

Generalizing Dimensionality in Combinatory Categorical Grammar

Geert-Jan M. Kruijff
Computational Linguistics
Saarland University
Saarbrücken, Germany
gj@coli.uni-sb.de

Jason Baldridge
ICCS, Division of Informatics
University of Edinburgh
Edinburgh, Scotland
jbaldr@dinf.ed.ac.uk

Abstract

We extend Combinatory Categorical Grammar (CCG) with a generalized notion of multi-dimensional sign, inspired by the types of representations found in constraint-based frameworks like HPSG or LFG. The generalized sign allows multiple levels to share information, but only in a resource-bounded way through a very restricted indexation mechanism. This improves representational perspicuity without increasing parsing complexity, in contrast to full-blown unification used in HPSG and LFG. Well-formedness of a linguistic expressions remains entirely determined by the CCG derivation. We show how the multidimensionality and perspicuity of the generalized signs lead to a simplification of previous CCG accounts of how word order and prosody can realize information structure.

1 Introduction

The information conveyed by linguistic utterances is diverse, detailed, and complex. To properly analyze what is communicated by an utterance, this information must be encoded and interpreted at many levels. The literature contains various proposals for dealing with many of these levels in the description of natural language grammar.

Since information flows between different levels of analysis, it is common for linguistic formalisms to bundle them together and provide some means for communication between them. Categorical grammars, for example, normally employ a Saussurian sign that relates a surface string with its syntactic category and the meaning it expresses. Syntactic analysis is entirely driven by the categories, and when information from other levels is used to affect the derivational possibilities, it is typically loaded as extra information on the categories.

Head-driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1993) and Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1982) also use complex signs. However, these signs are monolithic structures which permit information to be

freely shared across all dimensions: any given dimension can place restrictions on another. For example, variables resolved during the construction of the logical form can block a syntactic analysis. This provides a clean, unified formal system for dealing with the different levels, but it also can adversely affect the complexity of parsing grammars written in these frameworks (Maxwell and Kaplan, 1993).

We thus find two competing perspectives on communication between levels in a sign. In this paper, we propose a generalization of linguistic signs for Combinatory Categorical Grammar (CCG) (Steedman, 2000b). This generalization enables different levels of linguistic information to be represented but limits their interaction in a resource-bounded manner, following White (2004). This provides a clean separation of the levels and allows them to be designed and utilized in a more modular fashion. Most importantly, it allows us to retain the parsing complexity of CCG while gaining the representational advantages of the HPSG and LFG paradigms.

To illustrate the approach, we use it to model various aspects of the realization of information structure, an inherent aspect of the (linguistic) meaning of an utterance. Speakers use information structure to present some parts of that meaning as depending on the preceding discourse context and others as affecting the context by adding new content. Languages may realize information structure using different, often interacting means, such as word order, prosody, (marked) syntactic constructions, or morphological marking (Vallduví and Engdahl, 1996; Kruijff, 2002). The literature presents various proposals for how information structure can be captured in categorical grammar (Steedman, 2000a; Hoffman, 1995; Kruijff, 2001). Here, we model the essential aspects of these accounts in a more perspicuous manner by using our generalized signs.

The main outcomes of the proposal are threefold: (1) CCG gains a more flexible and general kind of sign; (2) these signs contain multiple levels that interact in a modular fashion and are built via CCG derivations without increasing parsing com-

plexity; and (3) we use these signs to simplify previous CCG’s accounts of the effects of word order and prosody on information structure.

2 Combinatory Categorial Grammar

In this section, we give an overview of syntactic combination and semantic construction in CCG. We use CCG’s multi-modal extension (Baldrige and Kruijff, 2003), which enriches the inventory of slash types. This formalization renders constraints on rules unnecessary and supports a universal set of rules for all grammars.

2.1 Categories and combination

Nearly all syntactic behavior in CCG is encoded in categories. They may be atoms, like np , or functions which specify the direction in which they seek their arguments, like $(s \backslash np)/np$. The latter is the category for English transitive verbs; it first seeks its object to its right and then its subject to its left.

