

A Minimalist Head-Corner Parser

Mettina Veenstra
vakgroep Alfa-informatica, University of Groningen
Postbus 716
NL-9700 AS Groningen
Mettina@let.rug.nl

Abstract

In the Minimalist Program (Chomsky, 1992) it is assumed that there are different types of projections (lexical and functional) and therefore different types of heads. This paper explains why functional heads are not treated as head-corners by the minimalist head-corner parser described here.

1 Introduction

In the Minimalist Program (Chomsky, 1992) 'surface' word order is determined in a very indirect way. Word order is no longer a property of phrase structure, because phrase structure is universal. Furthermore movements are universal. This implies in principle that when we parse comparable sentences in different languages, we always build the same tree. Word order differences are distinguished by the choice of the moment of Spell Out (SO). SO is the point in the derivation where instructions are given to an interface level called PF (Phonetic Form). Thus SO yields what was formerly called surface structure. SO determines in which position in the tree a certain constituent becomes visible and consequently it determines the relative order of the constituents of a sentence. This is illustrated in the simplified tree in figure 1. Note that each cluster of co-indexed positions (i.e. a chain) in the figure has only one visible constituent. This is the position in which the constituent is represented at the moment of SO. This moment is not universal. The verb chain of our English example gives instructions to the interface level PF when the verb is adjoined to AgrS (head of the agreement phrase of the subject). The verb chain of a comparable sentence in Dutch 'spells out' when the verb is in V. Thus in Dutch subordinate clauses the movement of the verb to AgrO (head of the agreement phrase of the object) and

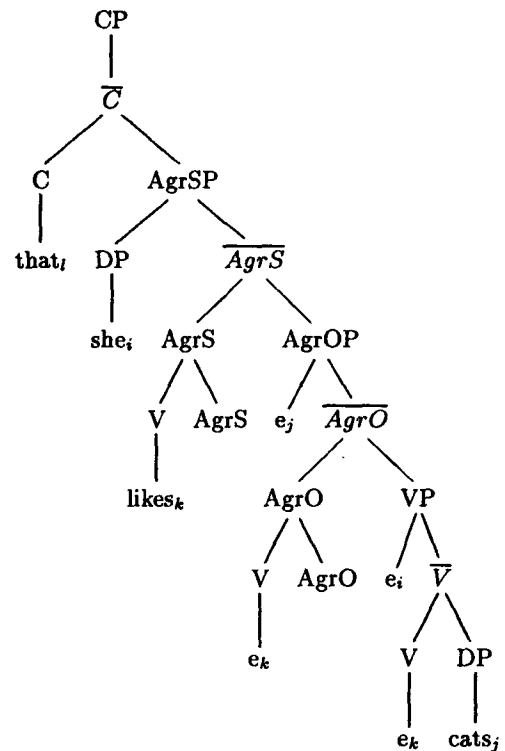


Figure 1: A simplified tree for a transitive subordinate clause in English

subsequently AgrS happens 'covertly'. The motivation for covert movement can be found in (Chomsky, 1992, pages 38–40).

In the following sections we will show that the structure building operations of the Minimalist Program are bidirectional operations. Because head-corner parsing is a bidirectional strategy, this type of parser seems more favorable for minimalist parsing, than the usual left to right parsing algorithms.

2 GT and Move- α

The central operations of the Minimalist Program are Generalized Transformation (GT) and Move- α . GT is a structure-building operation that builds trees in a bottom-up way as is illustrated in figure 2.

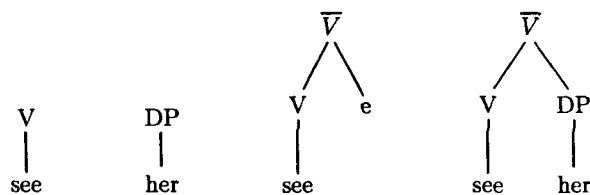


Figure 2: GT applied to V and DP yielding \bar{V} .

