Constituency and Semantic Interpretation

Torben Thrane
København

Abstract

The main point of this paper is to present an argument against phrase structure analysis as providing an efficient basis for an automated system that has language *understanding* as its primary goal. There are ususally several constraints on constituency analysis of the X-bar variety, the fundamental rule of which is $X^n \rightarrow SpecX^n\ X^{n-1}\ Comp^n$. Four such rule systems and their constraints are presented, and it is shown that only if one or more semantic constraints are taken into account can the number of potential tree structures be kept at a manageable level and result in 'correct' constituency analyses. But this appeal to semantics, it is argued, is ill advised as a means towards understanding, for it is only meant to secure a uniform *description of sentences*. A more viable appeal would be one according to which structural meaning is exploited for the purposes of constructing and revising models of *situations described by sentences*.

Context and Aim

The study of language is guided by a number of fundamental questions, among them the following:

1 a What constitutes knowledge of a language?
b How does such knowledge develop?
c How is such knowledge put to use?

I will be concerned here with certain aspects of the first and second of these problems (Chomsky 1981,32).

It has always been Chomsky's ultimate aim to answer 1b, and it has always been Chomsky's belief that an answer to 1b presupposes an answer to 1a that can be given in terms of an independent, autonomous description of language structure.

It is my ultimate aim to answer that part of 1c which is concerned with how we *understand* language, and it is my belief that the *purpose* of any investigation determines the format, methods, and principles to be adopted. It is, furthermore, my claim (cf. Thrane 1992a,b; 1993, fc) that computational linguistics in general has accepted Chomsky's belief, no matter what its purpose has been – and that this has prevented serious progress in the study of computational *understanding* of NL.

Chomsky's belief is the foundation of what might be called the *descriptive* paradigm in linguistics – cf. Chomsky (1981,33):

Proceedings of NODALIDA 1993, pages 277-289

> [W]e can say that a grammar constructed by a linguist is 'descriptively adequate' if it gives a correct account of the system of rules that is mentally represented, that is, if it correctly characterizes the rules and representations of the internally-represented grammar.

What I shall be specifically concerned with here is a central feature of that paradigm, the relationship between constituency rules and semantic interpretation. And although various forms of semantic motivation play a role in the choice of such rules, my conclusion will be that it is the wrong kind of semantic motivation when the purpose of the investigation is language *understanding* rather than language *description.*

PS-rules and constraints

There are two interpretations of any system of PS-rules:

2 a as an autonomous formalization of the knowledge of syntactic structure

b as a set of instructions for tree-building

There are at least four types of constraint on PS-rules:

3 a *assumptions* about the nature of PS-rules
[e.g. that terminals have already been exhaustively classified; that every constituent belongs to a category; that constituency is defined by movability, substitution and deletion; etc.]

b *graph-theoretic restrictions* on the formulation/application of PS-rules
[e.g. they must not lead to crossing branches; single-mother condition, etc.]

c *guidelines* for the formulation/application of PS-rules
[e.g. number of BAR-levels; type of recursiveness, etc.]

d *motivations* for the choice of PS-rules

Only 3d is my concern here, so I'll be a bit more specific about these. Two kinds of motivations for rule systems can be identified:

Data-oriented

4 a A rule system is chosen because it reveals structural differences between sentences S1 and S2 that correlate with *perceived differences* of meaning between S1 and S2.

b A rule system is chosen because it reveals structural properties of a sentence S that will play a role in *determining the meaning of S.*

Theory-oriented

c A rule system R is preferred over another R' because R is more constrained, consistent, and/or general than R'.

Only the data-oriented motivations are my concern here.

To keep the presentation at a manageable level, I shall confine myself to a discussion of PS-rules for the analysis of NP. (5) gives some data that should be handled by such rules. Even though the data are Danish and the rule systems to be discussed are for English, this shouldn't affect the general points being made.