Categories combine through a small set of universal combinatory rules. The simplest are application rules which allow a function category to consume its argument either on its right ($>$) or on its left ($<$):

$$\begin{aligned} (>) \quad X/_*Y \quad Y &\Rightarrow X \\ (<) \quad Y \quad X \backslash_*Y &\Rightarrow X \end{aligned}$$

Four further rules allow functions to *compose* with other functions:

$$\begin{aligned} (>\mathbf{B}) \quad X/_\diamond Y \quad Y/_\diamond Z &\Rightarrow X/_\diamond Z \\ (<\mathbf{B}) \quad Y \backslash_\diamond Z \quad X \backslash_\diamond Y &\Rightarrow X \backslash_\diamond Z \\ (>\mathbf{B}_\times) \quad X/_\times Y \quad Y \backslash_\times Z &\Rightarrow X \backslash_\times Z \\ (<\mathbf{B}_\times) \quad Y \backslash_\times Z \quad X \backslash_\times Y &\Rightarrow X \backslash_\times Z \end{aligned}$$

The modalities $*$, \diamond and \times on the slashes enforce different kinds of combinatorial potential on categories. For a category to serve as input to a rule, it must contain a slash which is compatible with that specified by the rule. The modalities work as follows. $*$ is the most restricted modality, allowing combination only by the application rules ($>$ and $<$). \diamond allows combination with the application rules and the order-preserving composition rules ($>\mathbf{B}$ and $<\mathbf{B}$). \times allows limited permutation via the crossed composition rules ($>\mathbf{B}_\times$ and $<\mathbf{B}_\times$) as well as the application rules. Additionally, a permissive modality \cdot allows combination by all rules in the system. However, we suppress the \cdot modality on slashes to avoid clutter. An undecorated slash may thus combine by all rules.

There are two further rules of *type-raising* that turn an argument category into a function over functions that seek that argument:

$$\begin{aligned} (>\mathbf{T}) \quad X &\Rightarrow Y/_i(Y \backslash_i X) \\ (<\mathbf{T}) \quad X &\Rightarrow Y \backslash_i(Y /_i X) \end{aligned}$$

The variable modality i on the output categories constrains both slashes to have the same modality.

These rules support the following incremental derivation for *Marcel proved completeness*:

$$(1) \quad \frac{\frac{\frac{Marcel}{np} \quad \frac{proved}{(s \backslash np)/np} \quad \frac{completeness}{np}}{s/(s \backslash np)} \quad >\mathbf{T}}{s/np} \quad >\mathbf{B}}{s} \quad >$$

This derivation does not display the effect of using modalities in CCG; see Baldrige (2002) and Baldrige and Kruijff (2003) for detailed linguistic justification for this modalized formulation of CCG.

2.2 Hybrid Logic Dependency Semantics

Many different kinds of semantic representations and ways of building them with CCG exist. We use Hybrid Logic Dependency Semantics (HLDS) (Kruijff, 2001), a framework that utilizes hybrid logic (Blackburn, 2000) to realize a dependency-based perspective on meaning.

Hybrid logic provides a language for representing relational structures that overcomes standard modal logic’s inability to directly reference states in a model. This is achieved via *nominals*, a kind of basic formula which explicitly names states. Like propositions, nominals are first-class citizens of the object language, so formulas can be formed using propositions, nominals, standard boolean operators, and the *satisfaction operator* “@”. A formula $@_i(p \wedge \langle F \rangle(j \wedge q))$ indicates that the formulas p and $\langle F \rangle(j \wedge q)$ hold at the state named by i and that the state j is reachable via the modal relation F .

In HLDS, hybrid logic is used as a language for describing semantic interpretations as follows. Each semantic head is associated with a nominal that identifies its *discourse referent* and heads are connected to their dependents via dependency relations, which are modeled as modal relations. As an example, the sentence *Marcel proved completeness* receives the representation in (2).

$$(2) \quad @_e(\mathbf{prove} \wedge \langle \mathbf{TENSE} \rangle \mathbf{past} \wedge \langle \mathbf{ACT} \rangle (m \wedge \mathbf{Marcel}) \wedge \langle \mathbf{PAT} \rangle (c \wedge \mathbf{comp}))$$

In this example, e is a nominal that labels the predications and relations for the head **prove**, and m and c label those for **Marcel** and **completeness**, respectively. The relations ACT and PAT represent the dependency roles **Actor** and **Patient**, respectively.