Two phrase markers (V and DP) are combined into one. One of these two is called the target (V). A projection of the target (\bar{V}) is added to the target. The projection of the target has two daughters: the target itself and an empty position. The empty position is substituted for by the second phrase marker (DP). This second phrase marker is itself built up in other applications of GT and/or Move- α .

Move- α is a special kind of GT. It is an operation that combines a target with a moved phrase marker. It is assumed that movement is always leftward (Kayne, 1994) and that in the universal trees of the Minimalist Program heads and specifiers, which are the only positions to move to, are always to the left of the projection line. These two assumptions in combination with the fact that GT and Move- α are bottom-up operations, effect that the moved phrase marker has to be contained in the tree that was built so far ¹.

The tree in figure 1 illustrates different kinds of movement. In the Minimalist Program movement occurs to check features. Elements move from the lexical domain (VP) to the functional domain (e.g. AgrOP, AgrSP) to compare their features with the features that are present in the functional domain.

3 Head-corner parsing

The main idea behind head-driven parsing (Kay, 1989) is that the lexical entries functioning as heads contain valuable information for the parsing process. For example, if a verb is intransitive it will not require a complement, if it is transitive it will require a complement. Therefore the head is parsed before its sisters in a head-driven parser. A head-corner parser (Kay, 1989; Bouma and van Noord, 1993) is a special type of head-driven parser. Its main characteristic is that it does not work from left to right but instead works bidirectionally. That is, first a potential head of a phrase is located and next the sisters of the head are parsed. The head can be in any position in the string and its sisters can either be to the right or to the left.

A head-corner parser starts the parsing process with a prediction step. This step is completed when

a lexical head is found that is the head-corner of the goal (i.e. the type of constituent that is parsed). The head-corner relation is the reflexive and transitive closure of the head relation. A is the head of B if there is a rule with B as left hand side (LHS) and A as the head daughter on the right hand side (RHS). When a (lexical) head-corner is found an \bar{X} rule is selected in which the (lexical) head is on the RHS. The sisters of the head are parsed recursively. The LHS of the rule contains the mother of the head. If this mother is a head-corner of the goal, and the mother and the goal are not equal the whole process is repeated by selecting a rule with the new head-corner (i.e. the mother of the first head-corner) on its RHS.

In section 2 it is assumed that movement is invariably leftward and that GT and Move- α are bottom-up mechanisms. GT builds the VP before other projections. Constituents of VP are moved to higher projections by Move- α , which is a special kind of GT. Suppose that the parser should consider AgrS as the head-corner of AgrSP, which accords with \bar{X} -Theory. Then the head (AgrS) that should be filled with an adjoined verb by movement from AgrO (in a transitive sentence) or V (in an intransitive sentence) is created before AgrO and V. To avoid moving constituents from a part of the tree that has not been built yet, the head-corner table for the minimalist head-corner parser is not constructed completely according to \bar{X} -Theory (see (1)).

- (1) $hc(\overline{AgrS}, \overline{AgrSP})$. $hc(\bar{V}, VP)$.
 $hc(\overline{AgrOP}, \overline{AgrS})$. $hc(V, \bar{V})$.
 $hc(\overline{AgrO}, \overline{AgrOP})$. $hc(\bar{N}, NP)$.
 $hc(VP, \overline{AgrO})$. $hc(N, \bar{N})$.

For example, instead of AgrO, VP is the head-corner of \overline{AgrO} . This solution is compatible with the Minimalist Program in the sense that in this way the tree is built up in an absolute bottom-up way (i.e. starting from V) so that a position that should be filled by movement is always created after the position from which the moved element comes. The head-corner table in (1) illustrates that functional heads like AgrO and AgrS are not processed as heads. Lexical projections like VP and NP are treated according to \bar{X} -Theory. If we follow (1) in combination with the tree in figure 1 we establish the fact that the parser searches its way down to the verb as soon as possible. The top-down prediction step moves from the goal AgrSP to \overline{AgrS} to AgrOP to \overline{AgrO} to VP to \bar{V} and finally to the lexical head-corner V where the bottom-up process starts as the Minimalist Program requires.