5 a alle de mange andre drenge
all the many other boys
b de mange andre drenge
c mange andre drenge
d andre drenge
e *alle mange andre drenge
f alle andre drenge
g alle drenge
h drenge

The four rule systems to be mentioned are rivals within the Chomsky-tradition, to some extent reflecting its historical development. They are all post X-bar and therefore couched in X-bar terminology, even though one of them is not explicitly presented in such terms by its authors. They all assume a transformational component.

I explicitly mention only those constraints that are unique to the rule-system in question. All of them share such X-bar defining constraints as

- Designated Head
- Introduction of at most one lexical item per rule
- A lexical item introduced by a rule is the Head of the Phrase under analysis
- Allowance for cross-generalization

Four proposed rule systems

System 1 (Jackendoff 1977)

(a) N''' → (N'''|Art''') - N''
(b) N'' → (N'''|Q''') - (A''')* - N' - ...

Constraints

- Uniform Three-Level Hypothesis
- An NP specifier may contain at most one demonstrative, one quantifier, and one numeral. [Jackendoff's (semantic) *Specifier Constraint*]
- Specifiers are not strictly subcategorized for

Problems

- Presupposes both syntactic and semantic subcategorization of specifiers (and lexicon), otherwise ...
- ... it will generate just about *anything*

System 2 (Stuurman 1985; simplified wrt category vs. function distinction)

(a) X' → (Spec) {X|X'} ... [where X = {N|Art|Q} in our context]

Constraints

- Single Projection-Type Hypothesis
- Specifiers are constituents (they have a Head)
- At most one specifier per projection
- Requires a level of 'q-interpretation' (a non-PS, semantic process)

Problems

- Overgeneration: will generate 5e

System 3 (Wexler & Culicover's (1980) rules to generalize Bartsch's (1973) semantic constraints on NPs (inferred – but they assume X-bar theory))

(a) N'''_ (D) N''
(b) N'' _ (Q) N'
(c) N' _ (A) N

Constraints

- Base order generation significant
- Operator – operand organization for semantic interpretation

Problems

- Under-generation: will not generate 5a, and only 5b – f if *andre* is classified as A.

System 4 (Haegemann's (1991) NP-rules and 'Metarules')

(i) (a) N'' → Spec; N'
    (b) N'* → N'; XP
    (c) N' → N; XP

(ii) (a) X'' → Spec; X'
     (b) X'* → X'; YP
     (c) X' → X; YP

Constraints

- (i) is just a category-specific instantiation of (ii)
- Requires a representational (semantic) level of Logical Form

Problems

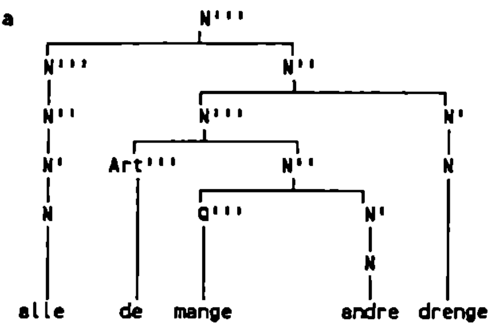
- Undergeneration: will not generate *any* of 5.

If we look at these four rule systems under interpretation 2a, they are clearly designed to answer questions 1a or b. Jackendoff's and Stuurman's rules are meant to provide partial answers to 1a, while Wexler & Culicover's and Haegeman's are designed to answer 1b. The members of each pair then differ among themselves. Jackendoff's and Wexler & Culicover's rules are data-oriented, whereas Stuurman's and Haegemann's are theory-oriented. There is nothing to choose between them, however, as far as the understanding vs. description dichotomy goes. They are all descriptive.

