SEGMENT GRAMMAR: THE NEXT STEP IN THE EVOLUTION OF TAG?

Gerard Kempen and Koenraad De Smedt
Nijmegen Institute for Cognition research and Information Technology (NICI)
University of Nijmegen
Postbus 9104, 6500 HE Nijmegen, The Netherlands
E-mail: kempen@hnykun52.bitnet, desmedt@hnykun53.bitnet

ABSTRACT

Segment Grammar (SG) is a grammar formalism which is closely related to TAG but operates on smaller increments called syntactic segments. Both use a similar unification operation as a general tree construction mechanism. Whereas TAG is committed to the distinction of syntactic constructions such as Subj-V-Obj, WH(Obj)-Subj-V, to-V(infinitive)-Obj *in the form of separate elementary structures, SG postpones such choices until a later stage. SG is therefore seen as a successor to TAG which computes variants of an elementary structure dynamically during the generation process.*

1 SEGMENT GRAMMAR

Segment Grammar (SG), first proposed by Kempen (1987), is designed to meet the requirements imposed by an incremental generation. It is therefore not surprising that in Segment grammar, like in TAG, syntactic structures are incrementally composed out of smaller units. However, the basic building blocks in SG are usually not trees, but *syntactic segments* which each represent a single ID relation between nodes. SG therefore has a place in the evolution of TAG, in which initial structures were originally required to be clauses, but could later also be rooted in other categories, and allowed non-terminals to appear at leaves, which generally resulted in more and smaller structures. The syntactic segment is simply the smallest possible structure.

A syntactic segment consists of two nodes representing grammatical categories, and an arc representing a grammatical function. Segments are graphically represented in vertical orientation, where the top node is called the *root* and the bottom node the *foot* (see (1c) for some examples). Segments join to form a syntactic structure by means of a local *unification* operation on nodes.

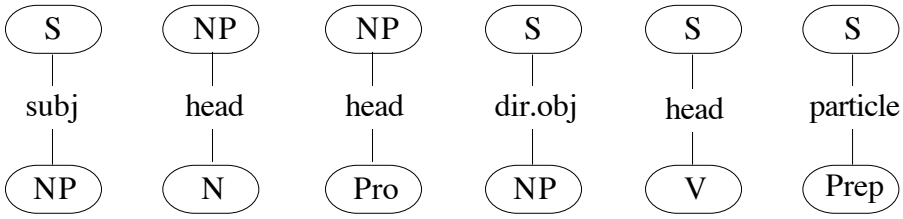
Word order in SG is factored out by distinguishing two subsequent structures—*functional structures* (or *f-structures*) and *constituent structures* (or *c-structures*). F-structures represent grammatical relations between syntactic elements while c-structures represent 'surface' constituency and word order. By way of example, the f-structure (1d) for the Dutch examples (1a) as well as (1b) consists of six segments (1c). The assignment of left-to-right positions to

constituents is modeled as the derivation of a different kind of structure—a *c-structure*. By way of example, c-structure (1e) is assigned to (1a). Left-to-right order of constituents is indicated by means of a number system.

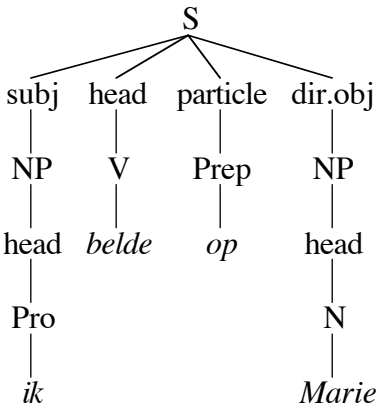
(1) a. Ik belde Marie op. (I called up Mary)

b. Marie belde ik op. (Mary I called up)

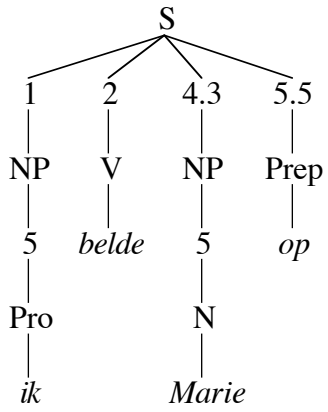
c.



d.



e.