By using the @ operator, hierarchical terms such as (2) can be flattened to an equivalent conjunction of fixed-size *elementary predications* (EPs):

$$(3) \quad @_e \mathbf{prove} \wedge @_e \langle \mathbf{TENSE} \rangle \mathbf{past} \wedge @_e \langle \mathbf{ACT} \rangle m \\ \wedge @_e \langle \mathbf{PAT} \rangle c \wedge @_m \mathbf{Marcel} \wedge @_c \mathbf{comp.}$$

2.3 Semantic Construction

Baldrige and Kruijff (2002) show how HLDS representations can be built via CCG derivations. White (2004) improves HLDS construction by operating on flattened representations such as (3) and using a simple semantic index feature in the syntax. We adopt this latter approach, described below.

EPs are paired with syntactic categories in the lexicon as shown in (4)–(6) below. Each atomic category has an index feature, shown as a subscript, which makes a nominal available for capturing syntactically induced dependencies.

$$(4) \quad \mathit{prove} \vdash (s_e \setminus np_x) / np_y : \\ @_e \mathbf{prove} \wedge @_e \langle \mathbf{TENSE} \rangle \mathbf{past} \\ \wedge @_e \langle \mathbf{ACT} \rangle x \wedge @_e \langle \mathbf{PAT} \rangle y \\ (5) \quad \mathit{Marcel} \vdash np_m : @_m \mathbf{Marcel} \\ (6) \quad \mathit{completeness} \vdash np_c : @_c \mathbf{completeness}$$

Applications of the combinatory rules co-index the appropriate nominals via unification on the categories. EPs are then conjoined to form the resulting interpretation. For example, in derivation (1), (5) type-raises and composes with (4) to yield (7). The index x is syntactically unified with m , and this resolution is reflected in the new conjoined logical form. (7) can then apply to (6) to yield (8), which has the same conjunction of predications as (3).

$$(7) \quad \mathit{Marcel proved} \vdash s_e / np_y : \\ @_e \mathbf{prove} \wedge @_e \langle \mathbf{TENSE} \rangle \mathbf{past} \\ \wedge @_e \langle \mathbf{ACT} \rangle m \wedge @_e \langle \mathbf{PAT} \rangle y \wedge @_m \mathbf{Marcel} \\ (8) \quad \mathit{Marcel proved completeness} \vdash s_e : \\ @_e \mathbf{prove} \wedge @_e \langle \mathbf{TENSE} \rangle \mathbf{past} \wedge @_e \langle \mathbf{ACT} \rangle m \\ \wedge @_e \langle \mathbf{PAT} \rangle c \wedge @_m \mathbf{Marcel} \wedge @_c \mathbf{completeness}$$

Since the EPs are always conjoined by the combinatory rules, semantic construction is guaranteed to be *monotonic*. No semantic information can be dropped during the course of a derivation. This provides a clean way of establishing semantic dependencies as informed by the syntactic derivation. In the next section, we extend this paradigm for use with any number of representational levels.

3 Generalized dimensionality

To support a more modular and perspicuous encoding of multiple levels of analysis, we generalize the notion of sign commonly used in CCG. The approach is inspired on the one hand by earlier work by Steedman (2000a) and Hoffman (1995), and on

the other by the signs found in constraint-based approaches to grammar. The principle idea is to extend White’s (2004) approach to semantic construction (see §2.3). There, categories and the meaning they help express are connected through co-indexation. Here, we allow for information in any (finite) number of levels to be related in this way.

A sign is an n -tuple of terms that represent information at n distinct dimensions. Each dimension represents a level of linguistic information such as prosody, meaning, or syntactic category. As a representation, we assume that we have for each dimension a language that defines *well-formed representations*, and a set of *operations* which can create new representations from a set of given representations.¹

For example, we have by definition a dimension for syntactic categories. The language for this dimension is defined by the rules for category construction: given a set of atomic categories \mathcal{A} , C is a category iff (i) $C \in \mathcal{A}$ or (ii) C is of the form $A \setminus_m B$ or $A /_m B$ with A, B categories and $m \in \{*, \diamond, \times, \cdot\}$. The set of combinatory rules defines the possible operations on categories.