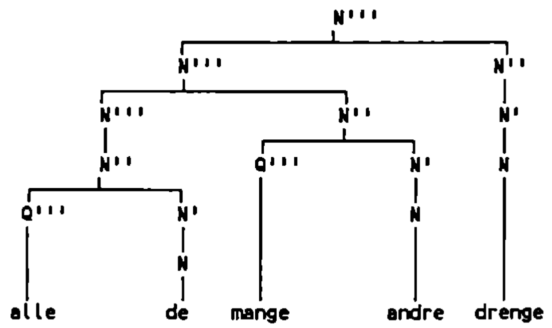
Computable Representations

Under interpretation 2b of a rule-system and its associated constraints, a parser is an implementation of a computational process which feeds on information provided by grammar rules and constraints, and then converts one representation – in the form of a NL sentence – into another representation – in the form of a tree. In this sense, trees are *computable representations*. The following are samples of trees computed from the rule systems and constraints we have been looking at.

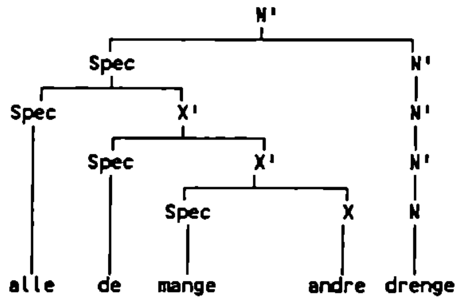
1a



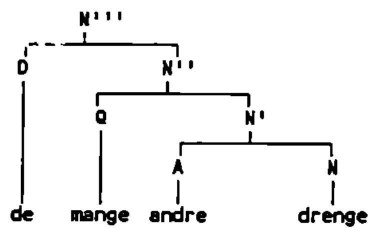
1b



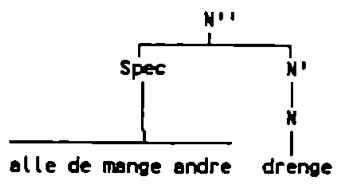
2



3



4



Now, if we horse around with various combinations of constraints and rule-systems, we find there are numerous theoretically possible tree-structures for 5a. If we relax all constraints except that every constituent must have a Head in connection with Jackendoff's rules, we get 312 different structures. If we add one – that *drenge* is the Designated Head – we reduce the number to 79. These statistics are fairly uninteresting. But what is interesting is that *only appeal to some semantically based constraint or motivation will produce the sort of configuration that is seriously considered in works on Phrase Structure.*

Despite this, we cannot assume that semantic motivations by *itself* will lead to the postulation of particular rule-systems. Consider the first semantic motivation (4a) in relation to 6:

6 a drengen købte en is
the boy bought an icecream
b en dreng købte en is
a boy bought an icecream

There is a perceived difference of meaning between 6a and b, which is the same in English as in Danish. None of the rule systems we've looked at would be prepared to propose different syntactic structures for 6a and b. So, a perceived difference in meaning is in itself neither a sufficient nor a necessary condition for proposing different syntactic structures.

Nevertheless, this seems to be precisely what we need to account for language *understanding*: to be able to say that perceived differences in *grammatical* meaning correlate with differences in computable representations – only that these representations are of a different sort from the tree-structures that we have been concerned with so far.

The difference can be explained with reference to the illustration of the relations between language, 'mind' and reality in Figure 1:

There are apparently three computable representations in this diagram:

- the tree is a representation of the syntactic structure of the sentence *it's a box* - assumed to be created on the basis of syntactic knowledge

- the house is a representation of a real house – assumed to be created on the basis of information provided by visual perception

These two are similar in being representations of the phenomena that gave rise to them. They are, in my terms, created on the basis of descriptive information, and they have *inclination of fit* towards a target which is identical to their source. They are *source-inclined.*

