments by *wrap*. Certainly, this is necessary for the description of 'tough'-adjectives (14a), for which a *wrap* analysis dates back to Bach (1979). Considering adjectives which take no additional arguments, however, at first blush the application of **I** appears to do no more than license vacuous *wrap*. As with transitive verbs though, there is empirical evidence that prenominal adjectives productively form discontinuous constituents, in this case with comparatives (14b).

(14) a. Bach is a tough composer to please.
(*cf. Bach is tough to please*)
b. Bach is a more creative composer than Telemann.
(*cf. Bach is more creative than Telemann*)

As we will see in §4, this particular formalization of **I**, in particular extension to cases which lead to vacuous *wrap*, makes important and empirically successful predictions in the domain of coordination.

2.3 cross-serial dependencies in Swiss German

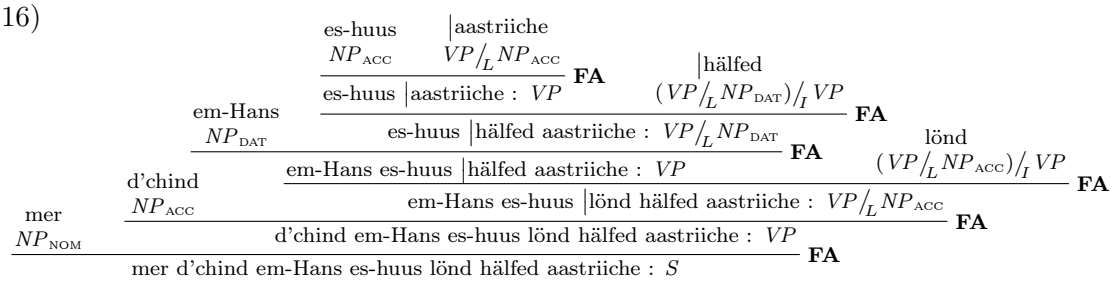
Since Pollard (1984) showed that a simple head-wrapping mechanism can successfully generate them, cross-serial dependencies in Dutch and Swiss German (15) have rightfully become a rite of passage for discontinuity calculi. Proven non-context-free (Shieber, 1985), the Swiss German construction is powerful evidence for including non-context-free operations like *wrap* in the grammar.

(15) ... mer d'chind em Hans es huus lönd hälfed aastriiche.
... we the children.ACC DAT Hans the house.ACC let help paint.
'...we let the children help Hans paint the house.'

Though a complete analysis in DCCG is outside the scope of this paper, (16) is something of a proof-of-concept. Of course, Swiss German's prosodic calculus need not be identical to English's—instead of the O_{IR} operation which provides a word with an infixation point to its right, Swiss German makes use of the O_{IL} which adds the infixation point on the left.

The general pattern is that embedded verbs are given infixation points on the left, while they are infixed into by the functors which take them. A similar analysis is proposed by Calcagno (1995).

(16)



While it may well be that Swiss German uses something like the **I** combinator, it does not appear relevant to cross-serial dependencies. Recall that **I** equips a functor with a *wrap* category and an infixation point, while in (16) it is the argument verb which requires the infixation point (*aastriiche* 'paint' has an infixation point while *lönd* 'let' does not).

Whatever the combinatory details, it is clear that with the natural addition of O_{IL}, DCCG's set of operations on strings is adequate for describing cross-serial dependencies.

3 NCC with discontinuity

3.1 the combinatory approach

Dowty's (1988) landmark account of non-constituent coordination (NCC) in a continuous CCG is among the greatest empirical successes of the CCG approach. The account relies on a syntactic use of the type-raising combinator of Partee and Rooth (1983) (henceforth **L** for 'lift') and function composition (which, following (17) Jacobson (2014a), the present account eschews in favor of its Curry-ed version—the unary Geach/Division combinator (18), henceforth **G**). However, Dowty (1997) notes that the account fails upon the introduction of canonical discontinuous constituents, as his combinators are not defined in such cases.

(17) **L** $\quad A \Rightarrow B/_L(B/_R A) \qquad A \Rightarrow B/_R(B/_L A)$

(18) **G** $\quad A/_R B \Rightarrow (A/_R C)/_R(B/_R C) \qquad A/_L B \Rightarrow (A/_L C)/_L(B/_L C)$

It is sentences like (19) which do not follow under the new set of assumptions in DCCG. In both systems, these sentences involve coordination of non-canonical constituents, however DCCG introduces the complication that part or all of the pivot canonically forms a discontinuous constituent with some of the conjunct.

(19) a. Bach **gives** [melismas to oboes] and [chorales to bassoons].
b. [Bach copied] and [Anna Magdalena sent] **the parts out**.

Considering the proof from Dowty (1988) (20), it is clear that a comparable proof is not possible yet in DCCG. The first argument into the verb must be lifted over the TV category and the second argument over the VP category (so that it takes a TV as argument). A comparable proof in DCCG would aim to *lift* the NP argument to VP/TV. However, recall that in DCCG, TV abbreviates $VP/_W NP$: Dowty's (1988) continuous version of **L** cannot produce this $/_W$.

(20) Where $TV = VP/_R PP$ and $DV = (VP/_R PP)/_R NP$:

$$\dfrac{\text{gave} : DV \qquad \dfrac{\dfrac{\text{melismas} : NP}{TV/_L DV}\,\mathbf{L} \qquad \dfrac{\dfrac{\text{to oboes} : PP}{VP/_L TV}\,\mathbf{L}}{(VP/_L DV)/_L(TV/_L DV)}\,\mathbf{G}}{VP/_L DV}\,\mathbf{FA}}{VP}\,\mathbf{FA}$$

To give **L** and **G** for the discontinuous mode, we need only adopt the intuition underlying Lambek's (1958) calculus: that the definitions of these combinators preserve word order. **L** must in effect 'switch' a functor and argument pair, while preserving their relative orders. This is accomplished by making the main slash of the *lifted* category the reverse of the original functor's, and making the two slashes of the *lifted* categories reverses of each other. So just as in Dowty's (1988) CCG where the two ways to *lift* A over B were $B/_L(B/_R A)$ and $B/_R(B/_L A)$, the new *lifts* introduced in DCCG by the discontinuous mode are $B/_I(B/_W A)$ and $B/_W(B/_I A)$.

As such, however, **L** is not guaranteed to be word-order preserving. Recall that a crucial feature of DCCG is that the pair of operations on strings in a mode is not symmetric: i.e. it is not necessarily the case that $O_R(a,\ b) = O_L(b,\ a)$

$$O_R(a|b,\ c|d) \stackrel{?}{=} O_L(c|d,\ a|b)$$
$$a|bcd \neq abc|b$$

Therefore, simply assuring that the slashes 'disagree' is not sufficient to ensure word-order preservation. An order preserving convention is easy, though, with the introduction of a new (cross-modal) feature on slashes: for all modes i, $/_i^A$ will denote a functor expression whose phonology, upon taking its argument by **FA** , is the *second* argument to the operation on strings *opposite* to i. This mechanism simply ensures that the same operation on strings combines functor and argument whether or not **L** applies. In most cases, this detail is irrelevant and omitted.

(21) **FA**
Given expressions $\alpha : \langle [\alpha]\ ;\ A/_i^A B\ ;\ \alpha' \rangle$ and $\beta : \langle [\beta]\ ;\ B\ ;\ \beta' \rangle$, infer an expression $\gamma : \langle [O_j(\beta,\ \alpha)]\ ;\ A\ ;\ \alpha'(\beta') \rangle$ where $i \neq j$ and $i, j \in \{R, L\}$ or $\{W, I\}$.

Now an order preserving **L** is possible. In addition to the modes on the slashes 'disagreeing', the superscripts must not match either.

(22) **L**
For all categories A, B with semantic types a, b respectively, given an expression $\langle [\alpha]\ ;\ A\ ;\ x \rangle$, infer an expression $\langle [\alpha]\ \ ;\ B/_i^M(B/_j^N A)\ \ ;\ \lambda F_{a\to b}[Fx] \rangle$ where $i \neq j$ and $i, j \in \{R, L\}$ or $\{W, I\}$; and $M \neq N$ and $M, N \in \{A, \emptyset\}$.

Discontinuous **G** also requires a bit of additional detail. **G** 'divides' both sides of a functor category by the same category—the first step of Curry-ed function composition. Continuous **G** ensures word-order preservation by stipulating that the introduced slashes 'agree' with the main slash of the original expression. In the discontinuous mode, however, we no longer find that it is necessary to make all the slashes' subscripts 'agree' as in the continuous version. (23)[5] shows a pair of proofs which both prove the same string from the same initial expressions, but differ in their internal details. Crucially, in (23b) *none* of the slashes following the **G** operation 'agree'. Despite this (or rather *because* of this), the same result is gotten as in (23a) which uses only **FA**.

[5]Note that operations on strings are not rules of inference in the grammar—their appearance in these proofs is purely expositional.

(23) a.