A specification of syntactic agreement can be achieved by having the root and the foot of a segment *share* certain features, whatever their values may be. For example, the feature *nominative* is shared in the NP-head-NOUN segment as depicted in Figure 1.

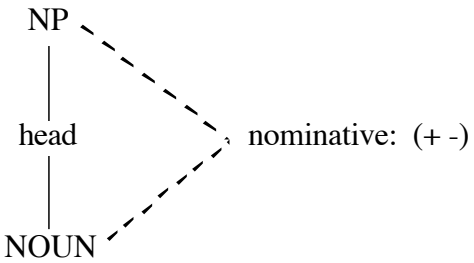


Figure 1: feature sharing

Specifying shared features in a segment thus serves the same purpose as coindexing to indicate structure sharing in other unification-based formalisms. Any subsequent changes to the feature in either the foot or the root will also affect the other. Agreement can be enforced simply by concatenating or furcating several segments with shared features.

SG has been implemented in an object-oriented programming language and is the grammar formalism used in IPF (Incremental Parallel Formulator), a simulation model of some aspects of spontaneous speech (De Smedt, 1990). IPF allows the construction of several parts of the conceptual input in parallel, simultaneously combining several segments

by means of the unification operation. IPF thus distributes the formulation process among several independent processes.

2 A COMPARISON OF SG AND TAG

Tree Adjoining Grammar (TAG; Joshi, 1987) is a tree generating system consisting of a finite set of elementary structures and two composition operations—*substitution* and *adjoining*—which build derived trees out of elementary ones. Like SG, and as opposed to PS-rules, TAG is a tree-based rather than a string-based formalism. Most important, both TAG and SG allow sentences to be incrementally built or only partially built.

Unification as a local operation

FTAG, the recent *feature structures based* extension of TAG (Vijay-Shanker & Joshi 1988) uses unification of features as a clearer way of specifying restrictions on adjoining, and has therefore made TAG even more similar to SG. Both SG and FTAG employ unification as a local operation on nodes. In contrast, other unification-based formalisms such as Functional Unification Grammar (FUG) take seriously the idea of feature structures as the only stores of linguistic information, semantic as well as syntactic, and do not represent grammar rules other than in the form of feature structures. Since the parts of a feature structure operate conjunctively, the grammar must therefore also incorporate the notion of disjunction (or alternation). A FUG is thus a large, disjunctive set of feature structures. SG nor FTAG incorporates this notion of disjunction, and unification is therefore a local, non-disjunctive, and in principle also a non-recursive operation.

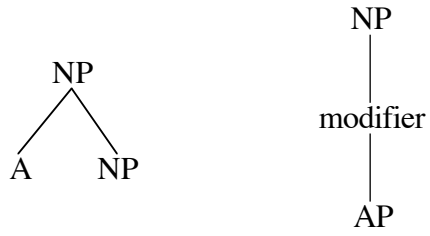
Segments and trees as domains of locality

Both SG and TAG view their basic units, the segment resp. the tree, as separate units in which grammar rules can be fully encapsulated. In SG, e.g, agreement is only specified on the level of individual segments, i.e., as shared features between root and foot. It is not possible to specify feature sharing between sister nodes in SG. Such agreement relations can only be indirectly enforced by specifying several individual sharing relations which are 'chained' over furcated and concatenated segments. The combination of feature sharing and unification thus amounts to 'feature transport', although features are actually unified rather than transported. SG thus *distributes* grammatical constraints among the various syntactic segments which occur in a language. In general, an SG expresses information at the level of the segment, i.e., encapsulated in many separate objects rather than in one large structure or rule base.

Adjoining as a recursive mechanism

The role of some elementary trees in TAG is comparable to that of SG segments, while unification also plays a role in adjoining. For example, the auxiliary tree for an adjective (2a) can be said to correspond to the NP-modifier-AP segment (2b), although it must be noted that SG always creates an adjectival *phrase* rather than just an adjective.

