# A MODEL-THEORETIC FORMULATION OF DEPENDENCY GRAMMAR

**Norbert Bröker**

*Institute for Natural Language Processing*
*University of Stuttgart*
*Azenbergstraße 12*
*70194 Stuttgart, Germany*
*nobi@ims.uni-stuttgart.de*

Abstract: In this talk, we give a mathematically precise characterization of Dependency Grammar (DG). We sketch the ideas of model theory and modal logic, and apply them to DG. The resulting description distinguishes between linguistic objects (e.g., syntactic structures) and their description in terms of logical formulae. Complex descriptive mechanisms, such as underspecification and disjunction, are handled in the formulae, while the syntactic structures themselves are very simple. We illustrate the idea of coupling several descriptional dimensions with a word order description separated from the syntactic structure, which allows the dependency relations to be non-projective.

Keywords: Dependency Grammar, Syntax, Word Order, Model Theory, Modal Logic.

## 1. INTRODUCTION

In recent years, relational descriptions of language not based on configurations of categories have (re)gained popularity, both in linguistic theory (as theta-role, subcategorization, or domain of locality) and in processing accounts (Rambow and Joshi, 1994; Eisner, 1997). The most radical of these approaches, Dependency Grammar (DG, Tesnière (1959), Hudson (1993)), completely eliminates nonlexical categories and builds on binary relations between words only. Unfortunately from a theoretical and computational point, there is no convincing formal framework for DG.

This work proposes a state-of-the-art formal framework for Dependency Grammar. It defines a (multi-)modal logical language (Fitting, 1984) for describing dependency structures, and gives a model-theoretic semantics in terms of annotated graphs. This mathematical formulation

ISBN: 0 08 043 438X

addresses the question of how DGs may be constrained in three ways: First, only a precise account of DG provides the basis for formal results on, e.g., generative capacity (exemplified by Neuhaus and Bröker (1997)). Second, the proposal is representational and non-derivational, eliminating rules of various sorts as a source of complexity. These two effects apply to any model-theoretic approach, while the third is specific to the choice of modal logic as formal basis: Previous work on different varieties of modal logic may suggest some avenues of further restricting the mathematical power of our model.

Prior proposals to formalize DG suffer from a lack in expressive power, from formal inconsistencies, or employ such a big number of mechanism that not all of them have been successfully integrated or even specified. For example, identifying DG with context-free grammars (Gaifman, 1965; Lombardo and Lesmo, 1996) is clearly insufficient from an empirical point of view due to the problem of non-projective dependencies (cf. the argument in Neuhaus and Bröker (1997)). Dependency Unification Grammar (Hellwig, 1993) does not clearly distinguish between descriptional devices such as disjunction and the analysis structures themselves, and cannot properly describe discontinuities. Word Grammar (Hudson, 1990) tries to separate analysis structures from their descriptions, providing a propositional language for linguistic specification. Unfortunately, the notion of inheritance as defined by Hudson (1990) is formally unsound, and the word order description given there suffers from formal inconsistencies as well. The Prague tradition of Functional Generative Description (Sgall et.al., 1986) has long been described informally or by giving algorithms only, but recently Kruijff (1997) proposed a very interesting categorial approach. As other categorial grammars, it has the drawback not to be representational, but rather to encode word order in rule application sequences. Our proposal will be representational, i.e., will assign static structures to linguistics objects, which are constructed to satisfy descriptive logical formulae.

In the following, we will first give an overview of model theory, its usage in syntax, and sketch modal logic as one of ist instantiations. We then define dependency structures to consist of several related levels of representation, and illustrate how dependency structures may be described by modal logic. We develop logical descriptions for atomic properties of words and relational structures such as dependency trees, and discuss how one might relate the dependency tree to other levels of representation. We will also propose a new description of word order in DG by so-called word order domains, and discuss their logical specification.

## 2. MODEL THEORY

Model theory (cf. slide 2) emphasizes the existence of a priori structures, the *models*, which represent the empirical domain. They are built by abstracting from the domain properties which cannot or shall not be described. In a second step, a logical *calculus* is defined whose formulae describe the model structures. The so-called *satisfaction relation* for formulae tightly links the logic to the models by assigning to each formulae a set of models satisfying the formulae. (One might of course view it the other way round: Given a model, you can check whether it satisfies a certain formula.) Predictions derived from the calculus (e.g., by inference rules) correspond to properties of the models, and may be tested against the empirical domain (these predictions of course carry over only within the limits of the abstraction applied).

Such a setup has several wellcome consequences:

1) In many cases, model structures may be taken over from previous theoretical research; in the case of linguistics, e.g., syntactic representations can often be used directly as model structures.

2) The separation between object and meta level of description (i.e., the distinction between models and formulae) allows one to have simple models and at the same time complex and expressive descriptions. Underspecification and ambiguity can be dealt with at the formula level, while syntactic representations themselves are always fully specified and disambiguated. E.g., a description of an NP unspecified for case could be mapped by the satisfaction relation

ISBN: 0 08 043 438X

to the set of all NPs, each specified for case. This distinction has consequences for the clarity of the models, for the syntactic and lexical specification, and for comparison to other theories on the meta level (Blackburn, 1994; Kracht, 1995).

3) The mathematical precision allows to derive results on generative capacity or computational complexity. The question of whether DG is a notational variant of context-free grammars (Gaifman, 1965) can then be evaluated anew. Work on logical calculi has often revealed certain mechanism to introduce complexity (e.g., negation in attribute-value formalisms), which might then be inspected whether they are strictly necessary or can be substituted by a less complex, but empirically still adequate mechanism.

There is now a number of model-theoretic approaches to syntax (cf. slide 3), and Rogers and Cornell (1997) have presented a tutorial with bibliography on the subject. Applications range from mathematical investigations (e.g., the complexity issue) to questions of independency and redundancy of parts of a theory.

## 3. MODAL LOGIC

Modal logic (cf. slide 4) constitutes a family of related calculi based on very simple models. These so-called *Kripke models* consist of a set of objects, a set of properties of the objects, and a set of binary relations between objects. The properties are represented by subsets of objects containing all objects having this particular property.

Although modal logic, and therefore Kripke models, are usually discussed in the context of beliefs and possible worlds, the models themselves constitute simple graphs with annotated nodes: They consist of a set of nodes, linked by some relations, and annotated with certain properties. These are exactly the building blocks of syntactic structure, and we will identify objects with, e.g., words, properties with features, and relations with hierarchical dependencies.

The set of formulae is - in the simplest case - built from only four primitives: For each property, there is a suitably named propositional *atom* $\mathbf{p}_i$; for each relation type, there is a suitably named modal operator, or *modality*, $\mathbf{r}_i$; and there is a truth-functionally complete set of Boolean connectives (here, conjunction and negation).

The definition of the satisfaction relation differs from that of first-order logic in an important respect: Modal satisfaction is determined locally, i.e., with respect to a certain object in the Kripke model. So we speak of a formula being satisfied in a model $\mathbf{M}$ at object $\mathbf{o}$. A propositional atom $\mathbf{p}_i$ is satisfied in $\mathbf{M}$ at $\mathbf{o}$ if and only if $\mathbf{o}$ has the property indicated by $\mathbf{p}_i$. A modality $\mathbf{r}_i$ allows to impose restrictions only on neighbors of $\mathbf{o}$, so that we never have to inspect the complete model, but only those objects related to $\mathbf{o}$ (possibly across several relations). The satisfaction of formulae containing Boolean connectives is defined in the standard way.

The example on slide 5 gives a model in graphical and set notation, and a formula describing part of the model.

## 4. A LOGIC FOR DEPENDENCY TREES

The transfer to a modal logic describing dependency trees is trivial, and shown in two example structures. First, the words of the sentence are taken to be the objects of the Kripke model, linked by dependency relations (cf. slide 6). We define one modality for each dependency type, and can then encode the hierachical structure of the dependency tree using a straighforward formula. Properties of words, such as the surface realization or the word class, are viewed as

ISBN: 0 08 043 438X

simple properties and give rise to a set of modal atoms, which are conjoined as appropriate to describe the individual words (cf. slide 7).

Note that this translation takes linguistic notions such as dependency relation or word class seriously, and only gives them a formal representation. Hence, it is very easy to translate a conventional stemma into a logical formula. This is one of the benefits of using pre-existing model structures in our model-theoretic framework.

## 5. MULTIPLE LEVELS OF DESCRIPTION

Clearly, one would like to add more information to the dependency tree. We briefly mention two of them, morphosyntactic features and conceptual interpretations, and give more detail on a third, namely word order descriptions.

The basic idea is very simple (cf. slide 8): For each additional level of representation, we define Kripke models; for example, attribute-value matrices can be viewed as a special notation for Kripke models of feature annotation (Blackburn and Spaan, 1993). This gives rise to additional modal atoms (e.g., feature values) and additional modalities (e.g., feature labels), but in principle nothing changes. To link the levels, one defines a mapping relation associating words (objects in the dependency tree) with feature points (objects in the feature structure); this mapping is a binary relation between objects of the combined model and thus adds another modality. Such an architecture is comparable to the projection idea of Lexical-Functional Grammar (Kaplan and Bresnan, 1982), which maps nodes of different representation levels onto each other.

In this way, one may include a broad range of information into the syntactic representation without reverting to a stratified theory with rules mapping one level onto another. The example on slide 9 gives model and formula for a dependency tree associated with a frame-like semantic structure.

## 6. WORD ORDER DOMAINS

Turning to word order, we first review existing approaches (cf. slide 10). There are procedural descriptions, which are typically not applicable to both analysis and synthesis (Hajicova, et.al., 1993), and which often have empirically inadequate limitations coded into the algorithm (such as the meta-projectivity constraint in Covington (1990)). Dependency Unification Grammar also relies on a special interpretation of certain features, using them to define abstract positions for modifiers. It cannot, however, express island constraints. Binary precedence predicates have also been adapted from phrase-structure grammars, but they lead to formal inconsistencies in a pure DG (Hudson, 1990), or require phrasal, i.e., non-lexical categories (Pericliev, 1997).

An alternative approach is illustrated in slide 11. We define Word Order Domains as sets of words, which roughly correspond to the positions in DUG. Each word may define a sequence of such order domains, into which it is placed together with its modifiers. In the example, „*know*" defines two order domains, $\mathbf{o}_1$ for the topicalized position, and $\mathbf{o}_2$ for the rest of the sentence. Order domains additionally limit the scope of precedence predicates, thereby avoiding the formal inconsistencies mentioned above. E.g., the precedence predicate **self<obj** is satisfied within domain $\mathbf{o}_2$ since within this domain, no object precedes the verb „*likes*" which introduced the precedence predicate.

As the example shows, discontinuous dependencies arise when a word is not contained in a word order domain of its direct head, but rather in a domain of another transitive head („*Beans*" contained in $\mathbf{o}_1$ defined by „*know*" ). Under this view, continuous attachment is a special case of discontinuous attachment, namely when the direct head of a word coincides with the transitive head defining the order domain it is placed into.

ISBN: 0 08 043 438X

Limiting discontinuities to certain constructions is possible with reference to the dependency types. Note that usually the syntactic construction is assumed to determine extractability, and that it is represented in DG by different dependency relations (for subject vs. object, for example). E.g., the determiner may not be extracted from the NP, while the relative clause may be (cf. slide 12). We can encode this restriction by saying that a modifier occupying dependency relation d may be extracted across a set of dependency relations $\{d_1, \dots d_n\}$. If this set is empty (i.e., $n=0$), it may not be extracted at all, and continuous attachment is required. The set of German examples indicates that extraction (of a relative clause) from the subject is possible, and that extracting an object from a infinitival verb phrase is possible. But, as the last example shows, extracting an object from an infinitival verb phrase used as subject is impossible. Such restrictions can easily be encoded in our scheme.

The most prominent feature of this approach to word order is that it separates the syntactic dependency tree from the surface word order (cf. slide 13). Word order is delegated to another level of description, and can be viewed as an additional constraint on syntactic relations. Note that the order domains can be partially ordered by set inclusion, and then form a projective tree. In a way, the order domains encode the context-free backbone of phrase-based grammars. In particular, phrasal categories in these approaches constitute scopes for precedence constraints (formulated over sister nodes), while here order domains play this role.

## 7. EXTENDING THE LOGIC TO ORDER DOMAINS

We can easily define model structures for order domains (cf. slide 14). The dependency tree is enriched by adding order domain nodes and a relation from words to order domains. Additional restrictions enforce projectivity of the order domain structure. The logical language (cf. slide 15) is extended by a modality mapping a word to its order domains, and by some modal atoms expressing cardinality restrictions on order domains or feature restrictions on elements contained in an order domain. Even precedence predicates may be converted into modal atoms and given a semantics in terms of a one-place predicate.

Slide 16 shows a real-world example of a lexical specification using this description language. It encodes (aspects of) the topological fields in the German main clause. We assume the finite verb to govern the main clause, and define the field structure as follows: Any word of class VerbFinite defines three order domains (first 3 lines). The first order domain contains exactly one element with features **[fld ini]**. It will contain the relative NP or PP in case of a verb-final relative clause, and is therefore unspecified for the rel feature. The second order domain contains an arbitrary number of elements, all of which have the features **[fld mid, rel -]**. The third field contains at most one element with features **[fld fin, rel -]** . The next three equivalences define the order types. The verb is of order type **V2**, corresponding to a verb-second clause, iff it occurs in the middle field, precedes all other elements within this field, and if the initial field does not contain a relative NP or PP. The verb is of order type **VEnd**, corresponding a verb-final clause, iff it occurs in the middle field and follows all other elements in this field. The verb is of order type **V1**, corresponding a verb-initial clause, iff is occurs in the initial field and has the feature **[rel -]**.

Apart from certain specifics in the analysis of subordinating conjunctions and the relative clause, it is important to note that this formula describes in a compact way all order alternative of finite verbs. In particular, attaching a modifier to the verb which precedes it in textual order results in the elimination of the **V1** disjunct, while the other alternatives remain. Put differently, no order-induced structural ambiguities arise during parsing because an underspecified representation represents all remaining ambiguities without introducing structural alternatives. This cuts down the search space for parsing considerably.

ISBN: 0 08 043 438X

## 8. FINAL REMARKS

We have presented a formal framework (cf. slide 17) for specifying dependency grammars which includes separate, but linked representation levels for dependency, morphosyntactic features, word order, and conceptual interpretation. Due to the separation of object level (model structure) and meta level of description (formulae), complex descriptions involving underspecification or ambiguity may easily be written. The choice of modal logic as a formal basis makes a wealth of mathematical results available and yields a description easily translatable into standard linguistic terminology.

Conceiving of the dimensions as separate restrictions of a basic dependency relation retains the positive properties of DG: Semantic interpretation is facilitated due to the semantic motivation of the dependencies, order ambiguities can be factored out in a separate dimension (in contrast to other lexicalized grammars such as TAG and CG, which suffer from systematic ambiguities in this respect (Joshi and Srinivas, 1994)), and the representation is integrated yet does not assign priorities to knowledge sources (such as the syntax-first approach implicit in MTT (Melc'uk, 1988)).

The framework provides for mathematical explicitness and precision, and can thus serve as basis for further investigations into the mathematical properties of DG. Since it is representational and does not employ tansformations of any sort, its formal power can easily be determined.

Future work can proceed in several directions (cf. slide 19): One might try to enrich the order domain structure to encode what is usually termed information structure, or one might tackle the question of how coordination and ellipsis is best described in a dependency grammar.

## REFERENCES


Blackburn, P. (1994).Structures, Languages and Translations: The Structural Approach to Feature Logic. In: *Constraints, Language and Computation* (C. Rupp, M. Rosner and R. Johnson (Ed.)), pp. 1--27. Academic Press, London.

Blackburn, P. and E. Spaan (1993). A Modal Perspective on the Computational Complexity of Attribute Value Grammar. In: *Journal of Logic, Language and Information* , **2**, pp. 129--169.

Covington, M. (1990). Parsing Discontinuous Constituents in Dependency Grammar. In: *Computational Linguistics* , **16(4)**, pp. 234--236.

Eisner, J. (1997). Bilexical Grammars and a cubic-time probabilistic parser. In: *Proc. Int'l Workshop on Parsing Technology*, pp. 54--65. Massachussetts Institute of Technology, Boston/MA.

Fitting, M. (1984). Basic Modal Logic. In: *Handbook of Philosophical Logic* (D. Gabbay and F. Guenthner (Ed.)), pp. 365--448. D.Reidel, Dordrecht/NL.

Gaifman, H. (1965).Dependency Systems and Phrase Structure Systems. In: *Information and Control*, **8**, 304--337.

Hajicova, E., Sgall, P. and Skoumalova, H. (1993). Identifying Topic and Focus by an automatic procedure In: *Proc. of the 6th Conference of the European Chapter of the ACL*, pp. 178--182.

Hellwig, P. (1993). Extended Dependency Unification Grammar. In: *Functional Description of Language* (E. Hajicova (Ed.)), pp. 163--178. Charles-University, Prague.

Hudson, R. (1990). *English Word Grammar*. Basil Blackwell, Oxford/UK.

Hudson, R. (1993). Recent developments in dependency theory. In: *Syntax. Ein internationales Handbuch zeitgenössischer Forschung* (J. Jacobs, A. v. Stechow, W. Sternefeld and T. Vennemann (Ed.)), pp. 329--338. Walter de Gruyter, Berlin.

Joshi, A. and B. Srinivas (1994). Integration of Structural and Statistical Information: Role of Complexity of Description of Primitives. In: *Proc. Int'l Conference on New Methods in Language Processing*, pp. 136--143. Univ. of Manchester, Manchester/UK.


ISBN: 0 08 043 438X

Kaplan, R. and J. Bresnan (1982). Lexical-Functional Grammar: A Formal System for Grammatical Representation. In: *The Mental Representation of Grammatical Relations* (J. Bresnan and R. Kaplan (Ed.)), pp. 173--281. MIT Press, Cambridge/MA.
Kracht, M. (1995). Is there a genuine modal perspective on feature structures? In: *Linguistics and Philosophy*, **18**, pp. 401--458.
van Kruijff, G.-J. (1997). *A Basic Deependency-Based Logical Grammar*. Manuscript. Charles-University, Prague.
Lombardo, V. and L. Lesmo (1996). An Earley-type recognizer for dependency grammar. In: *Proc. 16th Int'l Conf. on Computational Linguistics*, pp.723--728. Copenhagen/DK.
Melc'uk, I. and N. Pertsov (1987). *Surface Syntax of English: A Formal Model within the MTT Framework*. John Benjamins, Philadelphia/PA.
Neuhaus, P. and N. Bröker (1997). The Complexity of Recognition of Linguistically Adequate Dependency Grammars. In: *Proc. 35th Annual Meeting of the ACL and 8th Conf. of the EACL*, pp. 337--343. Madrid.
Pericliev, V. (1997). *From the Principle of Projectivity and Partial Orderings to Sophisticated Linearization Constraints*. Talk given at the workshop "How to constrain Dependency Grammar" during CIL-16, Paris.
Rambow, O., A. Joshi (1994). A Processing Model for Free Word Order Languages. In: *Perspectives on Sentence Processing* (C. J. Clifton, L. Frazier and K. Rayner (Ed.)). Lawrence Erlbaum, Hillsdale/NJ.
Rogers, J. and T. Cornell (1997). *Logical Aproaches to Syntactic Theories*. Tutorial given at the 35th Annual Meeting of the ACL and 8th Conf. of the EACL, Madrid.
Sgall, P., E. Hajicova and J. Panevova (1986). *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. D.Reidel, Dordrecht/NL.
Tesnière, L. (1959). *Eleménts de syntaxe structurale*. Klincksiek, Paris.

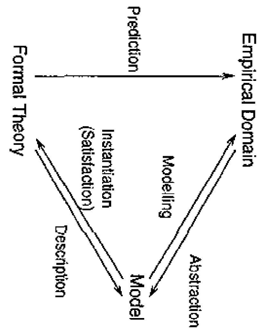
APPENDIX: THE SLIDES

ISBN: 0 08 043 438X

2

## I Model Theory




- Abstraction determines scope of the theory
- Satisfaction tightly links Formal Theory and Model
- Predictions can be derived from Formal Theory
- separation of object level and meta level
  - complex descriptive primitives
  - comparison of meta levels
- mathematical precision

3

## I Model-Theoretic Syntax

| | | |
|---|---|---|
| **CG** | Lambek '58 | |
| | v. Benthem '95 | |
| **GB** | Johnson '89 | redundancy of levels |
| | Stabler '92 | consistency of subset of principles |
| | | redundancy of principles |
| **GPSG** | Blackburn '93, | |
| | Kracht '95 | |
| **HPSG** | Carpenter '92 | (HPSG is undecidable) |
| **TAG** | Rogers & Vijay-Shanker '92, | |
| | Abrusci, Fourqueré & Vauzcilles '96 | |
| **DG** | (Hudson '90) | |
| | Bröker '97 | parsing is $\mathcal{NP}$-complete |

ISBN: 0 08 043 438X

## I Modal Logic & Kripke Models

Kripke Model $M = \langle O, P, R \rangle$

$O = \{o_1, o_2, \ldots\}$ set of objects

$P = \{P_1, P_2, \ldots\}$, set of predicates

$P_i \subseteq W$

$R = \{r_1, r_2, \ldots\}$, set of relations

$r_1 \subseteq (O \times O)$

Set of Modal Formulae $L$

$p_i$ iff $P_i \in P$

$\diamond_{r_i}\phi$ iff $r_i \in R, \phi \in L$

$\phi \wedge \psi$ iff $\phi, \psi \in L$

$\neg\phi$ iff $\phi \in L$

Satisfaction Relation $\models$

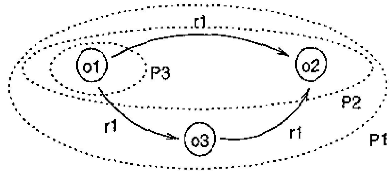
$M, o \models p_i$ iff $o \in P_i$

$M, o \models \diamond_r \phi$ iff $\exists o' \in O : \langle o, o' \rangle \in r$ and $M, o' \models \phi$

$M, o \models \phi \wedge \psi$ iff $M, o \models \phi$ and $M, o \models \psi$

$M, o \models \neg\phi$ iff not $M, o \models \phi$

## I Example Kripke Model




$M = \langle O, P, R \rangle$

$O = \{o_1, o_2, o_3\}$

$P = \{P_1, P_2, P_3\}$

$P_1 = \{o_1, o_2, o_3\}$

$P_2 = \{o_1, o_2\}$

$P_3 = \{o_1\}$

$R = \{r_1\}$

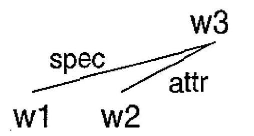
$r_1 = \{\langle o_1, o_2 \rangle, \langle o_1, o_3 \rangle, \langle o_2, o_3 \rangle\}$

$M, o_1 \models p_1 \wedge p_2 \wedge p_3 \wedge$
$\diamond_{r_1}((p_1 \wedge p_2) \vee p_2)$

ISBN: 0 08 043 438X

6

## II Dependency Relations

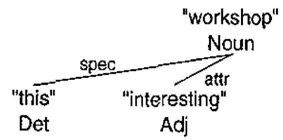



$$
\begin{aligned}
M = \langle W, P, D \rangle \\
W &= \{w_1, w_2, w_3\} && \text{set of words} \\
P &= \{p\} && \text{set of properties} \\
& \quad p = \{w_1, w_2, w_3\} \\
D &= \{spec, attr\} && \text{set of dependency relations} \\
& \quad spec = \{\langle w_3, w_1 \rangle\} \\
& \quad attr = \{\langle w_3, w_2 \rangle\}
\end{aligned}
$$

$$
\begin{aligned}
M, w_3 \models\ & p \wedge \\
& \diamond_{spec}\, p \wedge \\
& \diamond_{attr}\, p
\end{aligned}
$$

7

## II Atomic Properties




$$
\begin{aligned}
M = \langle W, P, D \rangle \\
W &= \{w_1, w_2, w_3\} \\
P &= \{\text{“workshop”}, \text{“this”}, \ldots, \text{Noun}, \text{Adj}, \ldots\} \\
& \quad \text{“workshop”} = \{w_3\} \\
& \quad \text{Adj} = \{w_2\} \\
& \quad \vdots \\
D &= \{spec, attr\} \\
& \quad spec = \{\langle w_3, w_1 \rangle\} \\
& \quad attr = \{\langle w_3, w_2 \rangle\}
\end{aligned}
$$