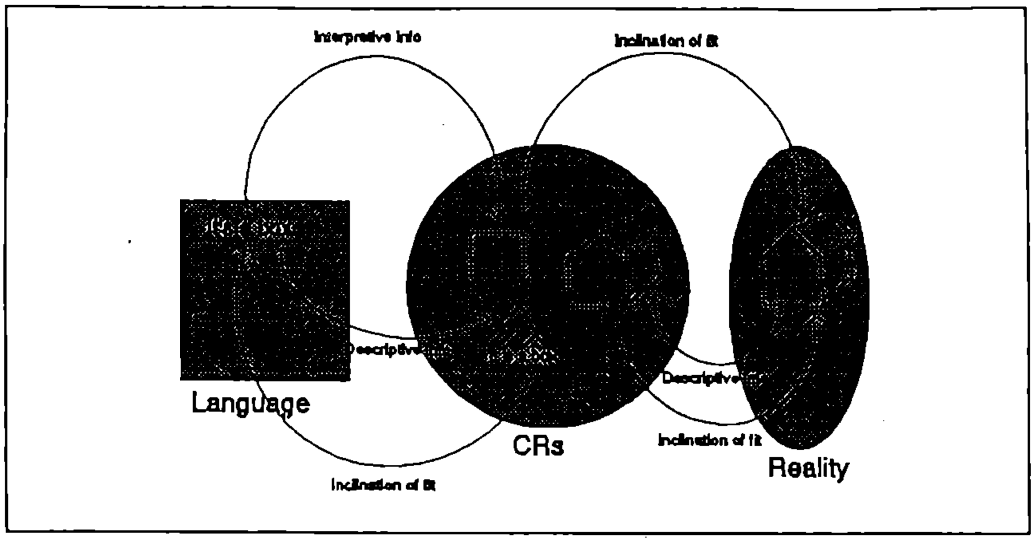


FIG 1 : The effect of descriptive and interpretive information, and *Inclination of Fit*

What is the box a representation of? It is standardly argued, I think, that the box is a representation of the meaning of the sentence *it's a box*. This argument is based on the assumption that lexical items and sentences *contain* meaning, and that this meaning can be independently *represented*. However, nothing so far has proved this assumption either useful or necessary for the purposes of language *understanding*. It is a purely descriptive view. For the purpose of language understanding it is much more fruitful to adopt the view that linguistic items have *semantic effects*. And that semantic effects have consequences for the creation and manipulation of *computable structures*. So,

- the box is not a representation *of* anything, but rather a computable structure with representational *potential*, created on the basis of information made explicit by the meaning of the sentence *it's a box*.

It is different from the other two in not being a *representation* of its source. It is similar to the others in being a structure with *inclination of fit*. I call it *target-inclined*, for it has inclination of fit towards a target which is different from its source. If it *has* a target, then it *becomes* a representation. It is created on the basis of *interpretive* information.

In general, the information that language carries in virtue of meaning is interpretive.

Semantic effect

So, for the purposes of language understanding, linguistic items do not *contain* meaning, they have *semantic effects*. Replacing the notion of semantic content by the notion of semantic effect need not force us to abandon the key principle of (formal) semantics, the principle of compositionality. We can reformulate it as the

Principle of uniformity of semantic effect

> Whatever semantic effect an expression has in one composite expression, it has the same semantic effect in another composite expression.

Pursuit of this principle has some interesting consequences. Firstly, the explanation of specificness and genericness in English, for example, cannot be upheld in its usual form, which in fact assigns two different semantic effects to the articles. Secondly, lexical (or descriptive) meaning is not subject to the principle. The assignment of a certain semantic effect to *bank*, for example, concerns its status as a noun or a verb, not its status as a homonym. The property of having a certain semantic effect is a matter of grammatical, or structural, meaning. It thus makes sense to inquire into, for example, the semantic effects of NP as a structural entity.

Semantic effects of NP

NP contains information that enables us to

• *individuate* entities	semantic effect of D
• *enumerate* entities	semantic effect of Q
• *classify* entities	semantic effect of N
• *assign properties* to entities	semantic effect of A
• *compare* entities	semantic effect of A
• *identify* entities	semantic effect of NP

In accord with Devlin (1991,20f;25), individuation presupposes a basic cognitive capacity to discriminate. Enumeration is a matter of recursive individuation. Classification is a function of individuation and our general cognitive capacity to categorize entities – ie. to realize that two distinct entities may be the 'same' in some respect. Property assignment is a function of individuation and our general cognitive capacity to localize entities – ie. to realize that the same entity may be in different places at different times. Subclassification and comparison are matters of recursive classification and property assignment, respectively. Finally, identification is a function of either classification or property assignment or both. Figure 2 illustrates these principles.