(2) a. b.



A difference between the two approaches is, that TAG allows the *recursive* insertion of A nodes, whereas SG allows the *iterative* addition of AP nodes. In terms of Phrase Structure grammar, the corresponding rules are:

(3) a. NP → A NP (TAG)

b. NP → AP* N (SG)

The adjoining operation of the auxiliary tree for an adjective yields two NP nodes in the resulting structure (Figure 2a), whereas the corresponding composition of SG segments will result in only one NP node (Figure 2b). Whereas it could be argued that the hierarchical embedding of NPs in TAG allows an easier semantic interpretation in some cases (e.g. for the order of non-compositional adjectives: *an old strong man* vs. *a strong old man*) it also makes for considerable redundancy in cases where recursiveness does not affect semantic interpretation.

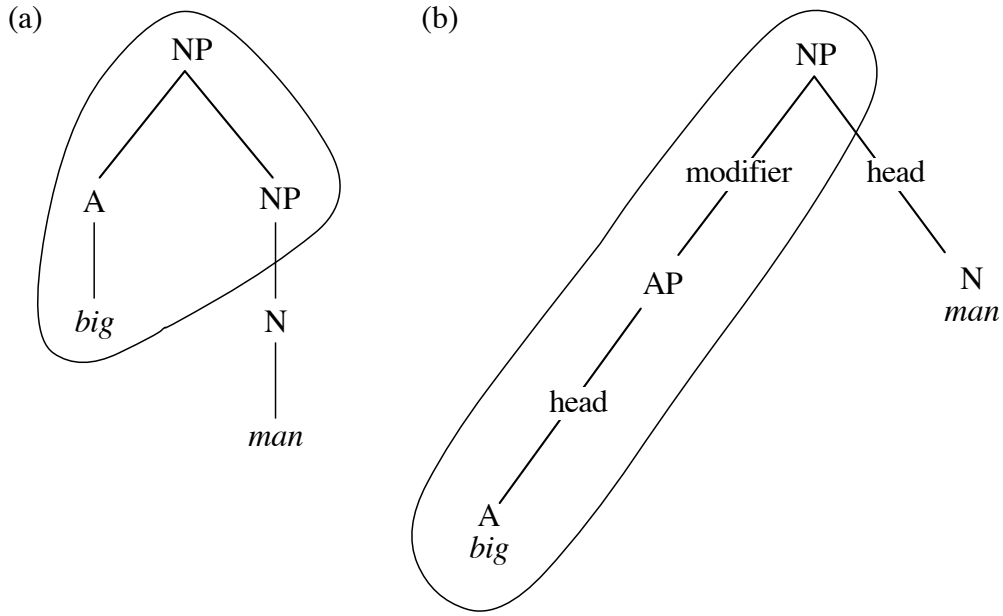


Figure 2: Results of TAG adjoining (left) and SG unification (right)

Growth in horizontal and vertical direction

As is the case in SG, word order and immediate dominance are factored in the TAG formalism (Joshi, 1987), which provides considerable flexibility in the generation process of a sentence. TAG allows incremental generation, but only as defined by the adjoining and substitution operations. Substitution, which has always been present in SG in the form of the concatenation of segments, has been added to TAG only recently (Abeille, 1988). TAG does not allow the addition of sister nodes *(furcation)* without structural growth in vertical

direction as defined by the adjoining operation. Unlike SG structures, TAG trees always contain all sisters. E.g., completely different elementary trees are distinguished for transitive and intransitive verbs. This is problematic for languages like English and Dutch which have many verbs that can be either intransitive or transitive. It does not seem possible to build a transitive tree incrementally by starting with an intransitive tree and expanding it into a transitive one by means of furcation, as SG might allow (if valence permits it). SG allows syntactic structures to expand upward as well as downward, combining structures in any direction by means of unification. On the other hand, SG (as it is currently defined) does not allow the *insertion* of a segment at a node in a previously made structure, because unifications cannot be undone. In contrast, TAG allows insertion by means of adjoining, and upward or downward expansion by means of substitution.

Dynamic choice of elementary trees

TAG seems to require, for many lexical items, a number of variants of elementary trees. For example, the transitive verb *eat* requires separate elementary trees for the constructions *Subj-V-Obj, WH(Obj)-Subj-V, to-V(infinitive)-Obj,* etc. Since the speaker presumably has to choose between such alternatives at an early stage, a TAG-based generation model faces choice problems which are avoided by SG. There, the choice between such constructions can be postponed to a later stage, so that minimal commitments with respect to further incrementation are imposed. In a sense, SG makes the choice between the various constructions dynamically rather than in advance. The meta-rules which relate variant elementary trees in TAG are dynamic rules for the construction of a c-structure in SG. Moreover, the construction of a c-structure can be influenced by separate factors in the speech context, e.g. aspects of timing. E.g., the choice between the Dutch synonymous sentences (4a,b) would require an early choice between separate elementary trees in TAG. In SG, however, these sentences have the same f-structure and only differ in their c-structure.

(4) a. Wie denk je dat daar loopt? (Who do you think walks there?)

b. Je denkt dat wie daar loopt? (You think who walks there?)

Joshi (1987) assumes that choices between variant elementary trees can be made on the basis of pragmatic information provided by the planner in an early stage, even before all descriptions are planned. However, it is unlikely—at least in spontaneous speech—that these choices are 'planned'. Rather, it has been suggested that timing factores may also determine word order choices (Pechmann, 1989). In a model based on SG, which has no predetermined choices on left-to-right ordering, it is easier to represent word order as an emergent factor of timing (De Smedt, 1990).

The verb-final cluster in Dutch

The nature of the sentence-final verb cluster in Dutch suggests that the TAG framework may be too limited to represent certain word order variations. Particles belonging to a Dutch verb, e.g. *weg+gaan* (to go away) may be placed anywhere to the left of the last verb in the cluster. Some examples are given in (5).

(5) a. … dat Jan Piet Marie zag laten *weg* gaan.

b. … dat Jan Piet Marie zag *weg* laten gaan.

c. … dat Jan Piet Marie *weg* zag laten gaan.

An analysis of these sentences in terms of adjoining, as is done by Harbusch (1990), does not seem to be able to accommodate these possibilities, because the particle cannot escape from its elementary structure, unless perhaps by a very *ad hoc* mechanism. In contrast, SG can account for these orderings by not representing the ultimate structure of these sentence as the result of adjoining, but rather flattening the structure so that all verbs and their particles collocate in one surface slot (De Smedt & Kempen, 1990). The internal ordering of this slot is determined by separate principles, which may be language-dependent; e.g., the German ordering of verbs is the exact reverse of the Dutch one.

3 DISCUSSION AND CONCLUSION

Several recent advances in natural language processing and the development of grammar formalisms are characterized by a growing concern for psycholinguistic aspects of the understanding and generation processes. Once it is realized that speech is produced in a temporally defined context, matters of time and memory become as important in the shaping of an utterance as purely structural aspects of language. We have compared and evaluated SG and TAG in the light of incremental sentence generation (Kempen & Hoenkamp, 1987; De Smedt & Kempen, 1987).

Given that speakers can compose an utterance in a piecemeal fashion from conceptual messages which are accessible at different moments, we must avoid grammar formalisms which define the construction of complete sentences only. Kempen (1987) proposes some requirements for a formalism for incremental generation. A first question concerns upward and downward expansion: *How can a syntactic structure be constructed from the bottom of the syntactic structure upward as well as from the top downward?* A formalism which generates sentences by making syntactic choices in a fixed (usually top-down) order, such as a rewriting system based on PS rules, or a hierarchy of procedures (Hoenkamp, 1983) is inadequate. A possible solution lies in the adoption of a *unification*-based formalism. Since unification operates irrespective of the orientation of syntactic segments, a structure can be composed from its substructures in any order. Both SG and FTAG meet this requirement, being based on unification as a mechanism for building structures irrespective of their vertical direction.