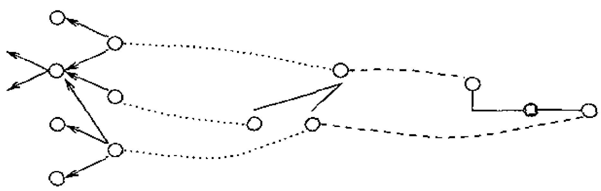
$$
\begin{aligned}
M, w_3 \models\ & \text{“workshop”} \wedge \text{Noun} \wedge \\
& \diamond_{spec} (\text{“this”} \wedge \text{Det}) \wedge \\
& \diamond_{attr} (\text{“interesting”} \wedge \text{Adj})
\end{aligned}
$$

ISBN: 0 08 043 438X

8

## II Multiple Levels

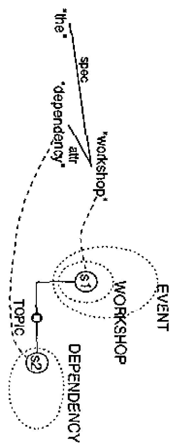



- several levels with different representational needs
- non-derivational representation

⇒ incorporate several levels into one model

9

## II Integrating Levels




$M = \langle W, P, D, S, C, R, V_S \rangle$

$\vdots$

| | |
|---|---|
| $S = \{s_1, s_2\}$ | set of semantic objects |
| $C = \{EVENT, \ldots\}$ | set of semantic predicates |
| $\quad WORKSHOP = \{s_1\}$ | (concepts) |
| $\vdots$ | |
| $R = \{TOPIC\}$ | set of semantic relations |
| $\quad TOPIC = \{\langle s_1, s_2 \rangle\}$ | (roles) |
| $V_S = \{\langle w_3, s_1 \rangle, \langle w_2, s_2 \rangle\}$ | mapping from words to semantic objects |

$M, w_3 \models$ "workshop" $\wedge \diamond_S$ WORKSHOP $\wedge$
$\diamond_{spec}$"the" $\wedge$
$\diamond_{attr}$("dependency" $\wedge \diamond_S$ DEPENDENCY)$\wedge$
$(\diamond_S \diamond_{TOPIC} = \diamond_{attr} \diamond_S)$

ISBN: 0 08 043 438X

10

## III Proposed Order Descriptions

**procedural**

- one-directional — Hajicova et.al. '93
- stack of discontinuities — Covington '90

**abstract positions**

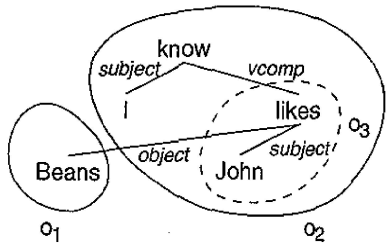
- no island constraints — Hellwig '80, '86

**(binary) precedence constraints** (ID/LP format)

- formal inconsistencies — Hudson '90
- no islands, phrasal categories — Pericliev

11

## III Word Order Domains

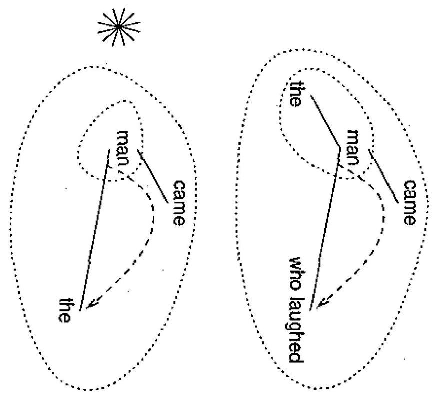



| Word | Domains | LPCs |
|---|---|---|
| know: | $o_1$, $o_2$ | `subj < self` |
| | | `self < vcomp` |
| likes: | $o_3$ | `subj < self` |
| | | `self < obj` |

ISBN: 0 08 043 438X

## III Islands




**Discontinuity depends on Construction**

Der Notebook ist schnell, den ich gekauft habe.
Einen Notebook zwingt er uns zu kaufen.
Einen Notebook zu kaufen, ist richtig.
* Einen Notebook ist richtig zu kaufen.