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{z} : C \qquad \text{y} : B/_L C}{\text{x}|\text{x} : A/_W B \qquad O_L(\text{y}, \text{z}) : B}\,\mathbf{FA}}{O_W(\text{x}|\text{x},\ O_L(\text{y}, \text{z})) : A}\,O_L}{O_W(\text{x}|\text{x}, \text{zy}) : A}\,O_W}{\text{xzyx} : A}$$

b.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{x}|\text{x} : A/_W B}{\text{x}|\text{x} : (A/_W C)/_R(B/_L C)}\,\mathbf{G} \qquad \text{y} : B/_L C}{O_R(\text{x}|\text{x}, \text{y}) : A/_W C}\,\mathbf{FA} \qquad \text{z} : C}{O_W(O_R(\text{x}|\text{x}, \text{y}), \text{z}) : A}\,\mathbf{FA}}{O_W(\text{x}|\text{xy}, \text{z}) : A}\,O_R}{\text{xzyx} : A}\,O_W$$

If **G** is to continue to license all and only those divisions which are order-preserving, its new definition must make this detail explicit, as this result can no longer be accomplished by stipulating that the slashes must agree. The pair of proofs in (23) suggests a new formalization. Note that they show the following:

$$O_W(x|x,\ O_L(y,\ z)) = O_W(O_R(x|x,\ y),\ z)\ .$$

Similarly, for any four (not necessarily distinct) operations O_i, O_j, O_k, O_l, if:

$$O_i(x,\ O_l(y,\ z)) = O_j(O_k(x,\ y),\ z)\ ,$$

then it must be the case that if $x \in A/_i B$, $y \in B/_l C$, and $z \in C$, then the type shift $A/_i B \Rightarrow (A/_j C)/_k(B/_l C)$ is order preserving. The new statement of **G** follows from this insight:

(24) **G**
For all categories A, B, C with respective semantic types a, b, c, given an expression $\langle [x]\ ;\ A/_i B\ ;\ F\rangle$, infer an expression $\left\langle [x]\ ;\ (A/_j C)/_k(B/_l C)\ ;\ \lambda G_{c\to b}[\lambda u_c[F(Gu)]]\right\rangle$, for all operations O_i, O_j, O_k, O_l such that for all strings $x \in A/_i B$, $y \in B/_l C$, $z \in C$, $O_i(x,\ O_l(y,\ z)) = O_j(O_k(x,\ y),\ z)$.

Having updated the combinators, NCC follows from DCCG just as before:

(25) Where $TV = VP/_W NP$ and $DV = (VP/_W NP)/_R PP$:

$$\dfrac{\dfrac{\dfrac{\text{melismas} : NP}{\text{melismas} : VP/_I TV}\,\mathbf{L}}{\text{melismas} : (VP/_I DV)/_R(TV/_L DV)}\,\mathbf{G} \qquad \dfrac{\text{to oboes} : PP}{\text{to oboes} : TV/_L DV}\,\mathbf{L}}{\text{melismas to oboes} : VP/_I DV}\,\mathbf{FA}$$

$$\dfrac{\dfrac{\text{give} : (VP/_R NP)/_R PP}{\text{give}| : DV}\,\mathbf{I} \qquad \dfrac{\begin{matrix}\vdots \\ \text{melismas to oboes} \\ VP/_I DV\end{matrix} \qquad \begin{matrix}\vdots \\ \text{and chorales to bassoons} \\ (VP/_I DV)/_L(VP/_I DV)\end{matrix}}{\text{melismas to oboes and chorales to bassoons} : VP/_I DV}\,\mathbf{FA}}{\text{give melismas to oboes and chorales to bassoons} : VP}\,\mathbf{FA}$$

3.2 the multimodal TLG approach

Discontinuity calculi in TLG are generally flexible enough to account for these data as well. Consider, for instance, Morrill's (1995) multimodal logic in which for each mode of adjunction $+_i$ (corresponding to my operations on strings), there is a corresponding pair of connectives such that:

$$s \in A/_i B \text{ iff } \forall\ s'_{\in B}[s +_i s' \in A] \qquad\qquad s \in B\backslash_i A \text{ iff } \forall\ s'_{\in B}[s' +_i s \in A].$$

Among the modes of adunction are concatenation and wrap. Therefore, this calculus effectively employs wrap while maintaining closure under all continuous non-canonical coordinations.

Dowty (1997) is of particular interest here for his use of Moortgat and Oehrle's (1994) logic to account for the same phenomena from his (1988) account in a continuous CCG. Moortgat and Oehrle (1994) achieve flexible constituency from both structural associativity rules and from slash introduction, and simulate *wrap* with structural commutativity. Moortgat and Oehrle (1994) propose a head-wrapping calculus (henceforth M&O94) in TLG which Dowty (1997) adopts to account for discontinuity in English and Dutch.

Dowty summarizes the intuition of the system in prose:

> 1. wrapping/infixing types (e.g. $VP/_W VP$; $(VP/_W NP)/VP$, etc.) combine with an argument via an abstract discontinuous mode of combination, $\bullet_w$.
> 2. A structual axiom (Mixed Commutativity) may then "commute" the $\bullet_w$ operation with respect to $\bullet$, and in doing so it "permutes" one of the operands of the first with an operand of the second [...]. By recursive use of this rule (and possibly Mixed Associativity), the original operand may 'move' some distance away from its position of origin.
> 3. All infxing/wrapping verbs are also assigned to a specific infixing sort, denoted $(A)_i$ for a verbal type A. As such they serve as a "trigger" for a sort inclusion rule: when the "permuting" element has become adjacent to the desired goal (a trigger sort or cluster) by 2., the sort interaction axiom can then eliminate the abstract mode $\bullet_w$ in favor of ordinary concatenation, simultaneously converting the wrapped/infixed element plus its verbal argument into a cluster sort. Once the linear concatenation operation has replaced $\bullet_w$, its operands are of course no longer subject to the Commutativity axiom.
> 4. A derivation is "complete" only when $\bullet_w$ modes have been eliminated, i.e. when the expression is completely linear. (In terms of Gentzen sequent derivations, only fully linear strings may be the input to analysis.)

Formally, 'Slash Elimination' and 'Slash Introduction' are paired with a prosodic calculus. In the phonological dimension, strings are fully bracketed and labeled with their prosodic sort, and every adjacent bracketing is joined with a modalized connective. The concatenation connective is simply '$\circ$'.

(26) a.
$$\dfrac{\dfrac{\langle (a \circ_i b)\ ;\ A\ ;\ \alpha' \rangle^n}{\vdots}}{\dfrac{\langle b\ ;\ B\ ;\ \beta' \rangle}{\langle a\ ;\ A/_i B\ ;\ \lambda\alpha'[\beta'] \rangle}\ /_i \mathrm{I}^n}$$

b.
$$\dfrac{\dfrac{\langle (b \circ_i a)\ ;\ A\ ;\ \alpha' \rangle^n}{\vdots}}{\dfrac{\langle b\ ;\ B\ ;\ \beta' \rangle}{\langle a\ ;\ B\backslash_i A\ ;\ \lambda\alpha'[\beta'] \rangle}\ \backslash_i \mathrm{I}^n}$$

In addition to the concatenation mode there is a *wrap* mode, with connective '$\circ_w$':

(27) a.
$$\dfrac{\langle a\ ;\ A/_i B\ ;\ \alpha' \rangle \quad \langle b\ ;\ B\ ;\ \beta' \rangle}{\langle (a \circ_i b)\ ;\ A\ ;\ \alpha'(\beta') \rangle}\ /_i \mathrm{E}$$

b.
$$\dfrac{\langle b\ ;\ B\ ;\ \beta' \rangle \quad \langle a\ ;\ B\backslash_i A\ ;\ \alpha' \rangle}{\langle (b \circ_i a)\ ;\ A\ ;\ \alpha'(\beta') \rangle}\ \backslash_i \mathrm{E}$$

A number of interaction axioms relate the concatenation and wrap modes. 'Mixed Associativity 2' allows for the rearranging of bracketings, while 'Mixed Communtativity 2'

simulates the *wrap* operation by commuting two elements.[6] Finally, the 'Inclusion' axiom eliminates the *wrap* connective when it is adjacent to an infixing word, shifting its phrasal bracketing '$(\dots)_{ph}$' to a phonological cluster bracketing '$(\dots)_c$'. These axioms have no semantic or syntactic effects, so these components are left out of their definitions.

(28) a. $\dfrac{(a \circ b) \circ_w c}{a \circ (b \circ_w c)}$ M-Assoc-2 b. $\dfrac{(a \circ b) \circ_w c}{(a \circ_w c) \circ b}$ M-Comm-2 c. $\dfrac{(a_i \circ_w b)_{ph}}{(a_i \circ b)_c}$ Incl

With these rules, the discontinuous ditransitive is highly natural. Verbs like *give* are of the infixing sort (i.e. are stored in the lexicon as, e.g. $give_i$) with category $(VP/_W NP)/_R PP$. Note that the heavy-NP shift version does not follow immediately in this system as in DCCG, though a rule like **I** is conceivable.