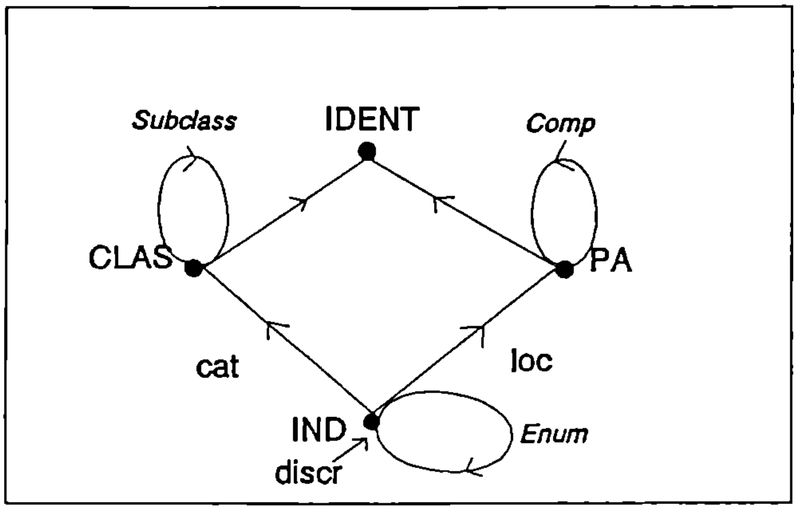


FIG 2 : The semantic effects of NP

This kind of semantic motivation is utterly deplorable for *descriptive* purposes. Yet for *functional* purposes it has two advantages:

- we can give a principled subclassification of (Danish) specifiers
- we can give a general layout of the organization of Danish NP in which the question of hierarchical structure is relegated to secondary importance – perhaps to be accounted for by lexical dependency rules – in deference to the question of linear order, which is far more important for language understanding.

Exhaustive		**Selective**			
UQ	D	EQ	AltD	A	N
alle	**de**	**mange**	**andre**	**store**	**drenge**
al	disse	få	øvrige		
alt	denne	ene	yderligere		
hele	dette	eneste	næste		
begge	den	to	første		
	det	tre	sidste		
	min-	nogen	anden		
	din-	noget	tredje		
	...	nogle	...		
	's	ingen			
		intet			
hver		en	**større**		
enhver		et			
ethvert					
hvilken		**største**			
hvilket					
hvilke					
hvaffor					

Notice that some of the otherwise distinct effects are neutralized in some cases. The interrogatives and distributives (*hv-*) neutralize the quantifier - determiner effect. They are just exhaustive, in the sense of instructing the listener to take everything in the universe of discourse which meets the conditions posed by whatever lexical material follows in the NP into account.

Conclusion

I was asked after delivering the present paper what it had to do with computational linguistics. Granted, if the term 'computational linguistics' is reserved for the automatic manipulation of strings in various ways – not a lot. But if it is taken as a term for those varied branches of study that converge on the common goal of "produc[ing] a comprehensive, computational theory of language understanding and production that is well-defined and linguistically motivated" (Allen 1987,2), then – quite a lot. Among the consequences for computational linguistics of the position defended above the following are of especial interest:

- *Rethinking of the nature of 'rules'.* PS-rules may be an efficient and elegant means of capturing the structural properties of sentences. Yet if what we are interested in is not primarily structural properties, but the effect of structural information on computable structures, then they may not be efficient. Perhaps production rules, embellished with instructions for actions, would be a better choice. Cf. Thrane (fc) and Dinsmore (1991).

