

## Coupling CCG and Hybrid Logic Dependency Semantics

**Jason Baldridge**

ICCS

Division of Informatics

2 Buccleuch Place

University of Edinburgh

Edinburgh, UK, EH8 9LW

jmb@cogsci.ed.ac.uk

**Geert-Jan M. Kruijff**

Universität des Saarlandes

Computational Linguistics

Lehrstuhl Uszkoreit

Building 17, Postfach 15 11 50

66041 Saarbrücken, Germany

gj@CoLi.Uni-SB.DE

### Abstract

Categorial grammar has traditionally used the  $\lambda$ -calculus to represent meaning. We present an alternative, dependency-based perspective on linguistic meaning and situate it in the computational setting. This perspective is formalized in terms of hybrid logic and has a rich yet perspicuous propositional ontology that enables a wide variety of semantic phenomena to be represented in a single meaning formalism. Finally, we show how we can couple this formalization to Combinatory Categorial Grammar to produce interpretations compositionally.

### 1 Introduction

The  $\lambda$ -calculus has enjoyed many years as the standard semantic encoding for categorial grammars and other grammatical frameworks, but recent work has highlighted its inadequacies for both linguistic and computational concerns of representing natural language semantics (Copestake et al., 1999; Kruijff, 2001). The latter couples a resource-sensitive categorial proof theory (Moortgat, 1997) to hybrid logic (Blackburn, 2000) to formalize a dependency-based perspective on meaning, which we call here Hybrid Logic Dependency Semantics (HLDS). In this paper, we situate HLDS in the computational context by explicating its properties as a framework for computational semantics and linking it to Combinatory Categorial Grammar (CCG).

The structure of the paper is as follows. In §2, we briefly introduce CCG and how it links syntax

and semantics, and then discuss semantic representations that use indexes to identify subparts of logical forms. §3 introduces HLDS and evaluates it with respect to the criteria of other computational semantics frameworks. §4 shows how we can build HLDS terms using CCG with unification and §5 shows how intonation and information structure can be incorporated into the approach.

### 2 Indexed semantic representations

Traditionally, categorial grammar has captured meaning using a (simply typed)  $\lambda$ -calculus, building semantic structure in parallel to the categorial inference (Morrill, 1994; Moortgat, 1997; Steedman, 2000b). For example, a (simplified) CCG lexical entry for a verb such as *wrote* is given in (1).

$$(1) \quad \textit{wrote} \vdash (s \setminus n) / n : \lambda x. \lambda y. \mathbf{write}(y, x)$$

Rules of combination are defined to operate on both categories and  $\lambda$ -terms simultaneously. For example, the rules allow the following derivation for *Ed wrote books*.

$$(2) \quad \frac{\frac{\frac{\textit{Ed}}{n:\mathbf{Ed}} \quad \textit{wrote}}{(s \setminus n) / n : \lambda x. \lambda y. \mathbf{write}(y, x)} \quad \textit{books}}{n:\mathbf{books}}}{s \setminus n : \lambda y. \mathbf{write}(y, \mathbf{books})} >}{s : \mathbf{write}(\mathbf{Ed}, \mathbf{books})} <$$

Derivations like (2) give rise to the usual sort of predicate-argument structure whereby the *order* in which the arguments appear (and are bound by the  $\lambda$ 's) is essentially constitutive of their meaning. Thus, the first argument could be taken to correspond to the writer, whereas the second argument corresponds to what is being written.

One deficiency of  $\lambda$ -calculus meaning representations is that they usually have to be type-raised to the worst case to fully model quantification, and this can reverberate and increase the complexity of syntactic categories since a verb like *wrote* will need to be able to take arguments with the types of generalized quantifiers. The approach we advocate in this paper does not suffer from this problem.

For CCG, the use of the  $\lambda$ -terms is simply a convenient device to bind arguments when presenting derivations (Steedman, 2000b). In implementations, a more common strategy is to compute semantic representations via unification, a tactic explicitly employed in Unification Categorical Grammar (UCG) (Zeevat, 1988). Using a unification paradigm in which atomic categories are bundles of syntactic and semantic information, we can use an entry such as (3) for *wrote* in place of (1). In the unification setting, (3) permits a derivation analogous to (2).

$$(3) \quad \textit{wrote} \vdash (s : \mathbf{write}(y, x) \setminus n : y) / n : x$$