This syntactic category dimension drives the grammatical analysis, thus guiding the composition of signs. When two categories are combined via a rule, the appropriate indices are unified. It is through this unification of indices that information can be passed between signs. At a given dimension, the co-indexed information coming from the two signs we combine must be unifiable.

With these signs, dimensions interact in a more limited way than in HPSG or LFG. Constraints (resolved through unification) may only be applied if they are invoked through co-indexation on categories. This provides a bound on the number of indices and the number of unifications to be made. As such, full recursion and complex unification as in attribute-value matrices with re-entrancy is avoided.

The approach incorporates various ideas from constraint-based approaches, but remains based on a derivational perspective on grammatical analysis and derivational control, unlike e.g. Categorical Unification Grammar. Furthermore, the ability for dimensions to interact through shared indices brings several advantages: (1) “parallel derivations” (Hoffman, 1995) are unnecessary; (2) non-isomorphic, functional structures across different dimensions can be employed; and (3) there is no longer a need to load all the necessary information into syntactic categories (as with Kruijff (2001)).

¹In the context of this paper we assume operations are multiplicative. Also, note that dimensions may differ in what languages and operations they use.

4 Examples

In this section, we illustrate our approach on several examples involving information structure. We use signs that include the following dimensions.

Phonemic representation: word sequences, composition of sequences is through concatenation

Prosody: sequences of tunes from the inventory of (Pierrehumbert and Hirschberg, 1990), composition through concatenation

Syntactic category: well-formed categories, combinatory rules (see §2)

Information structure: hybrid logic formulas of the form $@_d[in]r$, with r a discourse referent that has informativity in (theme θ , or rheme ρ) relative to the current point in the discourse d (Kruijff, 2003).

Predicate-argument structure: hybrid logic formulas of the form as discussed in §2.3.

Example (9) illustrates a sign with these dimensions. The word-form *Marcel* bears an H^* accent, and acts as a type-raised category that seeks a verb missing its subject. The H^* accent indicates that the discourse referent m introduces new information at the current point in the discourse d : i.e. the meaning $@_m$ marcel should end up as part of the rheme (ρ) of the utterance, $@_d[\rho]m$.

$$(9) \quad \frac{\text{Marcel}}{H^*} \\ \frac{s_n / (s_n \setminus np_m)}{@_d[\rho]m} \\ @_m \text{marcel}$$

If a sign does not specify any information at a particular dimension, this is indicated by \top (or an empty line if no confusion can arise).

4.1 Topicalization

We start with a simple example of topicalization in English. In topicalized constructions, a thematic object is fronted before the subject. Given the question *Did Marcel prove soundness and completeness?*, (10) is a possible response using topicalization:

- (10) Completeness, Marcel proved, and soundness, he conjectured.

We can capture the syntactic and information structure effects of such sentences by assigning the following kind of sign to (topicalized) noun phrases:

$$(11) \quad \frac{\text{completeness}}{\top} \\ \frac{s_i / (s_i / np_c)}{@_d[\theta]c} \\ @_c \text{completeness}$$

This category enables the derivation in Figure 1. The type-raised subject composes with the verb, and the result is consumed by the topicalizing category. The information structure specification stated in the sign in (11) is passed through to the final sign.

The topicalization of the object in (10) only indicates the informativity of the discourse referent realized by the object. It does not yield any indications about the informativity of other constituents; hence the informativity for the predicate and the Actor is left unspecified. In English, the informativity of these discourse referents can be indicated directly with the use of prosody, to which we now turn.

4.2 Prosody & information structure

Steedman (2000a) presents a detailed, CCG-based account of how prosody is used in English as a means to realize information structure. In the model, pitch accents and boundary tones have an effect on both the syntactic category of the expression they mark, and the meaning of that expression.