(29)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{give}_i : (VP/_W NP)/_R PP \qquad \text{to Telemann} : PP}{(\text{give}_i \circ \text{to Telemann}) : VP/_W NP}/_R\text{E} \qquad \text{the oboe} : NP}{((\text{give}_i \circ \text{to Telemann}) \circ_w \text{the oboe}) : VP}/_W\text{E}}{((\text{give}_i \circ_w \text{the oboe}) \circ \text{to Telemann}) : VP}\text{M-Comm-2}}{((\text{give}_i \circ \text{the oboe})_c \circ \text{to Telemann}) : VP}\text{Incl}$$

Dowty, unwittingly it seems, introduces a new mixed associaitivity axiom which I will call 'Mixed Associativity 4' ('Mixed Associativity 3' is already used by Whitman (2009)):

(30)
$$\dfrac{(\text{a} \circ_w \text{b}) \circ \text{c}}{\text{a} \circ_w (\text{b} \circ \text{c})}\text{M-Assoc-4}$$

This new rule allows for the bracketing of the non-canonical constituent *melismas to oboes*, which feeds into 'Infix-Slash Introduction'. (For notational consistency, I will continue to use $/_L$ and $/_I$ instead of $\backslash$ and $\backslash_W$. Note that in this system, $/_I$ is distinguished from $/_W$ only in that it introduces a *wrap* connective $\circ_w$ to the *left* of the functor.)

(31)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\ ]^a_i : (VP/_W NP)/_R PP \qquad \text{to oboes} : PP}{[\ ]^a_i \circ \text{to oboes} : VP/_W NP}/_R\text{E} \qquad \text{melismas} : NP}{([\ ]^a_i \circ \text{to oboes}) \circ_w \text{melismas} : VP}/_W\text{E}}{([\ ]^a_i \circ_w \text{melismas}) \circ \text{to oboes} : VP}\text{M-Comm-2}}{[\ ]^a_i \circ_w (\text{melismas} \circ \text{to oboes}) : VP}\text{M-Assoc-4}}{\text{melismas} \circ \text{to oboes} : VP/_I((VP/_W NP)/_R PP)}/_I\text{I}^a$$

We have seen then that the introduction of discontinuous constituency has a number of empirical benefits. In most cases, it does not significantly impact existing accounts of phenomena like NCC involving unbounded dependencies. As such, the proliferation of—mostly empirically successful—proposals for discontinuity calculi in CG has made it difficult to settle on a particular analysis best suited to natural language. The following section presents coordination data which suggest, at least by one metric, that DCCG most naturally and successfully describes discontinuous constituency in English.

[6] 'Mixed Associativity 1' and 'Mixed Commutativity 1' are proposed only for Dutch by Dowty (1997).

4 Right-Node Wrapping

This paper will now provide an account of a coordination phenomenon discovered independently by Wilder (1999) and Whitman (2009), known in Whitman's work under the rubric 'Right-Node Wrapping' (RNW)[7]. While terminology might suggest that RNW should be considered a separate phenomenon from RNR and NCC in general, I now presents evidence that RNW is simply a special case of NCC. Accordingly, the proposed analysis which follows requires no additional apparatus other than what is already proposed for ordinary NCC and discontinuity.

In the literature up to this point, NCC names those coordinations in which the pivot is on the periphery of the coordination (Dowty, 1988, 1997). By contrast, RNW is defined as those coordinations in which the pivot (shown in bold) is not peripheral, but rather internal to the rightmost conjunct (thus Kubota's (2014, Ms.) name 'Medial RNR'). Crucially, the material following the pivot is (by definition) a semantic argument not of both the conjuncts but rather of the right conjunct alone. Therefore, the right conjunct must be understood, at least if the correct semantics is to be gotten, as a discontinuous expression.

The discontinuous right conjunct is frequently a ditransitive verb and argument (32a)-(32b), or some sort of phrasal verb (32c)-(32d) and argument.

(32) a. [Bach fetched] and [Anna Magdalena gave **the oboe** to Telemann].
b. Bach [met] and [persuaded **Telemann** to write more fugues].
c. Bach [fetched] and [wiped **the oboe** clean].
d. [Bach edited] and [Anna Magdalena sent **the scores** out].

Indeed, Whitman (2009) cites many additional such attested sentences. Note that unequivocal cases of non-canonical constituent conjuncts as in continuous RNR/NCC are possible, e.g. (32a),(32d), where the subject and verb form a conjunct without the object.

Notably, however, the discontinuous conjunct may also consist of a verb and an optional element such as an adjunct. This case is of interest because even in a theory with discontinuous constituency, it is implausible that the discontinuous elements in (33) form canonical discontinuous constituents.

(33) a. Several years ago, in a Washington, D.C. suburb, and undercover police officer [followed] and then [shot **a young motorist** eight times]. (Whitman, 2009)
b. John [scolded] and then [eyed **his misbehaving puppy** silently].

Also consistent with NCC, RNW permits unbounded dependencies:

(34) [Bach thinks that Buxtehude speculated that Schütz fetched] and [Anna Magdalena knows that Monteverdi persuaded Lully to give **the oboe** to Telemann]

Finally, RNW appears to be generally scopally ambiguous—at least under the right context. Similarly, Kubota and Levine (forthcoming) show that (continuous) NCC is systematically scopally ambiguous for conjunction and disjunction, and for upward and downward entailing quantifiers.

(35) a. The lieutenant will either [arrest] or [shoot **every suspected arsonist** with his rifle]. (Sabbagh, 2014)

[7]Much in the way that the name 'Right-Node Raising' is still used in analyses lacking transformations such as raising, 'Right-Node Wrapping' is not intended to presuppose a *wrap* analysis (though indeed this paper advances such an analysis).

b. Carl Philip Emmanuel Bach [secretly hid] or [donated **every manuscript in his father's collection** to the library]. (Many of the former type remain lost, while the latter are well preserved.)

The $\vee > \forall$ reading in which the coordination obtains wide scope (35a) is acknowledged by all authors on RNW (Chaves, 2014; Sabbagh, 2014; Kubota, 2014, Ms.). However, the $\forall > \vee$ reading in which the pivot scopes over the coordination is judged unavailable for that same sentence by Sabbagh, and for all RNW sentences by Chaves. The judgments in cases like (35a) remain quite delicate, but Chaves's strong position does not hold up in light of cases like (35b), in which the *preferred* reading is the disputed one.

The essential conclusion from this discussion is that RNW is no more than a special case of NCC. Aside from discontinuity and its various complications, no empirical differences between RNW and NCC can be found. It follows then that RNW should fall out from adequate, independently motivated accounts of coordination and discontinuity. Indeed, this paper aims to show that just such an analysis is possible.

At this point, it is important to acknowledge that RNW is marginal for many English speakers. Such speakers, however, consistently acknowledge a marked contrast between RNW and a similar pattern—call it 'LNW'—in which the pivot *wraps* into the left conjunct (36).

(36) *[Anna Magdalena gave **the oboe** to Telemann] and [Bach tuned].

Such a sharp contrast should be taken as evidence that RNW is at the very least consistent with English grammar, while LNW is higly non-English-like. Therefore, it is a worthy enterprise to pursue a grammar which generates RNW without overgenerating LNW.[8]

4.1 RNW in DCCG

The attentive reader may have already noted that RNW follows completely for free from the version of DCCG already motivated in §2.2. While this result in itself is striking, just as significantly, the DCCG account of RNW demands a grammar with only a very limited set of available operations.

With only this mechanism in place, most cases of RNW are trivially good, as in (37). The transitive *fetched* and ditransitive *gave* are both equipped with their own infixation points by the combinatory rule **I**. The discontinuity is passed up to the constituent *and gave| to Mary* through two concatenation operations. Each of these operations is performed on exactly one discontinuous string and one continuous string, so the infixation point is always preserved. Then, crucially, the concatenation of *fetched|* with *and gave| to Mary* preserves only the second of those discontinuities because it is contained within the string with the functor category. This is due to the definition of O_L (see the steps in (37) shown in bold). This is how DCCG ensures the correct placement of the discontinuity, so that the object always infixes into the rightmost conjunct.

[8]This contrast may also be construed as evidence against including RNR, left-sided NCC, and RNW all under one rubric, as leftward sharing appears to be restricted in discontinuous coordination only. However, in the next section I show that this need not be the case: the badness of LNW may follow from the formalization of *discontinuity*, not coordination.