For creating predicate-argument structures of this kind, strategies using either  $\lambda$ -terms or unification to bind arguments are essentially notational variants. However, UCG goes beyond simple predicate-argument structures to instead use a semantic representation language called Indexed Language (InL). The idea of using indexes stems from Davidson (event variables), and are a commonly used mechanism in unification-based frameworks and theories for discourse representation. InL attaches one to every formula representing its discourse referent. This results in a representation such as (4) for the sentence *Ed came to the party*.

$$(4) \quad [e][\mathbf{party}(x), \mathbf{past}(e), \mathbf{to}(e, x), \mathbf{come}(e, \mathbf{Ed})]$$

InL thus flattens logical forms to some extent, using the indexes to spread a given entity or event through multiple predications. The use of indexes is crucial for UCG's account of modifiers, and as we will see later, we exploit such referents to achieve similar ends when coupling HLDS and CCG.

Minimal Recursion Semantics (MRS) (Copestake et al., 1999; Copestake et al., 2001) is a framework for computational semantics that is designed to simplify the work of algorithms which produce or use semantic representations. MRS provides the

means to represent interpretations with a flat, underspecified semantics using terms of the predicate calculus and generalized quantifiers. Flattening is achieved by using an indexation scheme involving *labels* that tag particular groups of elementary predications (EPs) and *handles* (here,  $h_1, h_2, \dots$ ) that reference those EPs. Underspecification is achieved by using unresolved handles as the arguments for scope-bearing elements and declaring constraints (with the  $\equiv_q$  operator) on how those handles can be resolved. Different scopes can be reconstructed by equating unresolved handles with the labels of the other EPs obeying the  $\equiv_q$  constraints. For example, (5) would be given as the representation for *every dog chases some white cat*.

$$(5) \quad \langle h_0, \{h_1 : \mathbf{every}(x, h_2, h_3), h_4 : \mathbf{dog}(x), \\ h_{11} : \mathbf{cat}(y), h_8 : \mathbf{some}(y, h_9, h_{10}), \\ h_{11} : \mathbf{white}(y), h_7 : \mathbf{chase}(x, y)\}, \\ \{h_0 \equiv_q h_7, h_2 \equiv_q h_4, h_9 \equiv_q h_{11}\} \rangle$$

Copestake et al. argue that these flat representations facilitate a number of computational tasks, including machine translation and generation, without sacrificing linguistic expressivity. Also, flatness permits semantic equivalences to be checked more easily than in structures with deeper embedding, and underspecification simplifies the work of the parser since it does not have to compute every possible reading for scope-bearing elements.

### 3 Hybrid Logic Dependency Semantics

Kruijff (2001) proposes an alternative way to representing linguistically realized meaning: namely, as terms of *hybrid modal logic* (Blackburn, 2000) explicitly encoding the dependency relations between heads and dependents, spatio-temporal structure, contextual reference, and information structure. We call this unified perspective combining many levels of meaning Hybrid Logic Dependency Semantics (HLDS). We begin by discussing how hybrid logic extends modal logic, then look at the representation of linguistic meaning via hybrid logic terms.

#### 3.1 Hybrid Logic

Though modal logic provides a powerful tool for encoding relational structures and their properties,

it contains a surprising inherent asymmetry: states (“worlds”) are at the heart of the model theory for modal logic, but there are no means to directly *reference* specific states using the object language. This inability to state where exactly a proposition holds makes modal logic an inadequate representation framework for practical applications like knowledge representation (Areces, 2000) or temporal reasoning (Blackburn, 1994). Because of this, computational work in knowledge representation has usually involved re-engineering first-order logic to suit the task, e.g., the use of metapredicates such as **Hold** of Kowalski and Allen. Unfortunately, such logics are often undecidable.

Hybrid logic extends standard modal logic while retaining decidability and favorable complexity (Areces, 2000) (cf. (Areces et al., 1999) for a complexity roadmap). The strategy is to add *nominals*, a new sort of basic formula with which we can explicitly name states in the object language. Next to propositions, nominals are first-class citizens of the object language: formulas can be formed using both sorts, standard boolean operators, and the *satisfaction operator* “@”. A formula  $@_i p$  states that the formula  $p$  holds at the state named by  $i$ .<sup>1</sup> (There are more powerful quantifiers ranging over nominals, such as  $\downarrow$ , but we do not consider them here.)

With nominals we obtain the possibility to explicitly refer to the state at which a proposition holds. As Blackburn (1994) argues, this is essential for capturing our intuitions about temporal reference. A standard modal temporal logic with the modalities F and P (future and past, respectively) cannot correctly represent an utterance such as *Ed finished the book* because it is unable to refer to the *specific* time at which the event occurred. The addition of nominals makes this possible, as shown in (6), where the nominal  $i$  represents the Reichenbachian event time.

$$(6) \quad \langle P \rangle (i \wedge \mathbf{Ed\text{-}finish\text{-}book})$$

Furthermore, many temporal properties can be defined in terms of *pure formulas* which use nominals and contain no propositional variables. For example, the following term defines the fact that the relations for F and P are mutually converse:

<sup>1</sup>A few notes on our conventions:  $p, q, r$  are variables over any hybrid logic formula;  $i, j, k$  are variables over nominals;  $d_i$  and  $h_i$  denote nominals (for dependent and head, respectively).

$$(7) \quad @_i[F]\langle P \rangle i \wedge @_i[P]\langle F \rangle i$$

It is also possible to encode a variety of other representations in terms of hybrid logics. For example, nominals correspond to tags in attribute-value matrices (AVMs), so the hybrid logic formula in (8) corresponds to the AVM in (9).

$$(8) \quad \langle \text{SUBJ} \rangle (i \wedge \langle \text{AGR} \rangle \textit{singular} \wedge \langle \text{PRED} \rangle \textit{dog}) \\ \wedge \langle \text{COMP} \rangle \langle \text{SUBJ} \rangle i$$

$$(9) \quad \left[ \begin{array}{l} \text{SUBJ} \quad \square \left[ \begin{array}{l} \text{AGR singular} \\ \text{PRED dog} \end{array} \right] \\ \text{COMP} \quad \left[ \text{SUBJ} \quad \square \right] \end{array} \right]$$

A crucial aspect of hybrid logic is that nominals are at the heart of a *sorting strategy*. Different sorts of nominals can be introduced to build up a rich sortal ontology without losing the perspicuity of a propositional setting. Additionally, we can *reason* about sorts because nominals are part and parcel of the object language. We can extend the language of hybrid logic with  $\{\text{Sort:Nominal}\}$  to facilitate the explicit statement of what sort a nominal is in the language and carry this modification into one of the existing tableaux methods for hybrid logic to reason effectively with this information. This makes it possible to capture the rich ontologies of lexical databases like WordNet in a clear and concise fashion which would be onerous to represent in first-order logic.

### 3.2 Encoding linguistic meaning

Hybrid logic enables us to logically capture two essential aspects of meaning in a clean and compact way, namely ontological richness and the possibility to refer. Logically, we can represent an expression’s linguistically realized meaning as a conjunction of modalized terms, anchored by the nominal that identifies the head’s proposition:

$$(10) \quad @_h(\mathbf{proposition} \wedge \langle \delta_i \rangle (d_i \wedge \mathbf{dep}_i))$$

Dependency relations are modeled as modal relations  $\langle \delta_i \rangle$ , and with each dependent we associate a nominal  $d_i$ , representing its *discourse referent*. Technically, (10) states that each nominal  $d_i$  names the state where a dependent expressed as a proposition  $\mathbf{dep}_i$  should be evaluated and is a  $\delta_i$  successor of  $h$ , the nominal identifying the head. As an example, the sentence *Ed wrote a long book in London* receives the representation in (11).

$$(11) \quad @_{h_1}(\mathbf{write} \wedge \langle \mathbf{ACT} \rangle (d_0 \wedge \mathbf{Ed}) \\ \wedge \langle \mathbf{PAT} \rangle (d_5 \wedge \mathbf{book} \wedge \langle \mathbf{GR} \rangle (d_7 \wedge \mathbf{long})) \\ \wedge \langle \mathbf{LOC} \rangle (d_9 \wedge \mathbf{London}))$$

The modal relations ACT, PAT, LOC, and GR stand for the dependency relations **Actor**, **Patient**, **Locative**, and **General Relationship**, respectively. See Kruijff (2001) for the model-theoretic interpretation of expressions such as (11).

Contextual reference can be modeled as a statement that from the current state (anaphor) there should be an accessible antecedent state at which particular conditions hold. Thus, assuming an accessibility relation  $XS$ , we can model the meaning of the pronoun *he* as in (12).

$$(12) \quad @_i \langle XS \rangle (j \wedge \mathbf{male})$$

During discourse interpretation, this statement is *evaluated* against the discourse model. The pronoun is resolvable only if a state where **male** holds is  $XS$ -accessible in the discourse model. Different accessibility relations can be modeled, e.g. to distinguish a local context (for resolving reflexive anaphors like *himself*) from a global context (Kruijff, 2001).

Finally, the rich temporal ontology underlying models of tense and aspect such as Moens and Steedman (1988) can be captured using the sorting strategy. Earlier work like Blackburn and Lascarides (1992) already explored such ideas. HLDS employs hybrid logic to integrate Moens and Steedman's notion of the *event nucleus* directly into meaning representations. The event nucleus is a tripartite structure reflecting the underlying semantics of a type of event. The event is related to a *preparation* (an activity bringing the event about) and a *consequent* (a state ensuing to the event), which we encode as the modal relations PREP and CONS, respectively. Different kinds of states and events are modeled as different *sorts of nominals*, shown in (13) using the notation introduced above.

$$(13) \quad @_{\{\text{Activity}:e1\}} \langle \mathbf{PREP} \rangle \{\text{Achievement}:e2\} \\ \wedge @_{\{\text{Achievement}:e2\}} \langle \mathbf{CONS} \rangle \{\text{State}:e3\}$$

To tie (13) in with a representation like (11), we equate the nominal of the head with one of the nominals in the event nucleus ( $\bar{E}$ ) and state its temporal relation (e.g.  $\langle P \rangle$ ). Given the event nucleus in (13), the representation in (11) becomes (14), where the event is thus located at a *specific* time in the past.

$$(14) \quad @_{h_1} (\bar{E}_{(13)} \wedge \mathbf{write} \wedge \langle \mathbf{ACT} \rangle (d_0 \wedge \mathbf{Ed}) \\ \wedge \langle \mathbf{PAT} \rangle (d_5 \wedge \mathbf{book} \wedge \langle \mathbf{GR} \rangle (d_7 \wedge \mathbf{long})) \\ \wedge \langle \mathbf{LOC} \rangle (d_9 \wedge \mathbf{London})) \\ \wedge @_{h_1} \{\text{Achievement}:e2\} \wedge \langle P \rangle \{\text{Achievement}:e2\}$$

Hybrid logic's flexibility makes it amenable to representing a wide variety of semantic phenomena in a propositional setting, and it can furthermore be used to formulate a discourse theory (Kruijff and Kruijff-Korbayová, 2001).

### 3.3 Comparison to MRS

Here we consider the properties of HLDS with respect to the four main criteria laid out by Copestake et al. (1999) which a computational semantics framework must meet: expressive adequacy, grammatical compatibility, computational tractability, and underspecifiability.

Expressive adequacy refers to a framework's ability to correctly express linguistic meaning. HLDS was designed not only with this in mind, but as its central tenet. In addition to providing the means to represent the usual predicate-valency relations, it explicitly marks the named dependency relations between predicates and their arguments and modifiers. These different dependency relations are not just labels: they all have unique semantic imports which project new relations in the context of different heads. HLDS also tackles the representation of tense and aspect, contextual reference, and information structure, as well as their interaction with discourse.

The criterion of grammatical compatibility requires that a framework be linkable to other kinds of grammatical information. Kruijff (2001) shows that HLDS can be coupled to a rich grammatical framework, and in §4 we demonstrate that it can be tied to CCG, a much lower power formalism than that assumed by Kruijff. It should furthermore be straightforward to use our approach to hook HLDS up to other unification-based frameworks.

The definition of computational tractability states that it must be possible to check semantic equivalence of different formulas straightforwardly. Like MRS, HLDS provides the means to view linguistic meaning in a flattened format and thereby ease the checking of equivalence. For example, (15) describes the same relational structure as (11).

$$(15) \quad @_{h_1}(\mathbf{write} \wedge \langle \text{ACT} \rangle d_0 \wedge \langle \text{PAT} \rangle d_5 \wedge \langle \text{LOC} \rangle d_9) \\ \wedge @_{d_0} \mathbf{Ed} \wedge @_{d_5} \mathbf{book} \wedge @_{d_9} \mathbf{London} \\ \wedge @_{d_7} \mathbf{long} \wedge @_{d_5} \langle \text{GR} \rangle d_7$$

This example clarifies how the use of nominals is related to the indexes of UCG’s InL and the labels of MRS. However, there is an important difference: nominals are full citizens of the object language with semantic import and are not simply a device for spreading meaning across several elementary predications. They simultaneously represent tags on subparts of a logical form and discourse referents on which relations are predicated. Because it is possible to view an HLDS term as a flat conjunction of the heads and dependents inside it, the benefits described by Copestake et al. with respect to MRS’s flatness thus hold for HLDS as well.

Computational tractability also requires that it is straightforward to express relationships between representations. This can be done in the object language of HLDS as hybrid logic implicational statements which can be used with proof methods to discover deeper relationships. Kruijff’s model connecting linguistic meaning to a discourse context is one example of this.

Underspecifiability means that semantic representations should provide means to leave some semantic distinctions unresolved whilst allowing partial terms to be flexibly and monotonically resolved. (5) shows how MRS leaves quantifier scope underspecified, and such formulas can be transparently encoded in HLDS. Consider (16), where the relations RESTR and BODY represent the *restriction* and *body* arguments of the generalized quantifiers, respectively.

$$(16) \quad @_{h_7}(\mathbf{chase} \wedge \langle \text{ACT} \rangle h_4 \wedge \langle \text{PAT} \rangle h_{11}) \\ \wedge @_{h_1}(\mathbf{every} \wedge \langle \text{RESTR} \rangle i \wedge \langle \text{BODY} \rangle j) \\ \wedge @_{h_8}(\mathbf{some} \wedge \langle \text{RESTR} \rangle k \wedge \langle \text{BODY} \rangle l) \\ \wedge @_{h_4} \mathbf{dog} \wedge @_{h_{11}} \mathbf{cat} \wedge @_{h_{11}} \langle \text{GR} \rangle (h_{12} \wedge \mathbf{white}) \\ \wedge @_i \langle \text{QEQ} \rangle h_4 \wedge @_k \langle \text{QEQ} \rangle h_{11}$$

MRS-style underspecification is thus replicated by declaring new nominals and modeling  $\Rightarrow_q$  as a modal relation between nominals. When constructing the fully-scoped structures generated by an underspecified one, the  $\Rightarrow_q$  constraints must be obeyed according to the **qeq** condition of Copestake et al. Because HLDS is couched directly in terms of hybrid logic, we can concisely declare the **qeq** condition as the following implication:

$$(17) \quad @_i \langle \text{QEQ} \rangle j \rightarrow @_i j \vee (@_i \langle \text{BODY} \rangle k \wedge @_k \langle \text{QEQ} \rangle j)$$

Alternatively, it would in principle be possible to adopt a truly modal solution to the representation of quantifiers. Following Alechina (1995), (generalized) quantification can be modeled as *modal operators*. The complexity of generalized quantification is then pushed into the model theory instead of forcing the representation to carry the burden.

## 4 CCG Coupled to HLDS

In Dependency Grammar Logic (DGL), Kruijff (2001) couples HLDS to a resource-sensitive categorial proof theory (CTL) (Moortgat, 1997). Though DGL demonstrates a procedure for building HLDS terms from linguistic expressions, there are several problems we can overcome by switching to CCG. First, parsing with CCG grammars for substantial fragments is generally more efficient than with CTL grammars with similar coverage. Also, a wide-coverage statistical parser which produces syntactic dependency structures for English is available for CCG (Clark et al., 2002). Second, syntactic features (modeled by unary modalities) in CTL have no intuitive semantic reflection, whereas CCG can relate syntactic and semantic features perspicuously using unification. Finally, CCG has a detailed syntactic account of the realization of information structure in English.

To link syntax and semantics in derivations, every logical form in DGL expresses a nominal identifying its head in the format  $@_i p$ . This handles dependents in a linguistically motivated way through a *linking theory*: given the form of a dependent, its (possible) role is established, after which its meaning states that it seeks a head that can take such a role. However, to subsequently bind that dependent into the verb’s argument slot requires logical axioms about the nature of various dependents. This not only requires extra reduction steps to arrive at the desired logical form, but could also lead to problems depending on the underlying theory of roles.

We present an alternative approach to binding dependents, which overcomes these problems without abandoning the linguistic motivation. Because we work in a lexicalist setting, we can compile the effects of the *linguistic* linking theory directly into category assignments.

The first difference in our proposal is that arguments express only their own nominal, not the nominal of a head as well. For example, proper nouns receive categories such as (18).

$$(18) \quad Ed \vdash n : @_{d_1} \mathbf{Ed}$$

This entry highlights our relaxation of the strict connection between syntactic and semantic types traditionally assumed in categorial grammars, a move in line with the MRS approach.

In contrast with DGL, the semantic portion of a syntactic argument in our system does not declare the role it is to take and does not identify the head it is to be part of. Instead it identifies only its own referent. Without using additional inference steps, this is transmuted via unification into a form similar to DGL's in the result category. (19) is an example of the kind of head category needed.

$$(19) \quad sleeps \vdash s : @_{h_2} (\mathbf{sleep} \wedge \langle \mathbf{ACT} \rangle (i \wedge p)) \setminus n : @_i p$$

To derive *Ed sleeps*, (18) and (19) combine via backward application to produce (20), the same term as that built in DGL using one step instead of several.

$$(20) \quad @_{h_2} (\mathbf{sleep} \wedge \langle \mathbf{ACT} \rangle (d_1 \wedge \mathbf{Ed}))$$

To produce HLDS terms that are fully compatible with the way that Kruijff and Kruijff-Korbayová (2001) model discourse, we need to mark the informativity of dependents as *contextually bound* (CB) and *contextually nonbound* (NB). In DGL, these appear as modalities in logical forms that are used to create a topic-focus articulation that is merged with the discourse context. For example, the sentence *he wrote a book* would receive the following (simplified) interpretation:

$$(21) \quad @_{h_1} ([\mathbf{NB}] \mathbf{write} \wedge [\mathbf{NB}] \langle \mathbf{PAT} \rangle (d_5 \wedge \mathbf{book}) \wedge [\mathbf{CB}] \langle \mathbf{ACT} \rangle (d_6 \wedge \langle \mathbf{XS} \rangle (d_3 \wedge \mathbf{male})))$$

DGL uses feature-resolving unary modalities (Moortgat, 1997) to instantiate the values of informativity. In unification-based approaches such as CCG, the transferal of feature information into semantic representations is standard practice. We thus employ the feature *inf* and mark informativity in logical forms with values resolved syntactically.

$$(22) \quad Ed \vdash n_{inf=CB} : @_{d_1} \mathbf{Ed}$$

$$(23) \quad sleeps \vdash s : @_{h_2} ([\mathbf{NB}] \mathbf{sleep} \wedge [q] \langle \mathbf{ACT} \rangle (i \wedge p)) \setminus n_{inf=q} : @_i p$$

Combining these entries using backward application gives the following result for *Ed sleeps*:

$$(24) \quad s : @_{h_2} ([\mathbf{NB}] \mathbf{sleep} \wedge [\mathbf{CB}] \langle \mathbf{ACT} \rangle (d_1 \wedge \mathbf{Ed}))$$

A major benefit of having nominals in our representations comes with adjuncts. With HLDS, we consider the prepositional verbal modifier in the sentence *Ed sleeps in the bed* as an optional **Locative** dependent of *sleeps*. To implement this, we follow DGL in identifying the discourse referent of the head with that of the adjunct. However, unlike DGL, this is compiled into the category for the adjunct.

$$(25) \quad in \vdash (s : @_i (p \wedge [r] \langle \mathbf{LOC} \rangle (j \wedge q))) \setminus s : @_i p / n_{inf=r} : @_j q$$

To derive the sentence *Ed sleeps in the bed* (see Figure 1), we then need the following further entries:

$$(26) \quad the \vdash n_{inf=CB} : p / n_{inf=NB} : p$$

$$(27) \quad bed \vdash n_{inf=NB} : @_{d_3} \mathbf{bed}$$

This approach thus allows adjuncts to insert their semantic import into the meaning of the head, making use of nominals in a manner similar to the use of indexes in Unification Categorial Grammar.

## 5 Intonation and Information Structure

Information Structure (IS) in English is in part determined by intonation. For example, given the question in (28), an appropriate response would be (29).<sup>2</sup>

$$(28) \quad \text{I know what Ed READ. But what did Ed WRITE?}$$

$$(29) \quad (\text{Ed } \overset{\text{L+H*}}{\text{WROTE}}) \quad (\text{A } \overset{\text{H*}}{\text{BOOK}}).$$

Steedman (2000a) incorporates intonation into CCG syntactic analyses to determine the contribution of different constituents to IS. Steedman calls segments such as *Ed wrote* of (29) the **theme** of the sentence, and *a book* the **rheme**. The former indicates the part of the utterance that connects it with the preceding discourse, whereas the latter provides information that moves the discourse forward.

In the context of Discourse Representation Theory, Kruijff-Korbayová (1998) represents IS by splitting DRT structures into a topic/focus articulation of the form  $\boxed{\mathbf{TOPIC}} \boxtimes \boxed{\mathbf{FOCUS}}$ . We represent

<sup>2</sup>Following Pierrehumbert's notation, the intonational contour L+H\* indicates a low-rising pitch accent, H\* a sharply-rising pitch accent, and both LH% and LL% are boundary tones.

$$\begin{array}{c}
\frac{Ed\ sleeps\ (= (24))}{s : @_{h_2}([\text{NB}]\mathbf{sleep} \wedge [\text{CB}]\langle \text{ACT} \rangle(d_1 \wedge \mathbf{Ed}))} \quad \frac{in}{s : @_i(p \wedge [r]\langle \text{LOC} \rangle(j \wedge q)) \setminus s : @_i p / n_{inf=r} : @_j q} \quad \frac{the}{n_{inf=CB} : s / n_{inf=NB} : s} \quad \frac{bed}{n_{inf=NB} : @_{d_3} \mathbf{bed}} \\
\hline
\frac{n_{inf=CB} : @_{d_3} \mathbf{bed}}{s : @_i(p \wedge [\text{CB}]\langle \text{LOC} \rangle(d_3 \wedge \mathbf{bed})) \setminus s : @_i p} \\
\hline
s : @_{h_2}([\text{NB}]\mathbf{sleep} \wedge [\text{CB}]\langle \text{ACT} \rangle(d_1 \wedge \mathbf{Ed}) \wedge [\text{CB}]\langle \text{LOC} \rangle(d_3 \wedge \mathbf{bed}))
\end{array}$$

Figure 1: Derivation of *Ed sleeps in the bed*.

this in HLDS as a term incorporating the  $\bowtie$  operator. Equating topic and focus with Steedman’s theme and rheme, we encode the interpretation of (29) as:

$$(30) \quad @_{h_7}([\text{CB}]\mathbf{write} \wedge [\text{CB}]\langle \text{ACT} \rangle(d_1 \wedge \mathbf{Ed}) \bowtie [\text{NB}]\langle \text{PAT} \rangle(d_4 \wedge \mathbf{book}))$$

DGL builds such structures by using a rewriting system to produce terms with topic/focus articulation from the terms produced by the syntax.

Steedman uses the pitch accents to produce lexical entries with values for the *INFORMATION* feature, which we call here *sinf*. L+H\* and H\* set the value of this feature as  $\theta$  (for theme) or  $\rho$  (for rheme), respectively. He also employs categories for the boundary tones that carry blocking values for *sinf* which stop incomplete intonational phrases from combining with others, thereby avoiding derivations for utterances with nonsensical intonation contours.

Our approach is to incorporate the syntactic aspects of Steedman’s analysis with DGL’s rewriting system for using informativity to partition sentential meaning. In addition to using the syntactic feature *sinf*, we allow intonation marking to instantiate the values of the semantic informativity feature *inf*. Thus, we have the following sort of entry:

$$(31) \quad \text{WROTE (L+H*)} \vdash \\
s_{sinf=\theta} : \phi \setminus n_{inf=w, sinf=\theta} : @_i p / n_{inf=x, sinf=\theta} : @_j q \\
\phi = @_{h_2}([\text{CB}]\mathbf{write} \wedge [w]\langle \text{ACT} \rangle(i \wedge p) \wedge [x]\langle \text{PAT} \rangle(j \wedge q))$$

We therefore straightforwardly reap the syntactic benefits of Steedman’s intonation analysis, while IS itself is determined via DGL’s logical form rewriting system operating on the modal indications of informativity produced during the derivation. The articulation of IS can thus be performed uniformly across languages, which use a variety of strategies including intonation, morphology, and word order variation to mark the informativity of different elements. The resulting logical form plugs directly

into DGL’s architecture for incorporating sentence meaning with the discourse.

## 6 Conclusions and Future Work

Since it is couched in hybrid logic, HLDS is ideally suited to be *logically engineered* to the task at hand. Hybrid logic can be made to do exactly what we want, answering to the linguistic intuitions we want to formalize *without* yielding its core assets – a rich propositional ontology, decidability, and favorable computational complexity.

Various aspects of meaning, like dependency relations, contextual reference, tense and aspect, and information structure can be perspicuously encoded with HLDS, and the resulting representations can be built *compositionally* using CCG. CCG has close affinities with dependency grammar, and it provides a competitive and explanatorily adequate basis for a variety of phenomena ranging from coordination and unbounded dependencies to information structure. Nonetheless, the approach we describe could in principle be fit into other unification-based frameworks like Head-Driven Phrase Structure Grammar.

Hybrid logic’s utility does not stop with sentential meaning. It can also be used to model discourse interpretation and is closely related to logics for knowledge representation. This way we can cover the track from grammar to discourse with a *single* meaning formalism. We do not need to translate or make simplifying assumptions for different processing modules to communicate, and we can freely include and use information across different levels of meaning.