Steedman distinguishes pitch accents as markers of either the *theme* (θ) or of the *rheme* (ρ): $L+H^*$ and L^*+H are θ -markers; H^* , L^* , H^*+L and $H+L^*$ are ρ -markers. Since pitch accents mark individual words, not (necessarily) larger phrases, Steedman uses the θ/ρ -marking to spread informativity over the domain and the range of function categories. Identical markings on different parts of a function category not only act as features, but also as occurrences of a singular variable. The value of the marking on the domain can thus get passed down (“projected”) to markings on categories in the range.

Constituents bearing no tune have an η -marking, which can be unified with either η , θ or ρ . Phrases with such markings are “incomplete” until they combine with a boundary tone. Boundary tones have the effect of mapping phrasal tones into intonational phrase boundaries. To make these boundaries explicit and enforce such “complete” prosodic phrases to only combine with other complete prosodic phrases, Steedman introduces two further types of marking – ι and ϕ – on categories. The ϕ markings only unify with other ϕ or ι markings on categories, not with η , θ or ρ . These markings are only introduced to provide derivational control and are not reflected in the underlying meaning (which only reflects η , θ or ρ).

Figure 2 recasts the above as an abstract specification of which different types of prosodic constituents can, or cannot, be combined.² Steedman’s

²There is one exception we should note: two intermediate phrases *can* combine if a second one has a downstepped accent. We deal with this exception at the end of the section.

$$\begin{array}{c}
\begin{array}{ccc}
\textit{completeness} & \textit{Marcel} & \textit{proved} \\
\hline
s_i/(s_i/np_c) & s_j/(s_j/np_m) & (s_p/np_x)/np_y \\
\textcircled{d}[\theta]c & & \\
\textcircled{c}\textit{completeness} & \textcircled{m}\textit{Marcel} & \textcircled{p}\textit{prove} \wedge \textcircled{p}\langle\textit{ACT}\rangle x \wedge \textcircled{p}\langle\textit{PAT}\rangle y \\
\hline
& & s_p/np_y \\
& & \textcircled{p}\textit{prove} \wedge \textcircled{p}\langle\textit{ACT}\rangle m \wedge \textcircled{p}\langle\textit{PAT}\rangle y \wedge \textcircled{m}\textit{Marcel} \\
\hline
& & s_p \\
& & \textcircled{d}[\theta]c \\
\textcircled{p}\textit{prove} \wedge \textcircled{p}\langle\textit{ACT}\rangle m \wedge \textcircled{p}\langle\textit{PAT}\rangle c \wedge \textcircled{m}\textit{Marcel} \wedge \textcircled{c}\textit{completeness}
\end{array}
\end{array}$$

Figure 1: Derivation for topicalization.

system can be implemented using just one feature *pros* which takes the values *ip* for intermediate phrases, *cp* for complete phrases, and *up* for unmarked phrases. We write $s_{pros=ip}$, or simply s_{ip} if no confusion can arise.

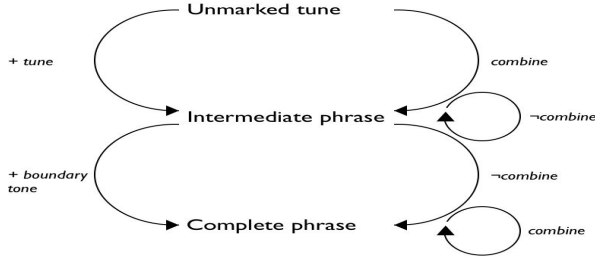


Figure 2: Abstract specification of derivational control in prosody

First consider the top half of Figure 2. If a constituent is marked with either a θ - or ρ -tune, the atomic result category of the (possibly complex) category is marked with *ip*. Prosodically unmarked constituents are marked as *up*. The lexical entries in (12) illustrates this idea.³

$$(12) \begin{array}{ccc}
\textit{MARCEL} & \textit{proved} & \textit{COMPLETENESS} \\
\textit{H}^* & & \textit{L+H}^* \\
\hline
s_{ip}/(s_{up}\backslash np) & (s_{up}\backslash np)/np & s_{ip}\$(s_{up}\$/np)
\end{array}$$

This can proceed in two ways. Either the marked *MARCEL* and the unmarked *proved* combine to produce an intermediate phrase (13), or *proved* and the marked *COMPLETENESS* combine (14).

$$(13) \begin{array}{ccc}
\textit{MARCEL} & \textit{proved} & \textit{COMPLETENESS} \\
\textit{H}^* & & \textit{L+H}^* \\
\hline
s_{ip}/(s_{up}\backslash np) & (s_{up}\backslash np)/np & s_{ip}\$(s_{up}\$/np) \\
\hline
& & s_{ip}/np
\end{array}$$

³The $\$$'s in the category for *COMPLETENESS* are standard CCG schematizations: $s\$\$$ indicates all functions into s , such as $s\backslash np$ and $(s\backslash np)/np$. See Steedman (2000b) for details.

$$(14) \begin{array}{ccc}
\textit{MARCEL} & \textit{proved} & \textit{COMPLETENESS} \\
\textit{H}^* & & \textit{L+H}^* \\
\hline
s_{ip}/(s_{up}\backslash np) & (s_{up}\backslash np)/np & s_{ip}\$(s_{up}\$/np) \\
\hline
& & s_{ip}\backslash np
\end{array}$$

For the remainder of this paper, we will suppress *up* marking and write s_{up} simply as s .

Examples (13) and (14) show that prosodically marked and unmarked phrases can combine. However, both of these partial derivations produce categories that cannot be combined further. For example, in (14), $s_{ip}/(s\backslash np)$ cannot combine with $s_{ip}\backslash np$ to yield a larger intermediate phrase. This properly captures the top half of Figure 2.

To obtain a complete analysis for (12), boundary tones are needed to complete the intermediate phrases tones. For example, consider (15) (based on example (70) in Steedman (2000a)):

$$(15) \begin{array}{ccc}
\textit{MARCEL} & \textit{proved} & \textit{COMPLETENESS} \\
\textit{H}^* & \textit{L} & \textit{L+H}^* \quad \textit{LH}\%
\end{array}$$

To capture the bottom-half of Figure 2, the boundary tones L and $LH\%$ need categories which create complete phrases out of those for *MARCEL* and *proved* *COMPLETENESS*, and thereafter allow them to combine. Figure 3 shows the appropriate categories and complete analysis.

We noted earlier that downstepped phrasal tones form an exception to the rule that intermediate phrases cannot combine. To enable this, we not only should mark the result category with *ip* (tune), but also any leftward argument(s) should have *ip* (downstep). Thus, the effect of (lexically) combining a downstep tune with an unmarked category is specified by the following template: add marking $\times_{ip}\$\y_{ip} to an unmarked category of the form $\times\$\y . The derivation in Figure 5 illustrates this idea on example (64) from (Steedman, 2000a).

To relate prosody to information structure, we extend the strategy used for constructing logical forms described in §2.3, in which a simple index feature

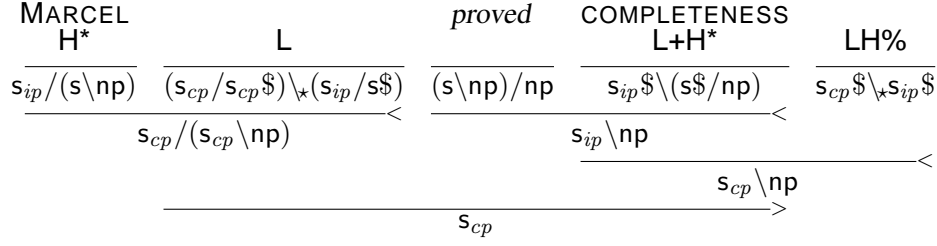


Figure 3: Derivation including tunes and boundary tones; (70) from (Steedman, 2000a)