The head-corner parsing algorithm and the

¹See (Veenstra, 1994) for further details.

structure-building operations of the Minimalist Program (GT and Move- α) have much in common. In both cases a tree is built up in a bottom-up way by starting with a head (lexical head-corner in the parsing algorithm, target in the structure building operations) and creating the sister of the head recursively, etc.² By treating only lexical heads as head-corners we achieved that our parsing algorithm completely represents GT. Only for Move- α we need an extra predicate to accomplish a movement if there is a possible movement to the node that has just been created.

4 Parsing vs. Generation

In section 3 we chose not to consider functional heads as head-corners. This choice was made because it allows GT and Move- α to start constructing a VP before the projections to which constituents from VP are moved are constructed. Another motivation to start with VP is that V contains information that is useful for the remainder of the structure building process. For example, if the verb is intransitive we know that V does not require a complement sister, and we know that we do not need an AgrOP on top of VP. The fact that V contains lexical information and functional heads like AgrO and AgrS do not, could be used as a justification for the fact that the latter are not head-corners. The main idea of head-driven parsing is, as was stated before, that heads contain relevant information for the parsing process, and that they therefore should be parsed before their sisters. Functional heads obtain their contents via movement of elements from positions lower in the tree. This special status makes them less useful for the parsing process.

The Minimalist Program is a generation-oriented framework. Because we are dealing with parsing (as opposed to generation) in this paper there are certain discrepancies between the parser and the framework it is based on. In the minimalist framework, lexical information belonging to a chain is available from the moment that the first position of the chain is created, because that is the moment when the lexicon is consulted. When parsing a sentence the lexicon is not by definition consulted at the beginning of the chain. Figure 1 shows a tree that contains traces and visible constituents. The position containing a visible constituent is the SO position of that chain. The parser consults the lexicon at the moment in which the SO position of a chain is reached. Conse-

²In the minimalist head-corner parser that is described here a head always has only one sister because minimalist trees are at most binary branching.

quently, when a trace is created before SO, the features belonging to that trace are unknown. The features of the traces of a certain chain are known as soon as the SO position is reached, because all positions in a chain are linked.

It can be concluded that the absolute bottom-up approach for the building of trees is more useful for generation than for parsing. In generation, lexical information can be used as soon as a position that is the beginning of a chain is created. In parsing we will have to wait until the SO position is reached. In spite of this, we chose not to consider functional heads as heads in order to accomplish an absolute bottom-up process. The reason for this is that, as was mentioned before, otherwise we would be reasoning backwards with relation to movement. This could be inefficient and it is too far removed from the ideas of the minimalist framework.

5 Future Plans

The parser described here can judge the grammaticality of simple declarative transitive and intransitive sentences, and of subordinate clauses. We will extend the parser in such a way that it will cover more advanced linguistic phenomena like anaphors and wh-questions. Furthermore other types of parsers will be built to determine if this 'lexical' head-corner parser is indeed more efficient.

6 Acknowledgements

I would like to thank Gosse Bouma, John Nerbonne, Gertjan van Noord and Jan-Wouter Zwart for their helpful comments on earlier versions of this paper.

References

- Gosse Bouma and Gertjan van Noord. 1993. Head-driven parsing for lexicalist grammars: Experimental results. In *6th Meeting of the European chapter of the Association for Computational Linguistics, Utrecht*.
- Noam Chomsky. 1992. A minimalist program for linguistic theory. MIT Occasional Papers in Linguistics.
- Martin Kay. 1989. Head driven parsing. In *Proceedings of Workshop on Parsing Technologies, Pittsburgh*.
- Richard S. Kayne. 1994. The antisymmetry of syntax. MIT Press, Cambridge.
- Mettina J.A. Veenstra. 1994. Towards a formalization of generalized transformation. In H. de Hoop A. de Boer and Henriette de Swart, editors, *Language and Cognition 4, Groningen*.