

A Linguistic and Computational Analysis of the German “Third Construction”*

Owen Rambow

Department of CIS, University of Pennsylvania
Philadelphia, PA 19104, USA
rambow@linc.cis.upenn.edu

1 The Linguistic Data

For German, most transformational linguistic theories such as GB posit center-embedding as the underlying word order of sentences with embedded clauses:

Weil ich [das Fahrrad zu reparieren] versprochen habe
Because I the bike (acc) to repair promised have
Because I promised to repair the bike

However, far more common is a construction in which the entire subordinate clause is extraposed: *Weil ich t_i versprochen habe, [das Fahrrad zu reparieren]_i*. In addition, a third construction is possible, which has been called the “third construction”, in which only the embedded verb, but not its nominal argument has been extraposed: *Weil ich das Fahrrad t_i versprochen habe [zu reparieren]_i*.

A similar construction can also be observed if there are two levels of embedding. In this case, the number of possible word orders increases from 3 to 30, 6 of which are shown in Figure 1. Of the 30 sentences, 7 are clearly ungrammatical (marked “*”), and 3 are extremely marginal, but not “flat out” (marked “?*”). The remaining 20 are acceptable to a greater or lesser degree (marked “ok” or “?”). No attempt has been made in the linguistic or computational literature to account for this full range of data.

2 A Linguistic TAG Analysis

Following [den Besten and Rutten 1989], [Santorini and Kroch 1990] argue that the third construction, rather than being a morphological effect of clause union, is in fact a syntactic phenomenon. The construction derives from two independently motivated syntactic operations, scrambling and (remnant) extraposition. In this work, I have implemented this suggestion in a variant of multi-component TAG (MC-TAG, [Weir 1988]) defined in [Lee 1991], which I will call SI-TAG. In SI-TAG, as in MC-TAG, the elementary structures are sets of trees, which can be initial or auxiliary trees. Contrary to the regular MC-TAG, in SI-TAG the trees can also be adjoined into trees

*This work was supported by the following grants: ARO DAAL 03-89-C-0031; DARPA N00014-90-J-1863; NSF IRI 90-16592; and Ben Franklin 91S.3078C-1. I would like to thank Bob Frank and Aravind Joshi for fruitful discussions relating to this paper.

from the same set (set-internal adjunction). Furthermore, the trees can be annotated with dominance constraints (or “links”), which hold between foot nodes of auxiliary trees and nodes of other trees. These constraints must be met when the tree set is adjoined.

The following SI-TAG accounts for the German data. We have 5 elementary sets: for the two verbs that subcategorize for clauses, *versuchen* ‘to try’ and *versprechen* ‘to promise’, there are two sets each, representing the center-embedded and extraposed versions. For *reparieren* ‘to repair’, there is only one set. Sample sets can be found in Figure 2. The dominance links are shown by dotted lines.

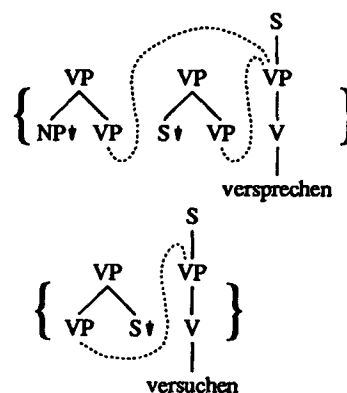


Figure 2: Sample tree sets for *versprechen* ‘to promise’, and *versuchen* ‘to try’ with extraposed subordinate clause

This analysis rules out those sentences that are ungrammatical, since the dominance constraints would be circular and could not be satisfied. Derivations are possible for the sentences that are acceptable. However, the analysis also provides derivations for the three sentences that are extremely marginal, but not ungrammatical. Since these sentences can be derived by a sequence of 3 licit steps, the combination of any two of which is also licit, a syntactic analysis cannot insightfully rule them out. Instead, I would like to explore a processing-based analysis. A processing account holds two promises: first, it should account for the differences in degree among the acceptable sentences; second, it should rule out the extremely marginal sentences.

(i)	Weil ich das Fahrrad zu reparieren zu versuchen versprochen habe	ok
(iv)	Weil ich das Fahrrad zu versuchen zu reparieren versprochen habe	?
(xvi)	Weil ich versprochen habe, zu versuchen, das Fahrrad zu reparieren	ok
(xxiii)	Weil ich zu versuchen versprochen habe, das Fahrrad zu reparieren	?
(xxv)	Weil ich das Fahrrad zu versuchen versprochen habe zu reparieren	?*
(xxvii)	Weil zu versuchen ich das Fahrrad versprochen habe zu reparieren	*

Figure 1: An excerpt from the data

3 A Processing Account Based on Bottom-Up EPDAs

[Joshi 1990] proposes to model human sentence processing with an Embedded Pushdown Automaton (EPDA), the automaton that recognizes tree adjoining languages. He defines the Principle of Partial Interpretation (PPI), which stipulates that structures are only popped from the EPDA when they are a properly integrated predicate-argument structure. