A Logical Basis for the D Combinator and Normal Form in CCG

Frederick Hoyt and Jason Baldridge
The Department of Linguistics
The University of Texas at Austin
{fmhoyt,jbaldrid}@mail.utexas.edu

Abstract

The standard set of rules defined in Combinatory Categorial Grammar (CCG) fails to provide satisfactory analyses for a number of syntactic structures found in natural languages. These structures can be analyzed elegantly by augmenting CCG with a class of rules based on the combinator **D** (Curry and Feys, 1958). We show two ways to derive the **D** rules: one based on unary composition and the other based on a logical characterization of CCG's rule base (Baldridge, 2002). We also show how Eisner's (1996) normal form constraints follow from this logic, ensuring that the **D** rules do not lead to spurious ambiguities.

1 Introduction

Combinatory Categorial Grammar (CCG, Steedman (2000)) is a compositional, semantically transparent formalism that is both linguistically expressive and computationally tractable. It has been used for a variety of tasks, such as wide-coverage parsing (Hockenmaier and Steedman, 2002; Clark and Curran, 2007), sentence realization (White, 2006), learning semantic parsers (Zettlemoyer and Collins, 2007), dialog systems (Kruijff et al., 2007), grammar engineering (Beavers, 2004; Baldridge et al., 2007), and modeling syntactic priming (Reitter et al., 2006).

A distinctive aspect of CCG is that it provides a very flexible notion of constituency. This supports elegant analyses of several phenomena (e.g., coordination, long-distance extraction, and intonation) and allows incremental parsing with the competence grammar (Steedman, 2000). Here, we argue that even with its flexibility, CCG as standardly defined is not permissive enough for certain linguistic constructions and greater incrementality. Following Wittenburg (1987), we remedy this by adding a set of rules based on the D combinator of combinatory logic (Curry and Feys, 1958).

(1) $\mathsf{x/(y/z)}{:}f \quad \mathsf{y/w}{:}g \;\Rightarrow\; \mathsf{x/(w/z)}{:}\lambda h.f(\lambda x.ghx)$

We show that CCG augmented with this rule improves CCG's empirical coverage by allowing better analyses of modal verbs in English and causatives in Spanish, and certain coordinate constructions.

The **D** rules are well-behaved; we show this by deriving them both from unary composition and from the logic defined by Baldridge (2002). Both perspectives on **D** ensure that the new rules are compatible with normal form constraints (Eisner, 1996) for controlling spurious ambiguity. The logic also ensures that the new rules are subject to modalities consistent with those defined by Baldridge and Kruijff (2003). Furthermore, we define a logic that produces Eisner's constraints as grammar internal theorems rather than parsing stipulations.

2 Combinatory Categorial Grammar

CCG uses a universal set of syntactic rules based on the **B**, **T**, and **S** combinators of combinatory logic (Curry and Feys, 1958):

(2) **B**: $((\mathbf{B}f)g)x = f(gx)$
T: $\mathbf{T}xf = fx$
S: $((\mathbf{S}f)g)x = fx(gx)$

CCG functors are functions over strings of symbols, so different linearized versions of each of the combinators have to be specified (ignoring **S** here):

(3)
$$\begin{array}{llllll}
\mathsf{FA}: & (>) & \mathsf{x}/_{\star}\mathsf{y} & \mathsf{y} & \Rightarrow & \mathsf{x}\\
 & (<) & \mathsf{y} & \mathsf{x}\backslash_{\star}\mathsf{y} & \Rightarrow & \mathsf{x}\\
\mathbf{B}: & (>\mathbf{B}) & \mathsf{x}/_{\diamond}\mathsf{y} & \mathsf{y}/_{\diamond}\mathsf{z} & \Rightarrow & \mathsf{x}/_{\diamond}\mathsf{z}\\
 & (<\mathbf{B}) & \mathsf{y}\backslash_{\diamond}\mathsf{z} & \mathsf{x}\backslash_{\diamond}\mathsf{y} & \Rightarrow & \mathsf{x}\backslash_{\diamond}\mathsf{z}\\
 & (>\mathbf{B}_{\times}) & \mathsf{x}/_{\times}\mathsf{y} & \mathsf{y}\backslash_{\times}\mathsf{z} & \Rightarrow & \mathsf{x}\backslash_{\times}\mathsf{z}\\
 & (<\mathbf{B}_{\times}) & \mathsf{y}/_{\times}\mathsf{z} & \mathsf{x}\backslash_{\times}\mathsf{y} & \Rightarrow & \mathsf{x}/_{\times}\mathsf{z}\\
\mathbf{T}: & (>\mathbf{T}) & \mathsf{x} & \Rightarrow & \mathsf{t}/_{i}(\mathsf{t}\backslash_{i}\mathsf{x}) & \\
 & (<\mathbf{T}) & \mathsf{x} & \Rightarrow & \mathsf{t}\backslash_{i}(\mathsf{t}/_{i}\mathsf{x}) &
\end{array}$$

The symbols $\{\star, \diamond, \times, \cdot\}$ are modalities that allow subtypes of slashes to be defined; this in turn allows the slashes on categories to be defined in a way that allows them to be used (or not) with specific subsets of the above rules. The rules of this multimodal version of CCG (Baldridge, 2002; Baldridge and Kruijff, 2003) are derived as theorems of a Categorial Type Logic (CTL, Moortgat (1997)).

This treats CCG as a compilation of CTL proofs, providing a principled, grammar-internal basis for restrictions on the CCG rules, transferring language-particular restrictions on rule application to the lexicon, and allowing the CCG rules to be viewed as grammatical universals (Baldridge and Kruijff, 2003; Steedman and Baldridge, To Appear).

These rules—especially the $\mathbf{B}$ rules—allow derivations to be *partially* associative: given appropriate type assignments, a string ABC can be analyzed as either A(BC) or (AB)C. This associativity leads to elegant analyses of phenomena that demand more effort in less flexible frameworks. One of the best known is "odd constituent" coordination:

(4) Bob gave **Stan a beer** and **Max a coke**.

(5) **I will buy** and **you will eat** a cheeseburger.

The coordinated constituents are challenging because they are at odds with standardly assumed phrase structure constituents. In CCG, such constituents simply follow from the associativity added by the $\mathbf{B}$ and $\mathbf{T}$ rules. For example, given the category assignments in (6) and the abbreviations in (7), (4) is analyzed as in (8) and (9). Each conjunct is a pair of type-raised NPs combined by means of the $>\mathbf{B}$-rule, deriving two composed constituents that are arguments to the conjunction:[1]

(6)
i. **Bob** $\vdash \mathsf{s}/(\mathsf{s}\backslash\mathsf{np})$
ii. **Stan**, **Max** $\vdash$ $((\mathsf{s}\backslash\mathsf{np})/\mathsf{np})\backslash(((\mathsf{s}\backslash\mathsf{np})/\mathsf{np})/\mathsf{np})$
iii. **a beer**, **a coke** $\vdash (\mathsf{s}\backslash\mathsf{np})\backslash((\mathsf{s}\backslash\mathsf{np})/\mathsf{np})$
iv. **and** $\vdash (\mathsf{x}\backslash_{\star}\mathsf{x})/_{\star}\mathsf{x}$
v. **gave** $\vdash ((\mathsf{s}\backslash\mathsf{np})/\mathsf{np})/\mathsf{np}$

(7)
i. $\mathsf{vp} = \mathsf{s}\backslash\mathsf{np}$
ii. $\mathsf{tv} = (\mathsf{s}\backslash\mathsf{np})/\mathsf{np}$
iii. $\mathsf{dtv} = ((\mathsf{s}\backslash\mathsf{np})/\mathsf{np})/\mathsf{np}$

(8)
$$\begin{array}{ccccc}
\underline{\textbf{Stan}} & \underline{\textbf{a beer}} & \underline{\textbf{and}} & \underline{\textbf{Max}} & \underline{\textbf{a coke}}\\
\mathsf{tv\backslash dt} & \mathsf{vp\backslash tv} & (\mathsf{x}\backslash_{\star}\mathsf{x})/_{\star}\mathsf{x} & \mathsf{tv\backslash dt} & \mathsf{vp\backslash tv}\\
\multicolumn{2}{c}{\overline{\quad\mathsf{vp\backslash dt}\quad}^{<\mathbf{B}}} & & \multicolumn{2}{c}{\overline{\quad\mathsf{vp\backslash dt}\quad}^{<\mathbf{B}}}\\
 & & \multicolumn{3}{c}{\overline{\quad(\mathsf{vp\backslash dt})\backslash(\mathsf{vp\backslash dt})\quad}^{>}}\\
\multicolumn{5}{c}{\overline{\qquad\qquad\mathsf{vp\backslash dt}\qquad\qquad}^{<}}
\end{array}$$

(9)
$$\begin{array}{ccc}
\underline{\textbf{Bill}} & \underline{\textbf{gave}} & \underline{\textbf{Stan a beer and Max a coke}}\\
\mathsf{s/vp} & \mathsf{dt} & \mathsf{vp\backslash dt}\\
 & \multicolumn{2}{c}{\overline{\qquad\mathsf{vp}\qquad}^{<}}\\
\multicolumn{3}{c}{\overline{\qquad\qquad\mathsf{s}\qquad\qquad}^{>}}
\end{array}$$

Similarly, *I will buy* is derived with category $\mathsf{s/np}$ by assuming the category (6i) for *I* and composing that with both verbs in turn.

CCG's approach is appealing because such constituents are not odd at all: they simply follow from the fact that CCG is a system of type-based grammatical inference that allows left associativity.

3 Linguistic Motivation for D

CCG is only *partially* associative. Here, we discuss several situations which require greater associativity and thus cannot be given an adequate analysis with CCG as standardly defined. These structures have in common that a category of the form $\mathsf{x}|(\mathsf{y}|\mathsf{z})$ must combine with one of the form $\mathsf{y}|\mathsf{w}$—exactly the configuration handled by the $\mathbf{D}$ schemata in (1).

3.1 Cross-Conjunct Extraction

In the first situation, a question word is distributed across auxiliary or subordinating verb categories:

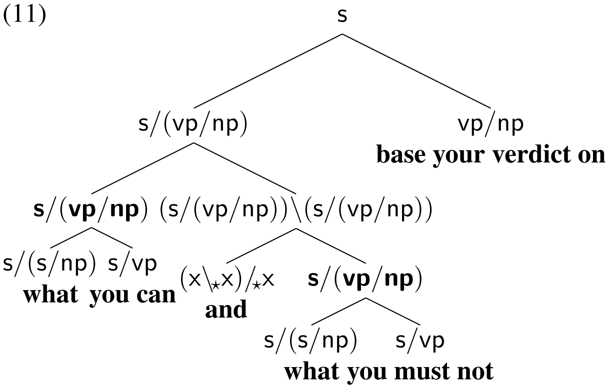
(10) ...**what you can** and **what you must not** base your verdict on.

We call this *cross-conjunct extraction.* It was noted by Pickering and Barry (1993) for English, but to the best of our knowledge it has not been treated in the

[1] We follow (Steedman, 2000) in assuming that type-raising applies in the lexicon, and therefore that nominals such as *Stan* have type-raised lexical assignments. We also suppress semantic representations in the derivations for the sake of space.

CCG literature, nor noted in other languages. The problem it presents to CCG is clear in (11), which shows the necessary derivation of (10) using standard multimodal category assignments. For the tokens of *what* to form constituents with *you can* and *you must not*, they must must combine directly. The problem is that these constituents (in bold) cannot be created with the standard CCG combinators in (3).

(11)



The category for *and* is marked for non-associativity with $\star$, and thus combines with other expressions only by function application (Baldridge, 2002). This ensures that each conjunct is a discrete constituent.

Cross-conjunct extraction occurs in other languages as well, including Dutch (12), German (13), Romanian (14), and Spanish (15):

(12) ***dat ik haar wil*** *en* ***dat ik haar moet*** *helpen.*
that I her want and that I her can help
"...that I want to and that I can help her."

(13) ***Wen kann ich*** *und* ***wen darf ich*** *noch* ***wählen****?*
who can I and who may I still choose
"Whom can I and whom may I still chose?"

(14) *Gandeste-te* ***cui*** ***çe*** *vrei,*
consider.imper.2s-refl.2s who.dat what want.2s
şi ***cui*** ***çe*** *poţi, să* ***dai.***
and who.dat what can.2s to give.subj.2s
"Consider to whom you want and to whom you are able to give what."

(15) ***Me lo puedes*** *y* ***me lo debes*** *explicar*
me it can.2s and me it must.2s ask
"You can and should explain it to me."

It is thus a general phenomenon, not just a quirk of English. While it could be handled with extra categories, such as (s/(vp/np))/(s/np) for *what*, this is exactly the sort of strong-arm tactic that inclusion of the standard **B**, **T**, and **S** rules is meant to avoid.

3.2 English Auxiliary Verbs

The standard CCG analysis for English auxiliary verbs is the type exemplified in (16) (Steedman, 2000, 68), interpreted as a unary operator over sentence meanings (Gamut, 1991; Kratzer, 1991):

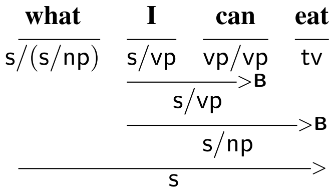
(16) **can** $\vdash$ (s\np)/(s\np) : $\lambda P_{et}\lambda x.\Diamond P(x)$

However, this type is empirically underdetermined, given a widely-noted set of generalizations suggesting that auxiliaries and raising verbs take no subject argument at all (Jacobson, 1990, a.o.).

(17)
i. Lack of syntactic restrictions on the subject;
ii. Lack of semantic restrictions on the subject;
iii. Inheritance of selectional restrictions from the subordinate predicate.

Two arguments are made for (16). First, it is necessary so that type-raised subjects can compose with the auxiliary in extraction contexts, as in (18):

(18)



Second, it is claimed to be necessary in order to account for subject-verb agreement, on the assumption that agreement features are domain restrictions on functors of type s\np (Steedman, 1992, 1996).

The first argument is the topic of this paper, and, as we show below, is refuted by the use of the **D**-combinator. The second argument is undermined by examples like (19):

(19) There **appear** to have been [neither [**any catastrophic consequences**], nor [**a drastic change** in the average age of retirement]] .

In (19), *appear* agrees with two negative-polarity-sensitive NPs trapped inside a *neither-nor* coordinate structure in which they are licensed. *Appear* therefore does not combine with them directly, showing that the agreement relation need not be mediated by direct application of a subject argument.

We conclude, therefore, that the assignment of the vp/vp type to English auxiliaries and modal verbs is unsupported on both formal and linguistic grounds.

Following Jacobson (1990), a more empirically-motivated assignment is (20):

(20) **can** $\vdash \mathsf{s/s} : \lambda p_t.\Diamond p$

Combining (20) with a type-raised subject presents another instance of the structure in (1), where that question words are represented as variable-binding operators (Groenendijk and Stokhof, 1997):

(21)
$$\begin{array}{ccc}
\textbf{what} & \textbf{I} & \textbf{can} \\
\hline
\mathsf{s/(s/np)} : \lambda Q_{et}?yQy & \mathsf{s/vp} : \lambda P_{et}.Pi' & \mathsf{s/s} : \lambda p_t.\Diamond p \\
 & *\!*\!*\!\longrightarrow_{>\mathbf{B}} *\!*\!* &
\end{array}$$

3.3 The Spanish Causative Construction

The schema in (1) is also found in the widely-studied Romance causative construction (Andrews and Manning, 1999, a.m.o), illustrated in (22):

(22) *Nos hizo leer El Señor de los Anillos.*
cl.1p made.3s read the Lord of the Rings
"He made us read *The Lord of the Rings*."

The aspect of the construction that is relevant here is that the causative verb *hacer* appears to take an object argument understood as the subject or agent of the subordinate verb (the *causee*). However, it has been argued that Spanish causative verbs do not in fact take objects (Ackerman and Moore, 1999, and refs therein). There are two arguments for this.

First, syntactic alternations that apply to object-taking verbs, such as passivization and periphrasis with subjunctive complements, do not apply to *hacer* (Luján, 1980). Second, *hacer* specifies neither the case form of the causee, nor any semantic entailments with respect to it. These are instead determined by syntactic, semantic, and pragmatic factors, such as transitivity, word order, animacy, gender, social prestige, and referential specificity (Finnemann, 1982, a.o). Thus, there is neither syntactic nor semantic evidence that *hacer* takes an object argument.

On this basis, we assign *hacer* the category (23):

(23) **hacer** $\vdash \mathsf{(s\backslash np)/s} : \lambda P \lambda x.cause' Px$

However, Spanish has examples of cross-conjunct extraction in which *hacer* hosts clitics:

(24) *No solo* ***le*** ***ordenaron****, sino que*
not only cl.dat.3ms ordered.3p but
le ***hicieron*** *barrer la verada.*
cl.dat.3ms made.3p sweep the sidewalk
"They not only ordered him to, but also made him sweep the sidewalk."

This shows another instance of the schema in (1), which is undefined for any of the combinators in (3):

(25)
$$\begin{array}{ccc}
\textbf{le} & \textbf{hicieron} & \textbf{barrer la verada} \\
\hline
\mathsf{(s\backslash np)/((s\backslash np)/np)} & \mathsf{(s\backslash np)/s} & \mathsf{(s|np)} \\
\!\!*\!\longrightarrow & _{>\mathbf{B}} *\!*\!* &
\end{array}$$

3.4 Analyses Based on D

The preceding data motivates adding **D** rules (we return to the distribution of the modalities below):

(26)
$$\begin{array}{lllll}
>\mathbf{D} & \mathsf{x}/_{\diamond}(\mathsf{y}/_{\diamond}\mathsf{z}) & \mathsf{y}/_{\diamond}\mathsf{w} & \Rightarrow & \mathsf{x}/_{\diamond}(\mathsf{w}/_{\diamond}\mathsf{z}) \\
>\mathbf{D}_{\times} & \mathsf{x}/_{\times}(\mathsf{y}/_{\times}\mathsf{z}) & \mathsf{y}\backslash_{\times}\mathsf{w} & \Rightarrow & \mathsf{x}\backslash_{\times}(\mathsf{w}/_{\times}\mathsf{z}) \\
>\mathbf{D}_{\diamond\times} & \mathsf{x}/_{\diamond}(\mathsf{y}\backslash_{\times}\mathsf{z}) & \mathsf{y}/_{\cdot}\mathsf{w} & \Rightarrow & \mathsf{x}/_{\diamond}(\mathsf{w}\backslash_{\times}\mathsf{z}) \\
>\mathbf{D}_{\times\diamond} & \mathsf{x}/_{\times}(\mathsf{y}\backslash_{\diamond}\mathsf{z}) & \mathsf{y}\backslash_{\cdot}\mathsf{w} & \Rightarrow & \mathsf{x}\backslash_{\times}(\mathsf{w}\backslash_{\diamond}\mathsf{z})
\end{array}$$

(27)
$$\begin{array}{lllll}
<\mathbf{D} & \mathsf{y}\backslash_{\diamond}\mathsf{w} & \mathsf{x}\backslash_{\diamond}(\mathsf{y}\backslash_{\diamond}\mathsf{z}) & \Rightarrow & \mathsf{x}\backslash_{\diamond}(\mathsf{w}\backslash_{\diamond}\mathsf{z}) \\
<\mathbf{D}_{\times} & \mathsf{y}/_{\times}\mathsf{w} & \mathsf{x}\backslash_{\times}(\mathsf{y}\backslash_{\times}\mathsf{z}) & \Rightarrow & \mathsf{x}/_{\times}(\mathsf{w}\backslash_{\times}\mathsf{z}) \\
<\mathbf{D}_{\diamond\times} & \mathsf{y}\backslash_{\cdot}\mathsf{w} & \mathsf{x}\backslash_{\diamond}(\mathsf{y}/_{\times}\mathsf{z}) & \Rightarrow & \mathsf{x}\backslash_{\diamond}(\mathsf{w}/_{\times}\mathsf{z}) \\
<\mathbf{D}_{\times\diamond} & \mathsf{y}/_{\cdot}\mathsf{w} & \mathsf{x}\backslash_{\times}(\mathsf{y}/_{\diamond}\mathsf{z}) & \Rightarrow & \mathsf{x}/_{\times}(\mathsf{w}/_{\diamond}\mathsf{z})
\end{array}$$

To illustrate with example (10), one application of $>$**D** allows *you* and *can* to combine when the auxiliary is given the principled type assignment $\mathsf{s/s}$, and another combines *what* with the result.

(28)
$$\begin{array}{ccc}
\textbf{what} & \textbf{you} & \textbf{can} \\
\hline
\mathsf{s}/_{\diamond}(\mathsf{s}/_{\diamond}\mathsf{np}) & \mathsf{s}/_{\diamond}(\mathsf{s}\backslash_{\times}\mathsf{np}) & \mathsf{s}/_{\cdot}\mathsf{s} \\
 & \longrightarrow_{>\mathbf{D}_{\diamond\times}} & \\
 & \mathsf{s}/_{\diamond}(\mathsf{s}\backslash_{\times}\mathsf{np}) & \\
 & \longrightarrow_{>\mathbf{D}} & \\
 & \mathsf{s}/_{\diamond}((\mathsf{s}\backslash_{\times}\mathsf{np})/_{\diamond}\mathsf{np}) &
\end{array}$$

The derivation then proceeds in the usual way.

Likewise, **D** handles the Spanish causative constructions (29) straightforwardly :

(29)
$$\begin{array}{ccc}
\textbf{lo} & \textbf{hice} & \textbf{dormir} \\
\hline
(\mathsf{s}\backslash\mathsf{np})/_{\diamond}((\mathsf{s}\backslash\mathsf{np})/_{\diamond}\mathsf{np}) & (\mathsf{s}\backslash\mathsf{np})/_{\diamond}\mathsf{s} & \mathsf{s/np} \\
 & \longrightarrow_{>\mathbf{D}} & \\
 & (\mathsf{s}\backslash\mathsf{np})/_{\diamond}(\mathsf{s}/_{\diamond}\mathsf{np}) & \\
 & \longrightarrow_{>} & \\
 & \mathsf{s\backslash np} &
\end{array}$$

The **D**-rules thus provide straightforward analyses of such constructions by delivering flexible constituency while maintaining CCG's committment to low categorial ambiguity and semantic transparency.

4 Deriving Eisner Normal Form

Adding new rules can have implications for parsing efficiency. In this section, we show that the **D** rules fit naturally within standard normal form constraints for CCG parsing (Eisner, 1996), by providing both combinatory and logical bases for $\mathbf{D}$. This additionally allows Eisner's normal form constraints to be derived as grammar internal theorems.

4.1 The Spurious Ambiguity Problem

CCG's flexibility is useful for linguistic analyses, but leads to *spurious ambiguity* (Wittenburg, 1987) due to the associativity introduced by the $\mathbf{B}$ and $\mathbf{T}$ rules. This can incur a high computational cost which parsers must deal with. Several techniques have been proposed for the problem (Wittenburg, 1987; Karttunen, 1989; Hepple and Morrill, 1989; Eisner, 1996). The most commonly used are Karttunnen's chart subsumption check (White and Baldridge, 2003; Hockenmaier and Steedman, 2002) and Eisner's normal-form constraints (Bozsahin, 1998; Clark and Curran, 2007).

Eisner's normal form, referred to here as *Eisner NF* and paraphrased in (30), has the advantage of not requiring comparisons of logical forms: it functions purely on the syntactic types being combined.

(30) For a set S of semantically equivalent[2] parse trees for a string ABC, admit the unique parse tree such that at least one of (i) or (ii) holds:

i. C is not the argument of (AB) resulting from application of $>\mathbf{B}^{1+}$.

ii. A is not the argument of (BC) resulting from application of $<\mathbf{B}^{1+}$.

The implication is that outputs of $\mathbf{B}^{1+}$ rules are *inert*, using the terminology of Baldridge (2002). Inert slashes are Baldridge's (2002) encoding in OpenCCG[3] of his CTL interpretation of Steedman's (2000) antecedent-government feature.

Eisner derives (30) from two theorems about the set of semantically equivalent parses that a CCG parser will generate for a given string (see (Eisner, 1996) for proofs and discussion of the theorems):

(31) *Theorem 1*: For every parse tree α, there is a semantically equivalent parse-tree $NF(\alpha)$ in which no node resulting from application of $\mathbf{B}$ or $\mathbf{S}$ functions as the primary functor in a rule application.

(32) *Theorem 2*: If $NF(\alpha)$ and $NF(\alpha')$ are distinct parse trees, then their model-theoretic interpretations are distinct.

Eisner uses a generalized form $\mathbf{B}^n$ ($n{\geq}0$) of composition that subsumes function application:[4]

(33) $>\mathbf{B}^n$: $\mathsf{x/y} \quad \mathsf{y}\$^n \quad \Rightarrow \quad \mathsf{x}\$^n$

(34) $<\mathbf{B}^n$: $\mathsf{y}\$^n \quad \mathsf{x\backslash y} \quad \Rightarrow \quad \mathsf{x}\$^n$

Based on these theorems, Eisner defines NF as follows (for R, S, T as $\mathbf{B}^n$ or $\mathbf{S}$, and Q=$\mathbf{B}^{n\geq 1}$):

(35) Given a parse tree α:

i. If α is a lexical item, then α is in Eisner-NF.

ii. If α is a parse tree $\langle R, \beta, \gamma\rangle$ and $NF(\beta)$, $NF(\gamma)$, then $NF(\alpha)$.

iii. If β is not in Eisner-NF, then
$NF(\beta) = \langle Q, \beta_1, \beta_2\rangle$, and
$NF(\alpha) = \langle S, \beta_1, NF(\langle T, \beta_2, \gamma\rangle)\rangle$.

As a parsing constraint, (30) is a filter on the set of parses produced for a given string. It preserves all the unique semantic forms generated for the string while eliminating all spurious ambiguities: it is both *safe* and *complete*.

Given the utility of Eisner NF for practical CCG parsing, the $\mathbf{D}$ rules we propose should be compatible with (30). This requires that the generalizations underlying (30) apply to $\mathbf{D}$ as well. In the remainder of this section, we show this in two ways.

4.2 Deriving D from B

The first is to derive the binary $\mathbf{B}$ rules from a unary rule based on the unary combinator $\hat{\mathbf{B}}$:[5]

(36) $\mathsf{x/y} : f_{xy} \quad \Rightarrow \quad \mathsf{(x/z)/(y/z)} : \lambda h_{zy}\lambda x_z.f(hx)$

We then derive $\mathbf{D}$ from $\hat{\mathbf{B}}$ and show that clause (iii) of (35) holds of Q schematized over both $\mathbf{B}$ and $\mathbf{D}$.

Applying $\mathbf{D}$ to an argument sequence is equivalent to compound application of binary $\mathbf{B}$:

(37) $(((\mathbf{D}f)g)h)x = (fg)(hx)$

(38) $((((\mathbf{BB})f)g)h)x = ((\mathbf{B}(fg))h)x = (fg)(hx)$

Syntactically, binary $\mathbf{B}$ is equivalent to application of unary $\hat{\mathbf{B}}$ to the primary functor Δ, followed by applying the secondary functor Γ to the output of $\hat{\mathbf{B}}$ by means of function application (Jacobson, 1999):

[2] Two parse trees are semantically equivalent if: (i) their leaf nodes have equivalent interpretations, and (ii) equivalent scope relations hold between their respective leaf-node meanings.

[3] http://openccg.sourceforge.net

[4] We use Steedman's (Steedman, 1996) "\$"-convention for representing argument stacks of length n, for $n \geq 0$.

[5] This is Lambek's (1958) *Division* rule, also known as the "*Geach* rule" (Jacobson, 1999).

(39)
$$\dfrac{\dfrac{\dfrac{\Delta}{\mathsf{x/y}}}{\mathsf{(x/z)/(y/z)}}{>}\hat{\mathbf{B}} \qquad \dfrac{\Gamma}{\mathsf{y/z}}}{\mathsf{x/z}}{>}$$

$\mathbf{B}^n$ ($n \geq 1$) is derived by applying $\hat{\mathbf{B}}$ to the primary functor n times. For example, $\mathbf{B}^{2}$ is derived by 2 applications of $\hat{\mathbf{B}}$ to the primary functor:

(40)
$$\dfrac{\dfrac{\dfrac{\dfrac{\Delta}{\mathsf{x/y}}}{\mathsf{(x/w)/(y/w)}}\hat{\mathbf{B}}}{\mathsf{((x/w)/z)/((y/w)/z)}}\hat{\mathbf{B}} \qquad \dfrac{\Gamma}{\mathsf{(y/w)/z}}}{\mathsf{(x/w)/z}}{>}$$

The rules for **D** correspond to application of $\hat{\mathbf{B}}$ to *both* the primary and secondary functors, followed by function application:

(41)
$$\dfrac{\dfrac{\dfrac{\Delta}{\mathsf{x/(y/z)}}}{\mathsf{(x/(w/z))/((y/z)/(w/z))}}{>}\hat{\mathbf{B}} \qquad \dfrac{\dfrac{\Gamma}{\mathsf{y/w}}}{\mathsf{(y/z)/(w/z)}}{>}\hat{\mathbf{B}}}{\mathsf{x/(w/z)}}{>}$$

As with $\mathbf{B}^n$, $\mathbf{D}^{n\geq 1}$ can be derived by iterative application of $\hat{\mathbf{B}}$ to both primary and secondary functors.

Because **B** can be derived from $\hat{\mathbf{B}}$, clause (iii) of (35) is equivalent to the following:

(42) If β is not in Eisner-NF, then
$NF(\beta) = \langle FA, \langle \hat{\mathbf{B}}, \beta_1 \rangle, \beta_2 \rangle$, such that
$NF(\alpha) = \langle S, \beta_1, NF(\langle T, \beta_2, \gamma \rangle) \rangle$

Interpreted in terms of $\hat{\mathbf{B}}$, both **B** and **D** involve application of $\hat{\mathbf{B}}$ to the primary functor. It follows that Theorem I applies directly to **D** simply by virtue of the equivalence between binary **B** and unary-$\hat{\mathbf{B}}$+FA.

Eisner's NF constraints can then be reinterpreted as a constraint on $\hat{\mathbf{B}}$ requiring its output to be an inert result category. We represent this in terms of the $\hat{\mathbf{B}}$-rules introducing an inert slash, indicated with "!" (adopting the convention from OpenCCG):

(43) $\mathsf{x/y} : f_{xy} \quad \Rightarrow \quad \mathsf{(x/^{!}z)/(y/^{!}z)} : \lambda h_{zy} \lambda x_z f h x$

Hence, both binary **B** and **D** return inert functors:

(44)
$$\dfrac{\dfrac{\dfrac{\Delta}{\mathsf{x/y}}}{\mathsf{(x/^{!}z)/(y/^{!}z)}}{>}\hat{\mathbf{B}} \qquad \dfrac{\Gamma}{\mathsf{y/z}}}{\mathsf{x/^{!}z}}{>}$$

(45)
$$\dfrac{\dfrac{\dfrac{\Delta}{\mathsf{x/(y/z)}}}{\mathsf{(x/^{!}(w/z))/((y/z)/^{!}(w/z))}}{>}\hat{\mathbf{B}} \qquad \dfrac{\dfrac{\Gamma}{\mathsf{y/w}}}{\mathsf{(y/^{!}z)/(w/^{!}z)}}{>}\hat{\mathbf{B}}}{\mathsf{x/^{!}(w/z)}}{>}$$

The binary substitution (**S**) combinator can be similarly incorporated into the system. Unary substitution $\hat{\mathbf{S}}$ is like $\hat{\mathbf{B}}$ except that it introduces a slash on only the argument-side of the input functor. We stipulate that $\hat{\mathbf{S}}$ returns a category with inert slashes:

(46) ($\hat{\mathbf{S}}$) $\quad \mathsf{(x/y)/z} \;\Rightarrow\; \mathsf{(x/^{!}z)/(y/^{!}z)}$

T is by definition unary. It follows that all the binary rules in CCG (including the **D**-rules) can be reduced to (iterated) instantiations of the unary combinators $\hat{\mathbf{B}}$, $\hat{\mathbf{S}}$, or **T** plus function application.

This provides a basis for CCG in which all combinatory rules are derived from unary $\hat{\mathbf{B}}$ $\hat{\mathbf{S}}$, and **T**.

4.3 A Logical Basis for Eisner Normal Form

The previous section shows that deriving CCG rules from unary combinators allows us to derive the **D**-rules while preserving Eisner NF. In this section, we present an alternate formulation of Eisner NF with Baldridge's (2002) CTL basis for CCG. This formulation allows us to derive the **D**-rules as before, and does so in a way that seamlessly integrates with Baldridge's system of modalized functors.

In CTL, $\mathbf{B}_{\diamond}$ and $\mathbf{B}_{\times}$ are proofs derived via structural rules that allow associativity and permutation of symbols within a sequent, in combination with the slash introduction and elimination rules of the base logic. To control application of these rules, Baldridge keys them to binary modal operators $\diamond$ (for associativity) and $\times$ (for permutation). Given these, $>\mathbf{B}$ is proven in (47):

(47)
$$\dfrac{\dfrac{\dfrac{\Delta \vdash \mathsf{x}/_{\diamond}\mathsf{y} \qquad \dfrac{\Gamma \vdash \mathsf{y}/_{\diamond}\mathsf{z} \qquad [a \vdash \mathsf{z}]}{(\Gamma \circ_{\diamond} a_i) \vdash \mathsf{y}}[/_{\diamond}E]}{(\Delta \circ_{\diamond} (\Gamma \circ_{\diamond} a_i)) \vdash \mathsf{x}}[/_{\diamond}E]}{((\Delta \circ_{\diamond} \Gamma) \circ_{\diamond} a_i) \vdash \mathsf{x}}[RA]}{(\Delta \circ_{\diamond} \Gamma) \vdash \mathsf{x}/_{\diamond}\mathsf{z}}[/_{\diamond}I]$$

In a CCG ruleset compiled from such logics, a category must have an appropriately decorated slash in order to be the input to a rule. This means that rules apply universally, without language-specific restrictions. Instead, restrictions can only be declared via modalities marked on lexical categories.

Unary $\hat{\mathbf{B}}$ and the $\mathbf{D}$ rules in 4.2 can be derived using the same logic. For example, $>\hat{\mathbf{B}}$ can be derived as in (48):

(48)
$$\dfrac{\dfrac{\dfrac{\dfrac{\Delta \vdash \mathsf{x}/_{\diamond}\mathsf{y} \quad \dfrac{[f \vdash \mathsf{y}/_{\diamond}\mathsf{z}]^{1} \quad [a \vdash \mathsf{z}]^{2}}{(f_{1} \circ_{\diamond} a_{2}) \vdash \mathsf{y}}\,[/E]}{(\Delta \circ_{\diamond} (f_{1} \circ_{\diamond} a_{2})) \vdash \mathsf{x}}\,[/_{\diamond}E]}{((\Delta \circ_{\diamond} f_{1}) \circ_{\diamond} a_{2}) \vdash \mathsf{x}}\,[RA]}{(\Delta \circ_{\diamond} f_{1}) \vdash \mathsf{x}/_{\diamond}\mathsf{z}}\,[/_{\diamond}I]}{\Delta \vdash (\mathsf{x}/_{\diamond}\mathsf{z})/_{\diamond}(\mathsf{y}/_{\diamond}\mathsf{z})}\,[/_{\diamond}I]$$

The $\mathbf{D}$ rules are also theorems of this system. For example, the proof for $>\mathbf{D}$ applies (48) as a lemma to each of the primary and secondary functors:

(49)
$$\dfrac{\dfrac{\Delta \vdash \mathsf{x}/_{\diamond}(\mathsf{y}/_{\diamond}\mathsf{z})}{\Delta \vdash (\mathsf{x}/_{\diamond}(\mathsf{w}/_{\diamond}\mathsf{z}))/_{\diamond}((\mathsf{y}/_{\diamond}\mathsf{z})/_{\diamond}(\mathsf{w}/_{\diamond}\mathsf{z}))}\,{>}\hat{\mathbf{B}} \quad \dfrac{\Gamma \vdash \mathsf{y}/_{\diamond}\mathsf{w}}{\Gamma \vdash (\mathsf{y}/_{\diamond}\mathsf{z})/_{\diamond}(\mathsf{w}/_{\diamond}\mathsf{z})}\,{>}\hat{\mathbf{B}}}{(\Delta \circ_{\diamond} \Gamma) \vdash \mathsf{x}/_{\diamond}(\mathsf{w}/_{\diamond}\mathsf{z})}\,[/E]$$

$>\mathbf{D}_{\diamond\times}$ involves an associative version of $\hat{\mathbf{B}}$ applied to the primary functor (50), and a permutative version to the secondary functor (51).

(50)
$$\dfrac{\dfrac{\dfrac{\dfrac{\Delta \vdash \mathsf{x}/_{\diamond}(\mathsf{y}\backslash_{\times}\mathsf{z}) \quad \dfrac{[f \vdash (\mathsf{y}\backslash_{\times}\mathsf{z})/_{.}(\mathsf{w}\backslash_{\times}\mathsf{z})]^{1} \quad [g \vdash \mathsf{w}\backslash_{\times}\mathsf{z}]^{2}}{(f_{1} \circ_{.} g_{2}) \vdash \mathsf{y}\backslash_{\times}\mathsf{z}}\,[/_{.}E]}{(\Delta \circ_{\diamond} (f_{1} \circ_{.} g_{2})) \vdash \mathsf{x}}\,[/_{\diamond}E]}{((\Delta \circ_{\diamond} f_{1}) \circ_{.} g_{2}) \vdash \mathsf{x}}\,[RA]}{(\Delta \circ_{\diamond} f_{1}) \vdash \mathsf{x}/_{.}(\mathsf{w}\backslash_{\times}\mathsf{z})}\,[/_{.}I]}{\Delta \vdash (\mathsf{x}/_{.}(\mathsf{w}\backslash_{\times}\mathsf{z}))/_{\diamond}((\mathsf{y}\backslash_{\times}\mathsf{z})/_{.}(\mathsf{w}\backslash_{\times}\mathsf{z}))}\,[/_{\diamond}I]$$

(51)
$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash \mathsf{y}/_{.}\mathsf{w} \quad \dfrac{[a \vdash \mathsf{z}]^{1} \quad [f \vdash \mathsf{w}\backslash_{\times}\mathsf{z}]^{2}}{(a_{1} \circ_{\times} f_{2}) \vdash \mathsf{w}}\,[\backslash_{\times}E]}{(\Gamma \circ_{.} (a_{1} \circ_{\times} f_{2})) \vdash \mathsf{y}}\,[/_{.}E]}{(a_{1} \circ_{\times} (\Gamma \circ_{.} f_{2})) \vdash \mathsf{y}}\,[LP]}{(\Gamma \circ_{.} f_{2}) \vdash \mathsf{y}\backslash_{\times}\mathsf{z}}\,[\backslash_{\times}I]}{\Gamma \vdash (\mathsf{y}\backslash_{\times}\mathsf{z})/_{.}(\mathsf{w}\backslash_{\times}\mathsf{z})}\,[/_{.}I]$$

Rules for $\mathbf{D}$ with appropriate modalities can therefore be incorporated seamlessly into CCG.

In the preceding subsection, we encoded Eisner NF with inert slashes. In Baldridge's CTL basis for CCG, inert slashes are represented as functors seeking non-lexical arguments, represented as categories marked with an *antecedent-governed* feature, reflecting the intuition that non-lexical arguments have to be "bound" by a superordinate functor.

This is based on an interpretation of antecedent-government as a unary modality $\Diamond_{ant}$ that allows structures marked by it to permute to the left or right periphery of a structure:[6]

(52)
$$\dfrac{((\Delta_{a} \circ_{\times} \Diamond_{ant}\Delta_{b}) \circ_{\times} \Delta_{c}) \vdash \mathsf{x}}{((\Delta_{a} \circ_{\times} \Delta_{c}) \circ_{\times} \Diamond_{ant}\Delta_{b}) \vdash \mathsf{x}} \qquad [\text{ARP}]$$

$$\dfrac{(\Delta_{a} \circ_{\times} (\Diamond_{ant}\Delta_{b} \circ_{\times} \Delta_{c})) \vdash \mathsf{x}}{(\Diamond_{ant}\Delta_{b} \circ_{\times} (\Delta_{a} \circ_{\times} \Delta_{c})) \vdash \mathsf{x}} \qquad [\text{ALP}]$$

Unlike permutation rules without $\Diamond_{ant}$, these permutation rules can only be used in a proof when preceeded by a hypothetical category marked with the $\Box^{\downarrow}_{ant}$ modality. The elimination rule for $\Box^{\downarrow}$-modalities introduces a corresponding $\Diamond$-marked object in the resulting structure, feeding the rule:

(53)
$$\dfrac{\dfrac{[a \vdash \Diamond_{ant}\Box^{\downarrow}_{ant}\mathsf{z}]^{2} \quad \dfrac{\dfrac{\Delta \vdash \mathsf{x}/_{\times}\mathsf{y} \quad \dfrac{\dfrac{[a \vdash \Box^{\downarrow}_{ant}\mathsf{z}]^{1}}{\Diamond_{ant}a_{1} \vdash \mathsf{z}}\,[\Box^{\downarrow}E] \quad \Gamma \vdash \mathsf{y}\backslash_{\times}\mathsf{z}}{(\Diamond_{ant}a_{1} \circ_{\times} \Gamma) \vdash \mathsf{y}}\,[\backslash_{\times}E]}{(\Delta \circ_{\times} (\Diamond_{ant}a_{1} \circ_{\times} \Gamma)) \vdash \mathsf{x}}\,[/_{\times}E]}{(\Diamond_{ant}a_{1} \circ_{\times} (\Delta \circ_{\times} \Gamma)) \vdash \mathsf{x}}\,[ALP]}{(a \circ_{\times} (\Delta \circ_{\times} \Gamma)) \vdash \mathsf{x}}\,[\Diamond E]}{(\Delta \circ_{\times} \Gamma) \vdash \mathsf{x}\backslash_{\times}\Diamond_{ant}\Box^{\downarrow}_{ant}\mathsf{z}}\,[\backslash_{\times}I]^{2}$$

Re-introduction of the $[a \vdash \Diamond_{ant}\Box^{\downarrow}_{ant}\mathsf{z}]^{k}$ hypothesis results in a functor the argument of which is marked with $\Diamond_{ant}\Box^{\downarrow}_{ant}$. Because lexical categories are not marked as such, the functor cannot take a lexical argument, and so is effectively an inert functor.

In Baldridge's (2002) system, only proofs involving the ARP and ALP rules produce inert categories. In Eisner NF, all instances of $\mathbf{B}$-rules result in inert categories. This can be reproduced in Baldridge's system simply by keying *all* structural rules to the ant-modality, the result being that all proofs involving structural rules result in inert functors.

As desired, the $\mathbf{D}$-rules result in inert categories as well. For example, $>\mathbf{D}$ is derived as follows ($\Box^{\downarrow}_{ant}$ and $\Diamond_{ant}$ are abbreviated as $\Box^{\downarrow}$ and $\Diamond$):

[6] Note that the diamond operator used here is a syntactic operator, rather than a semantic operator as used in (16) above. The unary modalities used in CTL describe accessibility relationships between subtypes and supertypes of particular categories: in effect, they define feature hierarchies. See Moortgat (1997) and Oehrle (To Appear) for further explanation.

(54)

$$\dfrac{[c \vdash \Diamond\Box^{\downarrow}z]^{3} \quad \dfrac{\dfrac{\Gamma \vdash y/_{\diamond}w \quad \dfrac{\dfrac{[a \vdash \Box^{\downarrow}(w/_{\diamond}z)]^{1}}{\Diamond a \vdash w/_{\diamond}z}[\Box^{\downarrow}E] \quad \dfrac{[b \vdash \Box^{\downarrow}z]^{2}}{\Diamond b \vdash z}[\Box^{\downarrow}E]}{(\Diamond a \ \circ_{\diamond} \Diamond b) \vdash w}[/_{\diamond}E]}{(\Gamma \ \circ_{\diamond} (\Diamond a \ \circ_{\diamond} \Diamond b)) \vdash y}[/_{\diamond}E]}{((\Gamma \ \circ_{\diamond} \Diamond a) \ \circ_{\diamond} \Diamond b) \vdash y}[RA]}{\dfrac{((\Gamma \ \circ_{\diamond} \Diamond a) \ \circ_{\diamond} c) \vdash y}{(\Gamma \ \circ_{\diamond} \Diamond a) \vdash y/_{\diamond}\Diamond\Box^{\downarrow}z}[/_{\diamond}I]^{3}}[\Diamond E]^{2}$$

(55)

$$\dfrac{[d \vdash \Diamond\Box^{\downarrow}(w/_{\diamond}z)]^{4} \quad \dfrac{\dfrac{\Delta \vdash x/_{\diamond}(y/_{\diamond}\Diamond\Box^{\downarrow}z) \quad \begin{array}{c}(54)\\ \vdots \\ (\Gamma \ \circ_{\diamond} \Diamond a) \vdash y/_{\diamond}\Diamond\Box^{\downarrow}z\end{array}}{(\Delta \ \circ_{\diamond} (\Gamma \ \circ_{\diamond} \Diamond a)) \vdash x}[/_{\diamond}E]}{((\Delta \ \circ_{\diamond} \Gamma) \ \circ_{\diamond} \Diamond a) \vdash x}[RA]}{\dfrac{((\Delta \ \circ_{\diamond} \Gamma) \ \circ_{\diamond} d) \vdash x}{(\Delta \ \circ_{\diamond} \Gamma) \vdash x/_{\diamond}\Diamond\Box^{\downarrow}(w/_{\diamond}z)}[/_{\diamond}I]^{4}}[\Diamond E]^{1}$$

(54)-(55) can be used as a lemma corresponding to the CCG rule in (57):

(56) $$\dfrac{\Delta \vdash x/_{\diamond}(y/_{\diamond}\Diamond\Box^{\downarrow}z) \quad \Gamma \vdash y/_{\diamond}w}{(\Delta \ \circ_{\diamond} \Gamma) \vdash x/_{\diamond}\Diamond\Box^{\downarrow}(w/_{\diamond}z)}[\mathbf{D}]$$

(57) $x/_{\diamond}(y/_{\diamond}^{!}z) \quad y/_{\diamond}w \quad \Rightarrow \quad x/_{\diamond}^{!}(w/_{\diamond}z)$

This means that all CCG rules compiled from the logic—which requires $\Diamond_{ant}$ to licence the structural rules necessary to prove the rules—return inert functors. Eisner NF thus falls out of the logic because all instances of **B**, **D**, and **S** produce inert categories. This in turns allows us to view Eisner NF as part of a theory of grammatical competence, in addition to being a useful technique for constraining parsing.

5 Conclusion

Including the **D**-combinator rules in the CCG rule set lets us capture several linguistic generalizations that lack satisfactory analyses in standard CCG. Furthermore, CCG augmented with **D** is compatible with Eisner NF (Eisner, 1996), a standard technique for controlling derivational ambiguity in CCG-parsers, and also with the modalized version of CCG (Baldridge and Kruijff, 2003). A consequence is that both the **D** rules and the NF constraints can be derived from a grammar-internal perspective. This extends CCG's linguistic applicability without sacrificing efficiency.