(37)

$$\dfrac{\text{Bach} : NP}{\text{Bach} : S/_R VP}\,\mathbf{L} \qquad \dfrac{\text{Bach} : S/_R VP}{\text{Bach} : (S/_W NP)/_R(VP/_W NP)}\,\mathbf{G} \qquad \dfrac{\text{fetched} : VP/_R NP}{\text{fetched}| : VP/_W NP}\,\mathbf{I}$$

$$\dfrac{\text{Bach} : (S/_W NP)/_R(VP/_W NP) \quad \text{fetched}| : VP/_W NP}{O_R(\text{Bach, fetched}|) : S/_W NP}\,\mathbf{FA} \qquad \dfrac{O_R(\text{Bach, fetched}|) : S/_W NP}{\text{Bach fetched}| : S/_W NP}\,O_R$$

$$\dfrac{\text{Anna} : NP}{\text{Anna} : S/_R VP}\,\mathbf{L} \qquad \dfrac{\text{Anna} : S/_R VP}{\text{Anna} : (S/_W NP)/_R(VP/_W NP)}\,\mathbf{G} \qquad \dfrac{\text{gave} : (VP/_R NP)/_R PP}{\text{gave}| : (VP/_W NP)/_R PP}\,\mathbf{I}$$

$$\dfrac{\text{gave}| : (VP/_W NP)/_R PP \quad \text{to Telemann} : PP}{O_R(\text{gave}|, \text{to Telemann}) : VP/_W NP}\,\mathbf{FA} \qquad \dfrac{O_R(\text{gave}|, \text{to Telemann}) : VP/_W NP}{\text{gave}|\text{to Telemann} : VP/_W NP}\,O_R$$

$$\dfrac{\text{Anna} : (S/_W NP)/_R(VP/_W NP) \quad \text{gave}|\text{to Telemann} : VP/_W NP}{O_R(\text{Anna, gave}|\text{to Telemann}) : VP}\,\mathbf{FA} \qquad \dfrac{O_R(\text{Anna, gave}|\text{to Telemann}) : VP}{\text{Anna gave}|\text{to Telemann} : VP}\,O_R$$

$$\dfrac{\text{and} : (X/_L X)/_R X \quad \text{Anna gave}|\text{to Telemann} : VP}{O_R(\text{and, Anna gave}|\text{to Telemann}) : (S/_W NP)/_L(S/_W NP)}\,\mathbf{FA} \qquad \dfrac{O_R(\text{and, Anna gave}|\text{to Telemann}) : (S/_W NP)/_L(S/_W NP)}{\text{and Anna gave}|\text{to Telemann} : (S/_W NP)/_L(S/_W NP)}\,O_R$$

$$\dfrac{\text{Bach fetched}| : S/_W NP \quad \text{and Anna gave}|\text{to Telemann} : (S/_W NP)/_L(S/_W NP)}{\boldsymbol{O_L(\textbf{and Anna gave}|\textbf{to Telemann, Bach fetched}|) : S/_W NP}}\,\mathbf{FA} \qquad \dfrac{\boldsymbol{O_L(\textbf{and Anna gave}|\textbf{to Telemann, Bach fetched}|) : S/_W NP}}{\textbf{Bach fetched and Anna gave}|\textbf{to Telemann} : \boldsymbol{S/_W NP}}\,\boldsymbol{O_L}$$

$$\dfrac{\text{Bach fetched and Anna gave}|\text{to Telemann} : S/_W NP \quad \text{the book} : NP}{O_W(\text{Bach fetched and Anna gave}|\text{to Telemann, the book}) : S}\,\mathbf{FA} \qquad \dfrac{O_W(\text{Bach fetched and Anna gave}|\text{to Telemann, the book}) : S}{\text{Bach fetched and Anna gave the book to Telemann} : S}\,O_W$$

Cases like (33) in which the discontinuous conjunct is a verb and its modifier which *wrap* around the object are easy to prove as well. (38) shows the proof of the right conjunct.

(38)

$$\dfrac{\text{shot} ; VP/_R NP; \mathbf{shot'}}{\text{shot}|; VP/_W NP; \mathbf{shot'}}\,\mathbf{I} \qquad \dfrac{\text{eight times} ; VP/_L VP; 8\times}{\text{eight times} ; (VP/_W NP)/_L(VP/_W NP) \; \lambda R[\lambda x[8 \times (Rx)]]}\,\mathbf{G}$$

$$\dfrac{\text{shot}|; VP/_W NP; \mathbf{shot'} \quad \text{eight times} ; (VP/_W NP)/_L(VP/_W NP) \; \lambda R[\lambda x[8 \times (Rx)]]}{\text{shot}| \text{ eight times} ; VP/_W NP; \lambda x[8 \times (\mathbf{shot'}x)]}\,\mathbf{FA}$$

$$\dfrac{\text{shot}| \text{ eight times} ; VP/_W NP; \lambda x[8 \times (\mathbf{shot'}x)] \quad \text{the victim} ; NP; \mathbf{v}}{\text{shot the victim eight times} ; VP; 8 \times (\mathbf{shot'v})}\,\mathbf{FA}$$

So RNW in DCCG appears to be no more than the interaction of the already motivated coordination and discontinuity accounts. It is no surprise then that, like (continuous) NCC, RNW is scopally ambiguous, as shown in (35). Hendriks (1993) proposes an argument lift combinator which successfully accounts for such ambiguities in canonical coordination. Such a combinator allows any part of a complex category to be 'lifted' (with an accompanying semantic lift), easily providing the desired readings for both the continuous and discontinuous non-canonical coordinations.[9]

Hudson (1976) notes cases in which RNR is possible without coordination (39).

(39) [Those who like] [outnumber those who dislike] **Bach cantatas**.

This phenomenon can be accounted for with the addition of a new combinator: Substitution (**S**), which was proposed by Steedman (1987) in his analysis of Parasitic Gaps. The syntax of the combinator comes in various forms—the 'Backwards Crossed' ($<\times\mathbf{S}$) version (40) relevant for both parasitic gaps and non-coordination RNR:

(40) $<\times\mathbf{S}$ $\qquad B/_R C \quad (A/_L B)/_R C \;\Rightarrow\; A/_R B$

[9] In fact, such a combinator is already motivated in DCCG by scopal ambiguities outside of coordination:

(i) Every organ prelude is based on some chorale.

In a number of discontinuous logics in TLG (Kubota, 2010; Moortgat, 1996), the reading in which the object scopes over the subject is obtained by way of discontinuous constituency—quantifiers can be analyzed as infixes which scope over the surrounding expression.

DCCG eschews this strategy, preferring the argument lift combinator for this reading as it is necessary for generating certain coordinations (along with a similar generalized **G**). The interested reader can verify that (ii) can be gotten by applying argument lift and then generalized **G** to the first coordination.

(ii) [Bach composed] and [Leipzig adored] [cantatas on Sunday] and [passions on Easter].

In the case of (39) $<\times\mathbf{S}$ applies to *those who like* with category $NP/_R NP$ and *outnumber those who dislike* with category $(S/_L NP)/_R NP$.

Since under the present treatment RNW is just a special case of NCC, we should expect it as well to be possible without coordination. In fact, this is a good prediction, as (41) are about as acceptable as RNW with coordination.

(41) a. [If Bach fetches], [then Anna Magdalena will give, **the oboe** to Telemann].
b. [The organist who composed], [also performed, **the prelude** eight times].

Such sentences can be gotten by adding a new syntactic variant of **S**—'Wrap Crossed Substitution' ($W\times\mathbf{S}$) (the semantics given is just the semantics of **S**).

(42) $W\times\mathbf{S}$
Given expressions $\alpha : \langle [\alpha] ; (A/_L B)/_W C ; \alpha' \rangle$ and $\beta : \langle [\beta] ; B/_W C ; \beta' \rangle$, infer an expression $\gamma : \langle [O_L(\alpha, \beta)] ; A/_W C ; \lambda c[\alpha'(c)(\beta'(c))] \rangle$.

Crucially, the functor expression α is on the right, so upon the *left-concatenation* (O_L) of the two discontinuous premises of $W\times\mathbf{S}$, it is the one which keeps its infixation point, giving the correct RNW pattern (43). As in the coordination case of RNW, it is the particular definition of O_L which ensures that the pivot wraps into the right side.

(43)
$$\dfrac{\dfrac{\dfrac{\text{the organist who composed}|: NP/_W NP \qquad \text{performed}|\ \text{eight times} : (S/_L NP)/_W NP}{O_L(\text{performed}|\ \text{eight times, the organist who composed}|) : S/_W NP}\ W\times\mathbf{S}}{\text{the organist who composed performed}|\ \text{eight times} : S/_W NP}\ O_L \qquad \text{the prelude} : NP}{\text{the organist who composed performed the prelude eight times} : S}\ \mathbf{FA}$$

Another successful prediction of the DCCG account is that discontinuous adjectives, like the *tough*-adjectives and adjective-comparative pairs, should be found in RNW coordinations. The existence of sentences like (44c) corroborate the extension of **I** to transitive predicative adjectives. It is this mechanism that allows adjectives beyond canonically discontinuous ones to form discontinuous constituents with modifiers in RNW, just as do transitive verbs.

(44) a. Bach is a [prickly] and [tough **man** to love].
b. Bach is a [prickly] but [more clever **composer** than Telemann].
c. Please move from the exit rows if you are [unwilling] or [unable **to perform the necessary actions** without injury]. (Whitman, 2009)

4.2 comparison with alternative accounts of RNW

The account of RNW in DCCG compares favorably with previous proposals (eg. Whitman, 2009; Chaves, 2014; Kubota, 2014, Ms.). In general, these accounts easily give the basic word order of the canonical case, but fail on cases with less canonical syntax or semantics. Furthermore, both previous CG accounts of RNW introduce operations far more complex than DCCG's *wrap* operation. As we shall see, then, the DCCG account is preferable both on theoretical and empirical grounds.

4.2.1 *wrap* in multimodal TLG

Whitman (2009) gives the first detailed discussion and account of RNW. His account is actually a very limited extension of Moortgat and Oehrle's (1994) logic (M&O94), i.e. the system used in Dowty's (1997) multimodal TLG account of NCC. In addition to the associative and commutative rules of Dowty's system, Whitman proposes an additional

inference rule: 'Mixed Association 3'—repeated in (45). In effect, this association rule allows for the formation of a non-canonical conjunct which takes its pivot by *wrap*.[10]

(45)
$$\frac{\text{a} \circ (\text{b} \circ_w \text{c})}{(\text{a} \circ \text{b}) \circ_w \text{c}}\ \text{M-Assoc-3}$$

The canonical case of RNW (32) is easy to show in this system. The left and right conjuncts both require a single appeal to Mixed Association 3 (see the steps in bold).

(46)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{Bach}}{NP} \quad \dfrac{\dfrac{\text{fetched}_i}{VP/_W NP} \quad \dfrac{[\ ]^a}{NP}}{\text{fetched}_i \circ_w [\ ]^a : VP}/_W\text{E}}{\mathbf{\text{Bach} \circ (\text{fetched}_i \circ_w [\ ]^a) : S}}/_L\text{E}}{\mathbf{(\text{Bach} \circ \text{fetched}_i) \circ_w [\ ]^a : S}}\ \mathbf{\text{M-Assoc-3}}}{\text{Bach} \circ \text{fetched}_i : S/_W NP}/_W\text{I}^a \quad \text{and} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{gave}_i : (VP/_W NP)/_R PP \quad \text{to Telemann} : PP}{\text{gave}_i \circ \text{to Telemann} : VP/_W NP}/_R\text{E} \quad \dfrac{[\ ]^b}{NP}}{(\text{gave}_i \circ \text{to Telemann}) \circ_w [\ ]^b : VP}/_W\text{E}}{\mathbf{\text{Anna} \circ ((\text{gave}_i \circ \text{to Telemann}) \circ_w [\ ]^b) : S}}/_W\text{E}}{\mathbf{(\text{Anna} \circ (\text{gave}_i \circ \text{to Telemann})) \circ_w [\ ]^b) : S}}\ \mathbf{\text{M-Assoc-3}}}{\text{Anna} \circ (\text{gave}_i \circ \text{to Telemann}) : S/_W NP}/_W\text{I}^b}{\text{and} \circ (\text{Anna (gave}_i \circ \text{to Telemann})) : (S/_W NP)/_L(S/_W NP)}/_R\text{E}}{(\text{Bach} \circ \text{fetched}_i) \circ (\text{and} \circ (\text{Anna (gave}_i \circ \text{to Telemann}))) : S/_W NP}/_L\text{E} \quad \dfrac{\text{the oboe}}{NP}$$

$$\dfrac{}{((\text{Bach} \circ \text{fetched}_i) \circ (\text{and} \circ (\text{Anna (gave}_i \circ \text{to Telemann})))) \circ_w \text{the oboe} : S}/_W\text{E}$$

$$\dfrac{}{(\text{Bach} \circ \text{fetched}_i) \circ ((\text{and} \circ (\text{Anna (gave}_i \circ \text{to Telemann}))) \circ_w \text{the oboe}) : S}\ \text{M-Assoc-2}$$

$$\dfrac{}{(\text{Bach} \circ \text{fetched}_i) \circ (\text{and} \circ ((\text{Anna (gave}_i \circ \text{to Telemann})) \circ_w \text{the oboe})) : S}\ \text{M-Assoc-2}$$

$$\dfrac{}{(\text{Bach} \circ \text{fetched}_i) \circ (\text{and} \circ (\text{Anna ((gave}_i \circ \text{to Telemann}) \circ_w \text{the oboe}))) : S}\ \text{M-Assoc-2}$$

$$\dfrac{}{(\text{Bach} \circ \text{fetched}_i) \circ (\text{and} \circ (\text{Anna ((gave}_i \circ_w \text{the oboe}) \circ \text{to Telemann}))) : S}\ \text{M-Comm-2}$$

$$\dfrac{}{(\text{Bach} \circ \text{fetched}_i) \circ (\text{and} \circ (\text{Anna ((gave}_i \circ \text{the oboe}_c \circ \text{to Telemann}))) : S}\ \text{Incl}$$

So the basic pattern of RNW is easy to obtain when the discontinuous element is a ditransitive verb and its indirect object. A transitive verb and particle is similarly straightforward; the proofs converge by the Mixed Association 3 steps. The relevant detail is that the particle rather than the verb is the functor, with category $TV/_L TV$.

Whitman himself notes a number of problems with his account. First, he points out that attested examples like (47) cannot be generated because the discontinuous conjunct *unable without injury* does not form a semantic unit. Rather, the rightmost modifier in that sentence modifies the pivot.

(47) Please move from the exit rows if you are [unwilling] or [unable **to perform the necessary actions** without injury].

The problem in M&O94 is with the slash-introduction inference rule, which requires the extracted element to be peripheral to the expression. The presentation of this rule results in another undergeneration not noted by Whitman. Under the usual assumption that non-particle verbal adjuncts belong to the category $VP/_L VP$ (i.e. they modify verb-*phrases*, not verbs themselves), discontinuous verb-adjunct conjuncts as found in (33) are not generated.

(48)
$$\dfrac{\dfrac{\text{shot}_i : VP/_W NP \quad \text{the victim} : NP}{\text{shot}_i \circ_w \text{the victim} : VP}/_W\text{E} \quad \text{eight times} : VP/_L VP}{(\text{shot}_i \circ_w \text{the victim}) \circ \text{eight times}}/_L\text{E}$$

By the end of this proof, there is no structural rule in M&O94 or Dowty's (1997) and Whitman's (2009) extensions which may apply, leaving no way to prove the discontinuous conjunct *shot eight times* to be of category $S/_W NP$. What is required for this inference is

[10] In fact, Dowty's (1997) analysis predicts the existence of RNW in the case where the conjuncts are canonical constituents, e.g. (32b), (32c), though Dowty himself does not acknowledge it (indeed RNW does not appear to have been known at the time).

a new rule we will call 'Inverse Mixed Commutativity 2', which, unsurprisingly, is simply the inverse of 'Mixed Commutativity 2'.[11]

(49)
$$\frac{(\text{a} \circ_w \text{b}) \circ \text{c}}{(\text{a} \circ \text{c}) \circ_w \text{b}}\ \text{Inv-M-Comm-2}$$

With the addition of this rule, a discontinuous non-canonical constituent consisting of verb and adjunct may be proven:

(50)
$$\dfrac{\dfrac{(\text{shot}_i \circ_w \text{the victim}) \circ \text{eight times}}{(\text{shot}_i \circ \text{eight times}) \circ_w \text{the victim}}\ \text{Inv-M-Comm-2}}{\text{shot}_i \circ \text{eight times}}\ {}_{/W}\text{I}$$

Whitman does not postulate this rule, as he does not recognize this undergeneration. It is possible, though he does not make this explicit, that he assumes all verb modifiers, like *eight times*, select the lexical verb itself, making the proof of (33) identical to that of particle-verb RNW. [12]

A similar proof exists for (47), requiring 'Mixed Associativity 1', a rule which Dowty and Whitman both assume is not available to English.

(51)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{unable}_i : (S[\text{A}]/_L NP)/_W VP \qquad \dfrac{[\ ]^a : VP \qquad \text{without injury} : VP\backslash_L VP}{[\ ]^a \circ \text{without injury} : VP}\ {}_{/L}\text{E}}{\text{unable}_i \circ_w ([\ ]^a \circ \text{without injury}) : S[\text{A}]/_L NP}\ {}_{/W}\text{E}}{(\text{unable}_i \circ_w [\ ]^a) \circ \text{without injury} : S[\text{A}]/_L NP}\ \text{M-Assoc-1}}{(\text{unable}_i \circ \text{without injury}) \circ_w [\ ]^a : S[\text{A}]/_L NP}\ \text{Inv-M-Comm-2}}{\text{unable}_i \circ \text{without injury} : (S[\text{A}]/_L NP)/_W VP}\ {}_{/W}\text{I}^a$$

The RNW analysis in DCCG also addresses this shortcoming of the M&O94 account. While (51) requires the addition of two new structural rules not previously considered for English, a comparable proof in DCCG can obtain such constituents with no additional apparatus (52). These proofs differ from ones like (38)—in which the rightmost part of the discontinuous constituent modifies the leftmost part as opposed to the pivot—only in that **G** applies to the functor rather than the adjunct.