## VI Remarks On Order Domains

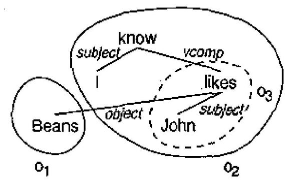
Sequence Of Words
*Is Not*
The Yield Of The Tree

- Word Order Domains Constitute Another Level
  (cf. information structure)
- Discontinuous Attachment Is Generalization Of Continuous Attachment
- Word Order Domains Isolate LPCs Just Like Phrasal Categories Do
  (cf. $S \Rightarrow XP\ VP$ For Topicalization)
- Discontinuities Are Bounded By Dependency Types,
  Which Represent Syntactic Constructions Just Like Phrasal Categories Do

ISBN: 0 08 043 438X

14

## III Extending the Model




$M = \langle W, P, D, O, V_O \rangle$

$\vdots$

$O = \{o_1, o_2, o_3\}$ set of order domains

$o_1 = \{w_1\}$

$o_2 = \{w_2, w_3, w_4, w_5\}$

$o_3 = \{w_4, w_5\}$

$V_O = \{\langle w_3, \langle o_1, o_2 \rangle\rangle, \langle w_5, \langle o_3 \rangle\rangle\}$ mapping from words to order domain sequences

Order Domain Structure $O$

- constitutes another description level
- obeys "projectivity"

15

## III Extending the Logic

Mapping from word to its $i$th order domain

$$M, w \models \diamond^i_{\mathcal{M}} \phi \text{ iff } \exists o \in O : \Phi_i(V_O(w)) = o \wedge M, o \models \phi$$

Restricting cardinality of order domain

$M, o \models$ single iff $|o| \leq 1$

$M, o \models$ mandatory iff $|o| \geq 1$

Requiring all elements to have certain properties

$$M, o \models p_i \text{ iff } \forall w \in o : P_i(w)$$

Expressing relative ordering

$$M, w \models <_{\{d_1,\ldots,d_n\}} \text{ iff } \exists o \in O : w \in o \wedge \forall w' \in o : \exists w'' \in W : (w \prec w' \vee \langle w'', w' \rangle \notin \bigcup_{i=1}^{n} d_i)$$

ISBN: 0 08 043 438X

16

## III Example Order Specification

$$\begin{aligned} \textsc{VerbFinite} \Rightarrow\ & \diamond^1_{\mathcal{M}} \text{ (single} \wedge \text{mandatory} \wedge \langle fld\rangle \text{ ini)} \\ \wedge\ & \diamond^2_{\mathcal{M}} \text{ (}\langle fld\rangle \text{ mid} \wedge \langle rel\rangle -\text{)} \\ \wedge\ & \diamond^3_{\mathcal{M}} \text{ (single} \wedge \langle fld\rangle \text{ fin} \wedge \langle rel\rangle -\text{)} \\ \wedge\ & \diamond_{\mathcal{U}}\langle order\rangle \text{ V2} \Leftrightarrow \begin{pmatrix} \diamond_{\mathcal{U}}\langle fld\rangle\text{mid} \\ \wedge\ <_* \\ \wedge\ \diamond^1_{\mathcal{M}}\langle rel\rangle - \end{pmatrix} \\ \wedge\ & \diamond_{\mathcal{U}}\langle order\rangle \text{ VEnd} \Leftrightarrow \begin{pmatrix} \diamond_{\mathcal{U}}\langle fld\rangle\text{mid} \\ \wedge\ >_* \end{pmatrix} \\ \wedge\ & \diamond_{\mathcal{U}}\langle order\rangle \text{ V1} \Leftrightarrow \begin{pmatrix} \diamond_{\mathcal{U}}\langle fld\rangle\text{ini} \\ \wedge\ \langle rel\rangle\text{-} \end{pmatrix} \end{aligned}$$

17

## VI Remarks On Formalization

Dependency Structures Are Described
By A
*(Multi-) Modal Logic*
With
Model-Theoretic Interpretation

- First Consistent Formalization Of DG
- Models Include Syntactic Tree, Order Domains, And Conceptual Interpretation
- Easy Integration Of Partial And Disjunctive Descriptions
- Based On (Standard) Modal Logic
- Simple Translation Between Linguistics and Formal Terms

ISBN: 0 08 043 438X

18

## IV Constraining DG

- framework which provides explicity and precision

- representational approach ⇒ no procedures in the model
- non-derivational approach ⇒ no mapping rules

- formal properties
  - relational vs. functional modalities
  - identify generative capacity

19

## IV Future Work

... **on the model** : Explore the notion "information structure"
... **on the logic** : Define a calculus
... **on the coverage** : Specify "scrambling", coordination
... **on the implications** : Collect cross-linguistic data on extraction

ISBN: 0 08 043 438X