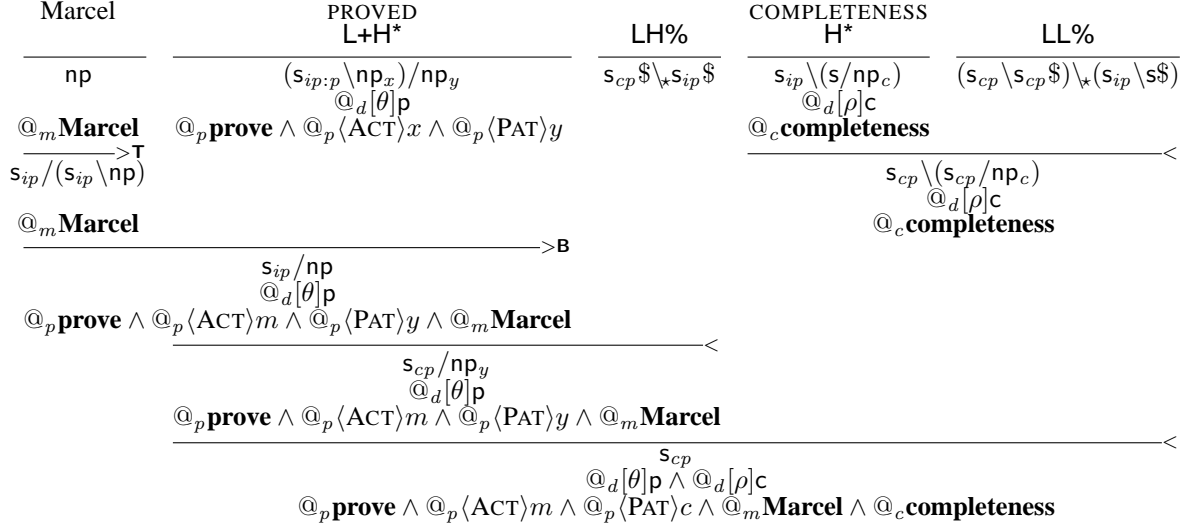


Figure 4: Information structure for derivation for (67)-(68) from (Steedman, 2000a)

on atomic categories makes a nominal (discourse referent) available. We represent information structure as a formula $@_d[i]r$ at a dimension separate from the syntactic category. The nominal r stands for the discourse referent, which has informativity i with respect to the current point in the discourse d (Kruijff, 2003). Following Steedman, we distinguish two levels of informativity, namely θ (theme) and ρ (rheme).

We start with a minimal assignment of informativity: a theme-tune on a constituent sets the informativity of the discourse referent r realized by the constituent to θ and a rheme-tune sets it to ρ . This is a minimal assignment in the sense that we do not *project* informativity; instead, we only set informativity for those discourse referents whose realization shows explicit clues as to their information status. The derivation in Figure 4 illustrates this idea and shows the construction of both logical form and information structure.

Indices can also impose constraints on the informativity of arguments. For example, in the downstep example (Figure 5), the discourse referents corresponding to ANNA and SAYS are both part of the

theme. We specify this with the constituent that has received the downstepped tune. The referent of the subject of SAYS (indexed x) must be in the theme along with the referent s for SAYS. This is satisfied in the derivation: a unifies with x , and we can unify the statements about a 's informativity coming from ANNA ($@_d[\theta]a$) and SAYS ($@_d[\theta]x$ with x replaced by a in the $>\mathbf{B}$ step).

5 Conclusions

In this paper, we generalize the traditional Sausurian sign in CCG with an n -dimensional linguistic sign. The dimensions in the generalized linguistic sign can be related through indexation. Indexation places constraints on signs by requiring that co-indexed material is unifiable, on a per-dimension basis. Consequently, we do not need to overload the syntactic category with information from different dimensions.

The resulting sign structure resembles the signs found in constraint-based grammar formalisms. There is, however, an important difference. Information at various dimensions can be related through co-indexation, but dimensions cannot be directly