(52)
$$\dfrac{\dfrac{\dfrac{\dfrac{\text{unable} \ ; (S[\text{A}]/_L NP)/_R VP; \neg\Diamond}{\text{unable}| \ ; (S[\text{A}]/_L NP)/_W VP; \neg\Diamond}\ \mathbf{I}}{\text{unable}| \ ; ((S[\text{A}]/_L NP)/_W VP)/_R(VP/_L VP) \ ; \lambda F[\lambda P[\neg\Diamond(F(P))]]}\ \mathbf{G} \quad \begin{array}{l}\text{without injury} \ ; \\ VP/_L VP; \neg\mathbf{injury'}\end{array}}{\text{unable}|\text{without injury} \ ; (S[\text{A}]/_L NP)/_W VP; \lambda P[\neg\Diamond(\neg\mathbf{injury'}(P))]}\ \mathbf{FA} \quad \begin{array}{l}\text{to perform} \ ; \\ VP \ ; \mathbf{perform'}\end{array}}{\text{unable to perform without injury} \ ; (S[\text{A}]/_L NP) \ ; \neg\Diamond(\neg\mathbf{injury'}(\mathbf{perform'}))}\ \mathbf{FA}$$

[11] Significantly, all the rules of M&O94—including this newest one—uphold the invariant that the argument taken by the functor with a '$/_W$' remains immediately to the right of the $\circ_w$, and therefore always ends up adjacent to the functor.

[12] Such an assumption may not be entirely unmotivated. One of the primary arguments for discontinuous ditransitives comes from weak crossover, e.g. (8). Similar data exist for transitive verbs with adjuncts:

(i) a. Bach composed every chorale$_i$ for its$_i$ own occasion.
b. *Bach composed it$_i$ for every chorale's$_i$ own occasion.

If the same analysis used to account for (8) is to apply here, transitive verbs must first combine with their adjuncts, and then with their objects.

Most concerning, Whitman overgenerates precisely that undesirable LNW pattern (36) which is markedly worse than RNW. Rather than appealing to M-Assoc-2 to embed the pivot deeper into the right conjunct as in (53), M-Comm-2 may apply earlier to bring the pivot adjacent to the left conjunct. There it can form a cluster next to the left verb.

(53)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\text{Bach} \circ \text{fetched}_i \quad S/_W NP} \quad \dfrac{\text{and} \quad \dfrac{\vdots}{\text{Anna (gave}_i \circ \text{to Telemann)} \quad S/_W NP}}{\text{and} \circ (\text{Anna (gave}_i \circ \text{to Telemann})) : (S/_W NP)/_L(S/_W NP)}\,/_R\text{E} \quad (X/_L X)/_R X}{(\text{Bach} \circ \text{fetched}_i) \circ (\text{and} \circ (\text{Anna (gave}_i \circ \text{to Telemann}))) : S/_W NP}\,/_L\text{E} \quad \text{the oboe} \quad NP}{((\text{Bach} \circ \text{fetched}_i) \circ (\text{and} \circ (\text{Anna (gave}_i \circ \text{to Telemann})))) \circ_w \text{the oboe} : S}\,/_W\text{E}}{((\text{Bach} \circ \text{fetched}_i) \circ_w \text{the oboe}) \circ (\text{and} \circ (\text{Anna (gave}_i \circ \text{to Telemann})))}\,\text{M-Comm-2}}{((\text{Bach} \circ (\text{fetched}_i \circ_w \text{the oboe})) \circ (\text{and} \circ (\text{Anna (gave}_i \circ \text{to Telemann})))}\,\text{M-Assoc-2}}{*((\text{Bach} \circ (\text{fetched}_i \circ \text{the oboe})_c) \circ (\text{and} \circ (\text{Anna (gave}_i \circ \text{to Telemann})))}\,\text{Incl}$$

DCCG fares better than M&O94 on this front, too. As mentioned previously, DCCG rules out such sentences by the definitions of the concatenation operations and the category of *and* which together ensure that the infixation point is inherited on the right side. Therefore, the pivot always *wrap*s into the right conjunct, never the left one.[13]

Finally, Whitman notes so-called 'RNW Sandwiches' in which multiple conjuncts take as argument the expression following the pivot (i.e. are discontinuous) and surround a conjunct which does not (54). Whitman considers these a problem for his account. However, they are closely related to better-known violations of the Coordinate Structure Constraint due to Lakoff (1986) in which a conjunct-part need not be extracted 'across the board'.

(54) Led by France and Canada, a majority of countries are asserting the right of governments to [safeguard], [promote] and even [protect **their cultures** from outside competition]. (Whitman, 2009)

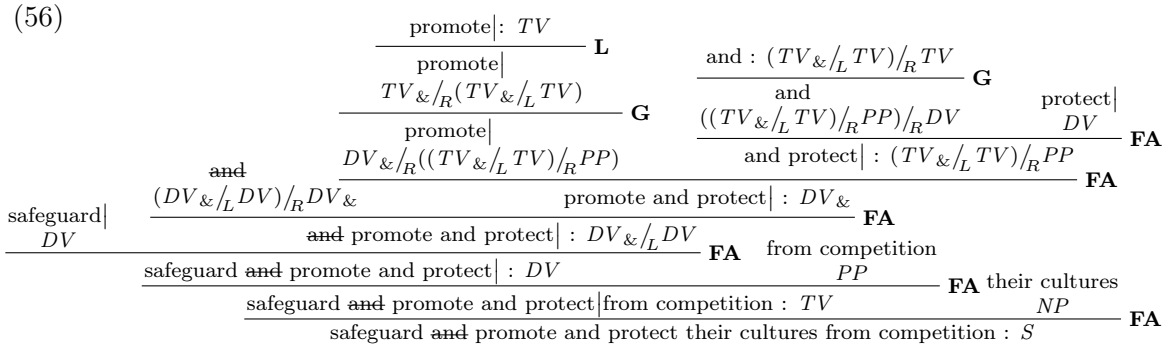
In an unpublished handout, Jacobson (2014b) notes that just the same phenomenon is possible in RNR—the pivot may belong to any subset of conjuncts which includes the final conjunct (55). She shows that in a CCG closely related to the present one, all of these facts are predicted.

(55) a. John went to the store, bought, and drank those 100 bottles of beer whose cans you see stacked on the wall.
b. *John (went to the store,) bought, drank, and fell asleep the 100 bottles of beer whose cans you see stacked on the wall.

Jacobson's analysis works for RNW sandwiches as well—i.e. they are gotten entirely for free. First, to account for iterated conjunction, the category of *and* is adjusted to $(X_{\&}/_L X)/_R X$, and a silent conjunction ~~*and*~~ with category $(X_{\&}/_L X)/_R X_{\&}$ is proposed. RNW (and RNR) sandwiches are gotten by instantiating X with the category of the 'fullest' conjunct, and appealing to **L** and **G**.

[13]Note that without the superscript A feature on slashes produced by **L**, the same overgeneration would be possible in DCCG. Since **L** shifts an argument into a functor, application of **L** to the left conjunct without adding the superscript A would cause only the left infixation point to remain in the coordination, giving LNW.

(56)



4.2.2 Linearization-Based Ellipsis in HPSG

Chaves (2014) gives an elliptical account of RNW without appealing to discontinuous constituency. Working under the rubric of Linearization-Based Ellipsis, a morpho-phonological deletion approach to NCC, Chaves proposes a new rule—'Backward Periphery Deletion' (BPD). Informally stated, this rule licenses the deletion of a prosodic phrase upon identity with a near-peripheral phrase:

(57) **BPD** $\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5 \ \Rightarrow\ \alpha_1\ \alpha_3\ \alpha_4\ \alpha_5$
for prosodic phrases $\alpha_1 - \alpha_5$ of the same prosodic sort, where $\alpha_2 = \alpha_4$.

Some cases of RNW—including non-coordination cases like (41)—follow from BPD when α_2 and α_4 are the pivot. Kubota and Levine (forthcoming) launch a detailed criticism of this elliptical approach, however. Not only does BPD fail to license cases of RNW in which the pivot scopes over the coordination (e.g. (35)), they note that it fails to do so for all NCC (barring certain *ad hoc* workarounds).

Furthermore, BPD is little more than a stipulation. Note that although LNW does not follow, a version of BPD which licenses it would be just as easy to state. By contrast, the relative badness of LNW follows deeply in DCCG from the proposed prosodic calculus for English.