- *Rejection of correspondence theory as the basis of semantics.* Whether a sentence is true or not is a question of whether the computable structure it gives rise to has inclination of fit towards a factual situation or not. This question is clearly of secondary importance to the primary question of how computable structures are created and maintained in the first place. If the information needed for these procedures emerges from various aspects of NL meaning, then equally clearly these aspects of meaning must take analytic precedence over other semantic matters.

- *Parsing vs. model construction.* Parsing as currently practised is an inherently descriptive endeavour. The product of a successful parse is a set of source-inclined trees that reveal structural properties of NL sentences. However, parsing is a complex procedure which subsumes recognition of input and production of output, and there is nothing to prevent us from writing a parser that will yield a different, target-inclined kind of output structure. Nothing, that is, except the problems of identifying and formalizing the features that constitute the 'situatedness' of natural language. This would entail, among other things, taking a procedural view of the meaning of specifiers, instead of just recording it and using it for grammaticality checks, as is usually done. Consider in this connection the following remark by Bolter (1984, 125) [my italics]:

> When humans speak to their robots or electronic brains, they do so in something approximating English, often omitting articles and other small words to suggest the computer's preference for reducing language to the bare bones of logic.

This is just utter nonsense in the present context. The implicit belief that 'the bare bones of logic' are embedded in lexical meaning has nothing to recommend it, even under standard assumptions about quantification in natural language and logic. Under present assumptions, withholding from 'our robots and electronic brains' the information provided by 'articles and other small words' is tantamount to preventing them from even beginning to understand what we are talking to them about.

References

Allen, James. 1987. *Natural Language Understanding*. Benjamin, Menlo Park.

Bartsch, Renate. 1973. *The Semantics and Syntax of Number an Numbers*. In Kimball, John P. (ed) *Syntax and Semantics*, Vol. 2. Academic Press, New York. Pp. 51–94.

Bolter, David. 1984. *Turing's Man*. Penguin, Harmondsworth.

Chomsky, Noam. 1970. *Remarks on nominalization*. In *Studies on Semantics in Generative Grammar*. The Hague. 1972.

Chomsky, Noam. 1981. *Principles and Parameters in Syntactic Theory*. In Hornstein & Lightfoot (eds.) *Explanation in Linguistics*. Longman, London. Pp. 32–75.

Devlin, Keith. 1991. *Logic and information*. CUP, Cambridge.

Dinsmore, John. 1991. *Partitioned Representations*. STUDIES IN COGNITIVE SYSTEMS, Vol. 8. Kluwer, Dordrecht.

Haegeman, Liliane. 1991. *Introduction to Government and Binding Theory*. Blackwell, Oxford..

Jackendoff, Ray S. 1977. *X-Bar Syntax*. MIT, Cambr. Mass.

Stuurman, Frits. 1985. *Phrase Structure Theory in Generative Grammar*. Foris, Dordrecht.

Thrane, Torben. 1992a. *The fallacy of descriptivism*. In Hansen, S.L. & F. Sørensen (eds.) *Issues in Semantic Representation*. Samfundslitteratur, Copenhagen.

Thrane, Torben. 1992b. *Dynamic Text Comprehension*. In Jansen, Steen et al. (eds.) *Computational Approaches to Text Understanding*. Museum Tusculanum, Copenhagen. Pp.173–90.

Thrane, Torben. 1993. *The Computations of Text Comprehension*. In Ahrenberg, Lars (ed.) *Papers from the 3rd NOTEX Conference on Text Comprehension in Man and Machines*. Linköping 1993. Pp. 173–82.

Thrane, Torben. forthcoming. *NP-structure and computation*. To appear in Herslund, M. (ed.) COPENHAGEN STUDIES IN LANGUAGE, Vol. 16. Samfundslitteratur, Copenhagen.

Wexler, K. and P. Culicover. 1980. *Formal Principles of Language Acquisition*. MIT, Cambr. Mass.