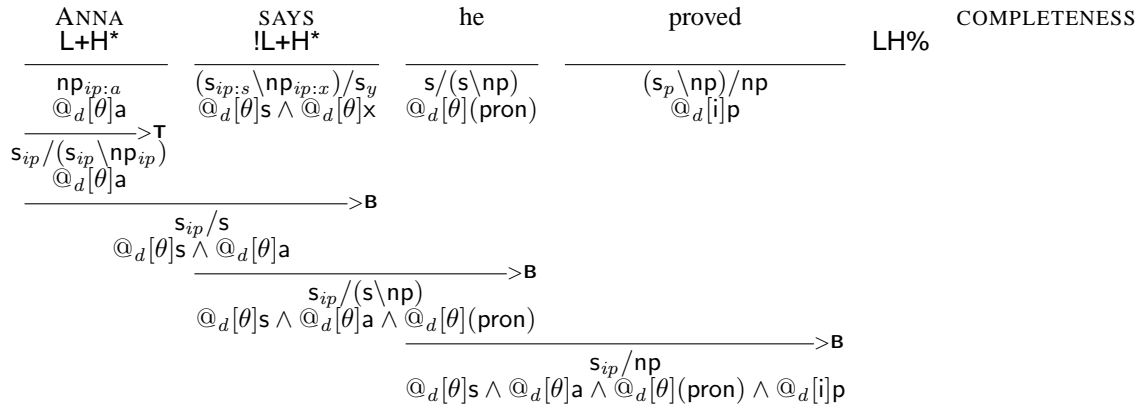


Figure 5: Information structure for derivation for (64) from (Steedman, 2000a)

referenced. As analysis remains driven only by inference over categories, only those constraints triggered by indexation on the categories are imposed. We do not allow for re-entrancy.

It is possible to conceive of a scenario in which the various levels *can* contribute toward determining the well-formedness of an expression. For example, we may wish to evaluate the current information structure against a discourse model, and reject the analysis if we find it is unsatisfiable. If such a move is made, then the complexity will be bounded by the complexity of the dimension for which it is most difficult to determine satisfiability.

Acknowledgments

Thanks to Ralph Debusmann, Alexander Koller, Mark Steedman, and Mike White for discussion. Geert-Jan Kruijff's work is supported by the DFG SFB 378 *Resource-Sensitive Cognitive Processes*, Project NEGRA EM 6.

References

Jason Baldridge and Geert-Jan Kruijff. 2002. Coupling CCG and Hybrid Logic Dependency Semantics. In *Proc. of 40th Annual Meeting of the ACL*, pages 319–326, Philadelphia, Pennsylvania.

Jason Baldridge and Geert-Jan Kruijff. 2003. Multi-Modal Combinatory Categorical Grammar. In *Proc. of 10th Annual Meeting of the EACL*, Budapest.

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

Patrick Blackburn. 2000. Representation, reasoning, and relational structures: a hybrid logic manifesto. *Journal of the Interest Group in Pure Logic*, 8(3):339–365.

Beryl Hoffman. 1995. Integrating "free" word order syntax and information structure. In *Proc. of 7th Annual Meeting of the EACL*, Dublin.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-functional grammar: A formal system for grammat-

ical representation. In *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press, Cambridge Massachusetts.

Geert-Jan M. Kruijff. 2001. *A Categorical-Modal Logical Architecture of Informativity: Dependency Grammar Logic & Information Structure*. Ph.D. thesis, Charles University, Prague, Czech Republic.

Geert-Jan M. Kruijff. 2002. Formulating a category of informativity. In Hilde Hasselgard, Stig Johansson, Bergljot Behrens, and Cathrine Fabricius-Hansen, editors, *Information Structure in a Cross-Linguistic Perspective*, pages 129–146. Rodopi, Amsterdam.

Geert-Jan M. Kruijff. 2003. Binding across boundaries. In Geert-Jan M. Kruijff and Richard T. Oehrle, editors, *Resource Sensitivity, Binding, and Anaphora*. Kluwer Academic Publishers, Dordrecht.

John T. III Maxwell and Ronald M. Kaplan. 1993. The interface between phrasal and functional constraints. *Computational Linguistics*, 19(4):571–590.

Janet Pierrehumbert and Julia Hirschberg. 1990. The meaning of intonational contours in the interpretation of discourse. In J. Morgan P. Cohen and M. Pollack, editors, *Intentions in Communication*. The MIT Press, Cambridge Massachusetts.

Carl Pollard and Ivan A. Sag. 1993. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago IL.

Mark Steedman. 2000a. Information structure and the syntax-phonology interface. *Linguistic Inquiry*, 31(4):649–689.

Mark Steedman. 2000b. *The Syntactic Process*. The MIT Press, Cambridge, MA.

Enric Vallduví and Elisabet Engdahl. 1996. The linguistic realization of information packaging. *Linguistics*, 34:459–519.

Michael White. 2004. Efficient realization of coordinate structures in Combinatory Categorical Grammar. *Research on Language and Computation*. To appear.