4.2.3 reordering in multimodal TLCG

Kubota (2014, Ms.) provides another multimodal TLG account. Though space does not permit a detailed recounting, the essential proposal is for a 'reordering' mode with a single commutativity axiom:

(58) Reordering $A\ \circ_r\ (B\ \circ_r\ C)\ \Rightarrow\ A\ \circ_r\ (C\ \circ_r\ B)$

Generally, the A expression is a functor (or head) and the B and C expressions arguments or modifiers. So transitive and ditransitives take their arguments by the reordering mode, as do modifiers. There are additional associativity axioms. From a continuous RNR syntax (e.g. *Bach fetched and gave to Telemann the oboe*), an appeal to 'reordering' can give the RNW surface order.

Kubota blocks LNW with the addition of a 'coordination' mode which constrains association, allowing only the axioms in (59). It is the absence of cases like $(A\ \circ_i\ B)\ \circ_i\ C\ \Rightarrow\ A\ \circ_i\ (B\ \circ_i\ C)$ (call this fake rule '*M-Assoc') which makes derivations like (60), and therefore LNW, impossible.

(59) R-M-Assoc $A\ \circ_i\ (B\ \circ_c\ C)\ \Rightarrow\ (A\ \circ_i\ B)\ \circ_c\ C$ $\quad i \neq c$
L-M-Assoc $(A\ \circ_c\ B)\ \circ_i\ C\ \Rightarrow\ A\ \circ_c\ (B\ \circ_i\ C)$

(60)
$$\dfrac{\dfrac{\dfrac{((\text{gave} \circ_r \text{to Telemann}) \circ_c (\text{and} \circ_c \text{tuned})) \circ_r \text{the oboe}}{*(\text{gave} \circ_r (\text{to Telemann} \circ_c (\text{and} \circ_c \text{tuned}))) \circ_r \text{the oboe}}\ *\text{M-Assoc}}{*(\text{gave} \circ_r \text{the oboe}) \circ_r (\text{to Telemann} \circ_c (\text{and} \circ_c \text{tuned}))}\ \text{Reordering}$$

The problem with this analysis is that it crucially relies on coordination to rule out LNW, while in fact we have seen RNW is not limited to coordination alone, e.g. (41). Assuming some mechanism can be added to Kubota's system to give these non-coordination sentences, it must also constrain association like the coordination mode. On the other hand, DCCG naturally rules out in both the coordination and non-coordination cases due to the details of the prosodic calculus.

One more possible overgeneration warrants discussion: Whitman (2009), Kubota (2014, Ms.), and DCCG all predict the goodness of RNW sentences in which both conjuncts are discontinuous, e.g. (61). Such cases appear to be a problem in general for the CG analysis of RNW. Supposing that RNW is licensed by virtue of both conjuncts having *wrap* or reordering categories, it would be difficult (in the syntax) to also prevent cases in which the left conjunct is not a vacuous *wrap*-per. However, it is not clear that these sentences are strictly ungrammatical: a few informants accepted them with reservations.

(61) a. ?Bach [took from Anna Magdalena] and [gave **the oboe** to Telemann].
b. ?Bach [copied out] and [sent **the scores** off].

Even an elliptical account like Chaves's (2014) is subject to the same problem, as his ellipsis operation would apply in cases like (62):

(62) Bach [took from Anna Magdalena ~~the oboe~~] and [gave **the oboe** to Telemann].

4.3 other discontinuity calculi and RNW

Here I consider a taxonomy for discontinuous logics, which makes four binary distinctions—whether or not the logic:

1) allows discontinuity anywhere inside a string,

2) assigns discontinuous categories to certain words,

3) permits multiple discontinuities within a single string, or

4) associates certain phonological effects with discontinuity.

I argue now that the particular features possessed by the proposed calculus in this paper, DCCG, conspire to allow it to account for RNW.

DCCG does not allow discontinuity freely; it only assigns discontinuity via the combinatory rule **I** to a very limited set of categories. Other systems like Moortgat's (1988) extension of the Lambek calculus, Kubota's (2010) Hybrid Type-Logical Grammar, and Morrill and Valentín's (2012) Displacement Calculus do allow discontinuity freely—i.e. they have the property that for $i = W$ or the relevant modality, a string $s \in A/_i B$ iff $\exists s_1, s_2[s = s_1 + s_2 \ \wedge \ \forall s'_{\in B}[s_1 + s' + s_2 \in A]]$. I will show that such generality leads to serious overgeneration.

Any account of RNW in such a system—that is the proposal of some mechanism to rule out the ungrammatical LNW case—must necessarily advance a very strong version of the Lambek Coordination Closure Hypothesis discussed by Dowty (1997). This hypothesis originally conjectures that any continuous substring of a well-formed string may appear as a conjunct in NCC, a position which follows from most TLG accounts of NCC. This stance

becomes highly suspect when extended to a general discontinuous logic. Such systems must also predict that the discontinuous rightmost conjunct may be any substring of a sentence with a single discontinuity. Contrary to this prediction, some discontinuous strings clearly do not appear to make well-formed conjuncts (i.e. constituents):

(63) *[Bach wrote the fugue that], and [Herreweghe thinks that **Harnoncourt performed** the prelude, too].
(*cf.* [*Bach wrote the fugue that Harnoncourt performed*] *and* [*Herreweghe thinks that Harnoncourt performed the prelude, too*].)

On the basis of such evidence, such general discontinuity must be dismissed under the assumption that all discontinuous constituency can be subsumed under a single mechanism.

DCCG advances a weaker version of the Coordination Closure Hypothesis: namely that all discontinuous constituents can appear as discontinuous right conjuncts so long as the infixation point is directly following a word of a category that **I** is defined over. Since **I** does not apply to complementizers like *that*, (63) is correctly predicted ungrammatical.

This same evidence supports the position taken by DCCG that discontinuity is assigned at the word level. We have only seen discontinuity in RNW at the locus of transitive verbs, ditransitive verbs, and adjectives. Furthermore, we surveyed considerable evidence in §2.2 that words—though not necessarily just lexical items, as we saw with particle/verb pairs—of some categories appear to 'trigger' discontinuity, a fact which seems to demand a lexical rule like **I** or a property of lexical organization which manages discontinuity.

Third, DCCG permits only a single discontinuity within a string. This feature is motivated by RNW as well. We saw that systems which allows multiple discontinuities like M&O94 as extended by Whitman (2009), are prone to overgenerating the LNW case. This restriction of DCCG is precisely the feature which allows it to avoid LNW, as the conventions for passing up infixation points guarantees that the infixation point in a coordination will end up in the rightmost conjunct.[14]

It is not clear how a calculus which allows multiple discontinuities like Hybrid Type-Logical Categorial Grammar would avoid LNW. It seems that any such analysis would require the concatenation of two lambda expressions, which is not necessarily a well-defined operation. Morrill and Valentín's (2012) Displacement Calculus **D** defines a right-*wrap* operation on strings with any number $k > 0$ of discontinuities which replaces the k^{th} discontinuity, i.e. the rightmost one, in a string with an infix expression. Such an operation would allow multiple discontinuous conjuncts (vacuously or otherwise) to wrap around the pivot at the rightmost conjuct—just what is needed to get RNW without LNW. However, this logic is still too general, overgenerating strings like (63).

Finally, DCCG associates no strong phonological effects with discontinuity. It is possible that wrap is associated with some sort of encliticization process, as suggested by Moortgat and Oehrle (1994) and Dowty (1997), however this is not an essential feature of DCCG.

[14] A strikingly similar convention is settled on by Calcagno (1995), who considers a large set of possible concatenation and head-*wrap* operations. Instead of infixation points assigned to words, this logic distinguishes a *word* as the *head* of an expression. Left and right concatenation of two headed strings can preserve the head of the first or the second argument of the operation. DCCG uniformly preserves the first argument's. An expression may *wrap* to the left of the head or the right of the head in much the same way that O_{IR} and O_{IL} place infixation points to the right and left of words. Calcagno also considers that either the functor's head or the argument's head may become the head of the string following *wrap*. All *wrap* operations in DCCG are simply right-*wrap* in which the infix's infixation point (if it has one) is preserved. A calculus like Calcagno's could presumably give a natural account of RNW.

Certainly, a strongly phonological system like Hoeksema and Janda's (1988) is inadequate to describe RNW. Their *wrap* operations always split a prosodic structure after its first element or before its last one. Inconsistent with such an operation, we have found numerous counterexamples, e.g. (32a), showing that infixation points in discontinuous constituents need not be next to the left periphery.

In order to account for RNW (and other discontinuity phenomena), then, it seems that a discontinuity calculus must at least share many of the properties of DCCG. It remains to be seen whether more general systems like Morrill and Valentín's (2012) and Kubota and Levine's (forthcoming) can effectively unify their particular notions of discontinuity with that found in RNW.