Having dealt with expansion in the vertical direction, we must also deal with structural growth in the horizontal direction, since many formalisms incorporate all constituents of a nodes at the same time. Therefore, a second question is: *How can sister nodes be incrementally added to an existing phrase in the syntactic structure?* A solution may consist of the specification of individual Immediate Dominance (ID) relations between nodes, by factoring out lexico-syntactic restrictions on sisterhood. SG is more powerful in this respect, allowing sister nodes to be incrementally added without structural growth in vertical direction.

A third requirement for a grammar formalism concerns word order. Given that the order of conceptual inputs does not necessarily correspond to the eventual left-to-right order of corresponding nodes in the resulting utterance, it seems unpractical to work within a formalism which directly attempts to construct an ordered structure. Therefore the following question arises: *How can a grammar incrementally incorporate new nodes in the structure while making separate commitments with respect to their left-to-right ordering?* Both SG and TAG factor out knowledge about Linear Precedence (LP) from ID relations. However, SG is more flexible with respect to timing. It allows word order choices to be made separately, thereby allowing timing factors to have an influence on word order, as is

warranted in a realistic model of human speech. In contrast, word order is a purely structural choice in TAG.

REFERENCES

Abeillé, A. 1988. Parsing French with Tree Adjoining Grammar: some linguistic accounts. In: *Proceedings of Coling-88, Budapest*. ACL.

Abeillé, A. & Schabes, Y. 1989. Parsing idioms in lexicalized TAGs. In: *Proceedings of the 4th Conference of the European Chapter of the ACL,* Manchester (pp. 1-9). ACL.

De Smedt, K. 1990. Incremental sentence generation: a computer model of grammatical encoding. NICI Technical Report 90-01, Nijmegen Institute for Cognition research and Information technology.

De Smedt, K. & Kempen, G. 1987. Incremental sentence production, self-correction and coordination. In: Kempen, G. (ed.) *Natural language generation: new results in Artificial Intelligence, psychology and linguistics* (pp. 365-376). Dordrecht/Boston/Lancaster: Kluwer Academic Publishers.

De Smedt, K. & Kempen, G. 1990. Discontinuous constituency in Segment Grammar. In: *Proceedings of the Symposium on Discontinuous Constituency*, Tilburg (pp. 97-111).

De Smedt, K. & Kempen, G. (forthcoming) Segment Grammar: a formalism for incremental sentence generation. To appear in: Proceedings of the Fourth International Workshop on Natural Language Generation, Santa Catalina island, 17-21 July 1988.

Harbusch, K. 1990. An efficient parsing algorithm for Tree Adjoining Grammars. In: *Proceedings of the 28th Annual Meeting of the ACL* (pp. 284-291).

Joshi, A. 1987. The relevance of tree adjoining grammar to generation. In: Kempen, G. (ed.) *Natural language generation: new results in Artificial Intelligence, psychology and linguistics* (pp. 233-252). Dordrecht/Boston/Lancaster: Kluwer Academic Publishers.

Kempen, G. 1987. A framework for incremental syntactic tree formation. In: *Proceedings of the Tenth IJCAI*, Milan (pp. 655-660). Los Altos: Morgan Kaufmann.

Pechmann, Th. 1989. Incremental speech production and referential overspecification. *Linguistics* 27, 89-110.

Schabes, Y., Abeillé, A. & Joshi, A.K. 1988. Parsing strategies with 'lexicalized' grammars: application to Tree Adjoining Grammars. In: *Proceedings of the 12th International Conference on Computational Linguistics,* Budapest.

Vijay-Shankar, K. & Joshi, A.K. 1985. Some computational properties of Tree Adjoining Grammars. In: *Proceedings of the 23rd Annual Meeting of the ACL,* Chicago. ACL.

Vijay-Shanker, K. & Joshi, A.K. 1988. Feature structures based Tree Adjoining Grammars. In: *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest. ACL.