We have implemented a (preliminary) Java package for creating and manipulating hybrid logic terms and connected it to Grok, a CCG parsing system.<sup>3</sup> The use of HLDS has made it possible to improve

<sup>3</sup>The software is available at <http://opennlp.sf.net> and <http://grok.sf.net> under an open source license.

the representation of the lexicon. Hybrid logic nominals provide a convenient and intuitive manner of localizing parts of a semantic structure, which has made it possible to greatly simplify the use of inheritance in the lexicon. Logical forms are created as an accumulation of different levels in the hierarchy including morphological information. This is particularly important since the system does not otherwise support typed feature structures with inheritance.

Hybrid logics provide a perspicuous logical language for representing structures in temporal logic, description logic, AVMs, and indeed any relational structure. Terms of HLDS can thus be marshalled into terms of these other representations with the potential of taking advantage of tools developed for them or providing input to modules expecting them.

In future work, we intend to combine techniques for building wide-coverage statistical parsers for CCG (Hockenmaier and Steedman, 2002; Clark et al., 2002) with corpora that have explicitly marked semantic dependency relations (such as the Prague Dependency Treebank and NEGRA) to produce HLDS terms as the parse output.

### Acknowledgements

We would like to thank Patrick Blackburn, Johan Bos, Nissim Francez, Alex Lascarides, Mark Steedman, Bonnie Webber and the ACL reviewers for helpful comments on earlier versions of this paper. All errors are, of course, our own. Jason Baldridge's work is supported in part by Overseas Research Student Award ORS/98014014. Geert-Jan Kruijff's work is supported by the DFG Sonderforschungsbereich 378 *Resource-Sensitive Cognitive Processes*, Project NEGRA EM6.

### References

Natasha Alechina. 1995. *Modal Quantifiers*. Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands.

Carlos Areces, Patrick Blackburn, and Maarten Marx. 1999. A road-map on complexity for hybrid logics. In J. Flum and M. Rodríguez-Artalejo, editors, *Computer Science Logic*, number 1683 in Lecture Notes in Computer Science, pages 307–321. Springer-Verlag.

Carlos Areces. 2000. *Logic Engineering. The Case of Description and Hybrid Logics*. Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands.

Patrick Blackburn and Alex Lascarides. 1992. Sorts and operators for temporal semantics. In *Proc. of the Fourth Symposium on Logic and Language*, Budapest, Hungary.

Patrick Blackburn. 1994. Tense, temporal reference and tense logic. *Journal of Semantics*, 11:83–101.

Patrick Blackburn. 2000. Representation, reasoning, and relational structures: a hybrid logic manifesto. *Logic Journal of the IGPL*, 8(3):339–625.

Stephen Clark, Julia Hockenmaier, and Mark Steedman. 2002. Building deep dependency structures using a wide-coverage CCG parser. In *Proc. of the 40th Annual Meeting of the Association of Computational Linguistics*, Philadelphia, PA.

Ann Copestake, Dan Flickinger, Ivan Sag, and Carl Pollard. 1999. Minimal recursion semantics: An introduction. [www-csli.stanford.edu/~aac/newmrs.ps](http://www-csli.stanford.edu/~aac/newmrs.ps).

Ann Copestake, Alex Lascarides, and Dan Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proc. of the 39th Annual Meeting of the Association of Computational Linguistics*, pages 132–139, Toulouse, France.

Julia Hockenmaier and Mark Steedman. 2002. Generative models for statistical parsing with combinatory categorial grammar. In *Proc. of the 40th Annual Meeting of the Association of Computational Linguistics*, Philadelphia, PA.

Geert-Jan M. Kruijff and Ivana Kruijff-Korbayová. 2001. A hybrid logic formalization of information structure sensitive discourse interpretation. In *Proc. of the Fourth Workshop on Text, Speech and Dialogue*, volume 2166 of LNCS/LNAI, pages 31–38. Springer-Verlag.

Ivana Kruijff-Korbayová. 1998. *The Dynamic Potential of Topic and Focus: A Praguian Approach to Discourse Representation Theory*. Ph.D. thesis, Charles University, Prague, Czech Republic.

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

Marc Moens and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational Linguistics*, 14:15–28.

Michael Moortgat. 1997. Categorial type logics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*. Elsevier Science B.V.

Glyn V. Morrill. 1994. *Type Logical Grammar: Categorial Logic of Signs*. Kluwer Academic Publishers, Dordrecht, Boston, London.

Mark Steedman. 2000a. Information structure and the syntax-phonology interface. *Linguistic Inquiry*, 34:649–689.

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

Henk Zeevat. 1988. Combining categorial grammar and unification. In Uwe Reyle and Christian Rohrer, editors, *Natural Language Parsing and Linguistic Theories*, pages 202–229. Reidel, Dordrecht.