4.4 Discussion on complexity

M&O94 and Kubota's (2014, Ms.) multimodal logic rely on structural commutativity and associativity rules to simulate wrap and to generate RNW. These inference rules map one structured string to another structured string based on details of those those structures. In this way, structural rules are closely related to some kinds of movement operations in, e.g. the GB framework, which map an input tree to an output tree.[15]

Unlike these approaches, DCCG is still able to achieve discontinuity via the relatively simple *wrap* operation. Crucially, *wrap* requires the grammar to keep track of at most one internal detail of a string: the location of its single infixation point. This operations compares favorably with both M&O94's structural rules and systems which permit multiple discontinuities (e.g. Kubota, 2010; Morrill and Valentín, 2012), all of which demand an indefinite number of details to be 'remembered'. Particularly, the structural commutativity and associativity rules which M&O94 uses to simulate *wrap* (as well as the associativity rules) requires every string in the proof to be fully bracketed—i.e. a completely hierarchical structure like a tree (but with only the root category label). Therefore, structural rules, just like movement, are really mappings from one tree structure to another. In this regard, *wrap* in M&O94 entirely misses one of the great advantages of Bach's (1979) original *wrap* operation—namely that it achieved discontinuous constituencies without appealing to movement. DCCG preserves this original property of *wrap*, representing a much simpler conception of the grammar.

4.5 Gapping

So far, there has been no mention in this paper of Gapping (64a) and Psuedogapping (64b)—surely the most well-studied cases of the interaction of discontinuity with coordination.

(64) a. [Bach **likes** oboes] and [Telemann recorders].
b. [Bach has **written** more cantatas] than [Telemann has Passions].

Gapping's most concerning detail is that it appears to be exactly the LNW pattern in which the 'pivot' is found only in the left conjunct, and the avoidance of which was such

[15] M&O94 could actually be weakly equivalent to a wrap grammar. Four features of this calculus limit the kinds of strings it can generate: 1) The commutativity axioms are licensed only when there is a wrap connective, 2) commutativity is designed such that an infix always stays on the same side of the wrap connective it introduces, 3) no 'derivation' is complete until all wrap connectives are eliminated, and 4) wrap connectives are only eliminated by heads, so commutativity will never have any effect but 'transporting' the infixed argument to the head. Nonetheless, the intermediate stages in licensing a string may be quite different from those of a grammar with true *wrap*.

an essential part of the analysis of RNW in DCCG. In fact, this is not a concern, I believe, since there is good evidence that Gapping is a distinct phenomenon from NCC.

The elegance of the CG approach to NCC lies in the fact that it reduces the phenomenon simply to a case of coordination of like-categories. The same is true of DCCG's account of RNW. However, Gapping is considerably more restricted than we would expect were it like-category coordination. Gapping gaps always begin with verbs, never with objects (65) as we see in RNW.

(65) *[John told **Mary** that he loved her] and [Bill told that he would die for her]. (Hudson, 1976)

In order for DCCG as currently formulated to generate the discontinuous right conjunct of a Gapping coordination, **I** would have to be extended to apply to subjects (of category $S/_R VP$) in order to give them the necessary infixation point. Unlike all previously discussed cases in which **I** introduces discontinuity, such an extension has no external motivation.

Even stronger empirical evidence against considering Gapping a part of RNW is that Gapping allows mismatches in number agreement, while NCC does not:

(66) a. [Bach **composes** fugues] and [his sons symphonies].
b. *[On Sundays Bach's sons] and [on Tuesdays Bach] **composes fugues**.

Similarly, Pseudogapping permits mismatches in tense/aspect (67b), while Chaves (2014) argues that similar data like (67b) are cases of cataphoric VP-ellipsis, not of true RNR.

(67) a. [Bach **wrote** more cantatas] than [Telemann has Passions].
b. [Bach will] and [Telemann has] **composed a fugue**.

This is not a paper on Gapping, so I will not propose an analysis within the system. However, it is not a concern that gapping does not follow from the same mechanism as RNW. In the latter case, the pivot is taken as a syntactic (and semantic) argument by the coordinated expressions. In Gapping, on the other hand, the above inflectional mismatch data suggest that the pivot may not be a syntactic argument of the entire coordination. Hopefully, it is possible to advance an analysis for VP-ellipsis, Pseudogapping, and Gapping alike in which the semantics of the gapped or elided element is applied to the reduced conjunct anaphorically.

5 Conclusion

Strikingly, DCCG—simple though it is—straightforwardly accounts for nearly the full range of RNW data surveyed in this paper. In fact, the empirical successes of DCCG pertaining to RNW are even greater than those of more complex systems surveyed. In addition to this, it successfully accounts for binding asymmetries, particle verbs, heavy-NP shift, *tough*-adjectives, continuous NCC, and cross-serial dependencies all under a single discontinuity calculus. While ultimately it is possible that not all the proposed cases of discontinuous constituency to be found in the literature can be described with a unified notion of discontinuous constituency, the tentative success of the present approach supports the position that DCCG's particular notion of constituency is indeed much like that of natural language.

References

Ajdukiewicz, Kazimierz. 1935. Die syntaktische konnexitat. studia philosophica, 1: 1-27. english translation in storrs mccall, ed., polish logic: 1920-1939, 207-231.

Bach, Emmon. 1979. Control in montague grammar. *Linguistic inquiry* 515–531.

Bar-Hillel, Yehoshua. 1953. A quasi-arithmetical notation for syntactic description. *Language* 47–58.

Calcagno, Mike. 1995. A sign-based extension to the lambek calculus for discontinuous constituency. *Logic Journal of IGPL* 3(4):555–578.

Chaves, Rui P. 2014. On the disunity of right-node raising phenomena: Extraposition, ellipsis, and deletion. *Language* 90(4):834–886.

Dowty, David. 1988. Type raising, functional composition, and non-constituent conjunction. In *Categorial grammars and natural language structures*, 153–197. Springer.

Dowty, David. 1997. Non-constituent coordination, wrapping, and multimodal categorial grammars. In *Structures and norms in science*, 347–368. Springer.

Hendriks, Herman. 1993. Studied flexibility. *Institute for Logic, Language and Computation, Universiteit van Amsterdam. ILLC Dissertation Series* 5.

Hoeksema, Jacob. 1984. *Categorial morphology*. Ph.D. thesis, University of Groningen.

Hudson, Richard A. 1976. Conjunction reduction, gapping, and right-node raising. *Language* 535–562.

Jacobson, Pauline. 1987. Phrase structure, grammatical relations, and discontinuous constituents in discontinuous constituency. *Syntax and semantics* 20:27–69.

Jacobson, Pauline. 1999. Towards a variable-free semantics. *Linguistics and Philosophy* 22(2):117–185.

Jacobson, Pauline. 2014a. *Compositional semantics: An introduction to the syntax/semantics interface*. Oxford University Press.

Jacobson, Pauline. 2014b. Remarks on coordination: Or, (part of) why i love categorial grammar.

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

Kubota, Yusuke. 2014, Ms. Medial right-node raising and multi-modal categorial grammar .

Kubota, Yusuke and Robert Levine. forthcoming. Against ellipsis: Arguments for the direct licensing of 'non-canonical' coordinations. *Linguistics and Philosophy* .

Lakoff, George. 1986. *Frame semantic control of the coordinate stucture constraint*. Cognitive Science Program, Institute of Cognitive Studies, University of California at Berkeley.

Lambek, Joachim. 1958. The mathematics of sentence structure. *American mathematical monthly* 154–170.

Moortgat, Michael. 1988. *Categorial investigations: Logical and linguistic aspects of the Lambek calculus*. No. 9. Walter de Gruyter.

Moortgat, Michael. 1996. Multimodal linguistic inference. *Journal of Logic, Language and Information* 5(3-4):349–385.

Moortgat, Michael and Richard Oehrle. 1994. Adjacency, dependency and order. In *Proceedings 9th Amsterdam Colloquium*, 447–466.

Morrill, Glyn. 1995. Discontinuity in categorial grammar. *Linguistics and Philosophy* 18(2):175–219.

Morrill, Glyn and Oriol Valentín. 2012. Generalized discontinuity. In *Formal Grammar*, 146–161. Springer.

Partee, Barbara and Mats Rooth. 1983. Generalized conjunction and type ambiguity. *Formal Semantics: The Essential Readings* 334–356.

Pollard, Carl. 1984. Generalized context-free grammars, head grammars and natural language. *Stanford University dissertation* .

Sabbagh, Joseph. 2014. Right node raising. *Language and Linguistics Compass* 8(1):24–35.

Shieber, Stuart M. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 333–343.

Steedman, Mark. 1985. Dependency and coördination in the grammar of dutch and english. *Language* 523–568.

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

Steedman, Mark. 2000. *The syntactic process*, vol. 35. MIT Press.

Whitman, Neal. 2009. Right-node wrapping: Multimodal categorial grammar and the "friends in low places" coordination. *Theory and evidence in semantics* 235–256.

Wilder, Chris. 1999. Right node raising and the lca. In *Proceedings of WCCFL*, vol. 18, 586–598.

Zwicky, Arnold M. 1992. Some choices in the theory of morphology. *Formal grammar: Theory and implementation* 327–371.