Wittenburg (1987) originally proposed using rules based on **D** as a way to reduce spurious ambiguity, which he achieved by eliminating **B** rules entirely and replacing them with variations on **D**. Wittenburg notes that doing so produces as many instances of **D** as there are rules in the standard rule set. Our proposal retains **B** and **S**, but, thanks to Eisner NF, eliminates spurious ambiguity, a result that Wittenburg was not able to realize at the time.

Our approach can be incorporated into Eisner NF straightforwardly However, Eisner NF disprefers incremental analyses by forcing right-corner analyses of long-distance dependencies, such as in (58):

(58) (What (does (Grommet (think (Tottie (said (Victor (knows (Wallace ate)))))))))?

For applications that call for increased incrementality (e.g., aligning visual and spoken input incrementally (Kruijff et al., 2007)), CCG rules that do not produce inert categories can be derived a CTL basis that does not require $\Diamond_{ant}$ for associativity and permutation. The **D**-rules derived from this kind of CTL specification would allow for left-corner analyses of such dependencies with the competence grammar. An extracted element can "wrap around" the words intervening between it and its extraction site. For example, **D** would allow the following bracketing for the same example (while producing the same logical form):

(59) (((((((((What does) Grommet) think) Tottie) said) Victor) knows) Wallace) ate)?

Finally, the unary combinator basis for CCG provides an interesting additional specification for generating CCG rules. Like the CTL basis, the unary combinator basis can produce a much wider range of possible rules, such as **D** rules, that may be relevant for linguistic applications. Whichever basis is used, inclusion of the **D**-rules increases empirical coverage, while at the same time preserving CCG's computational attractiveness.

Acknowledgments

Thanks Mark Steedman for extensive comments and suggestions, and particularly for noting the relationship between the **D**-rules and unary $\hat{\mathbf{B}}$. Thanks also to Emmon Bach, Cem Bozsahin, Jason Eisner, Geert-Jan Kruijff and the ACL reviewers.

References

Farrell Ackerman and John Moore. 1999. Syntagmatic and Paradigmatic Dimensions of Causee Encodings. *Linguistics and Philosophy*, 24:1–44.

Avery D. Andrews and Christopher D. Manning. 1999. *Complex Predicates and Information Spreading in LFG*. CSLI Publications, Palo Alto, California.

Jason Baldridge and Geert-Jan Kruijff. 2003. Multi-Modal Combinatory Categorial Grammar. In *Proceedings of EACL 10*, pages 211–218.

Jason Baldridge, Sudipta Chatterjee, Alexis Palmer, and Ben Wing. 2007. DotCCG and VisCCG: Wiki and Programming Paradigms for Improved Grammar Engineering with OpenCCG. In *Proceedings of GEAF 2007*.

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

John Beavers. 2004. Type-inheritance Combinatory Categorial Grammar. In *Proceedings of COLING-04*, Geneva, Switzerland.

Robert Borsley and Kersti Börjars, editors. To Appear. *Non-Transformational Syntax: A Guide to Current Models*. Blackwell.

Cem Bozsahin. 1998. Deriving the Predicate-Argument Structure for a Free Word Order Language. In *Proceedings of COLING-ACL '98*.

Stephen Clark and James Curran. 2007. Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. *Computational Linguistics*, 33(4).

Haskell B. Curry and Robert Feys. 1958. *Combinatory Logic*, volume 1. North Holland, Amsterdam.

Jason Eisner. 1996. Efficient Normal-Form Parsing for Combinatory Categorial Grammars. In *Proceedings of the ACL 34*.

Michael D Finnemann. 1982. *Aspects of the Spanish Causative Construction*. Ph.D. thesis, University of Minnesota.

L. T. F. Gamut. 1991. *Logic, Language, and Meaning*, volume II. Chicago University Press.

Jeroen Groenendijk and Martin Stokhof. 1997. Questions. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, chapter 19, pages 1055–1124. Elsevier Science, Amsterdam.

Mark Hepple and Glyn Morrill. 1989. Parsing and Derivational Equivalence. In *Proceedings of EACL 4*.

Julia Hockenmaier and Mark Steedman. 2002. Generative Models for Statistical Parsing with Combinatory Categorial Grammar. In *Proceedings. of ACL 40*, pages 335–342, Philadelpha, PA.

Pauline Jacobson. 1990. Raising as Function Composition. *Linguistics and Philosophy*, 13:423–475.

Pauline Jacobson. 1999. Towards a Variable-Free Semantics. *Linguistics and Philosophy*, 22:117–184.

Lauri Karttunen. 1989. Radical Lexicalism. In Mark Baltin and Anthony Kroch, editors, *Alternative Conceptions of Phrase Structure*. University of Chicago Press, Chicago.

Angelika Kratzer. 1991. Modality. In Arnim von Stechow and Dieter Wunderlich, editors, *Semantics: An International Handbook of Contemporary Semantic Research*, pages 639–650. Walter de Gruyter, Berlin.

Geert-Jan M. Kruijff, Pierre Lison, Trevor Benjamin, Henrik Jacobsson, and Nick Hawes. 2007. Incremental, Multi-Level Processing for Comprehending Situated Dialogue in Human-Robot Interaction. In *Language and Robots: Proceedings from the Symposium (LangRo'2007)*, Aveiro, Portugal.

Joachim Lambek. 1958. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–169.

Marta Luján. 1980. Clitic Promotion and Mood in Spanish Verbal Complements. *Linguistics*, 18:381–484.

Michael Moortgat. 1997. Categorial Type Logics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 93–177. North Holland, Amsterdam.

Richard T Oehrle. To Appear. Multi-Modal Type Logical Grammar. In Boersley and Börjars (Borsley and Börjars, To Appear).

Martin Pickering and Guy Barry. 1993. Dependency Categorial Grammar and Coordination. *Linguistics*, 31:855–902.

David Reitter, Julia Hockenmaier, and Frank Keller. 2006. Priming Effects in Combinatory Categorial Grammar. In *Proceedings of EMNLP-2006*.

Mark Steedman and Jason Baldridge. To Appear. Combinatory Categorial Grammar. In Borsley and Börjars (Borsley and Börjars, To Appear).

Mark Steedman. 1996. *Surface Structure and Interpretation*. MIT Press.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press.

Michael White and Jason Baldridge. 2003. Adapting Chart Realization to CCG. In *Proceedings of ENLG*.

Michael White. 2006. Efficient Realization of Coordinate Structures in Combinatory Categorial Grammar. *Research on Language and Computation*, 4(1):39–75.

Kent Wittenburg. 1987. Predictive Combinators: A Method for Efficient Processing of Combinatory Categorial Grammars. In *Proceedings of ACL 25*.

Luke Zettlemoyer and Michael Collins. 2007. Online Learning of Relaxed CCG Grammars for Parsing to Logical Form. In *Proceedings of EMNLP-CoNLL 2007*.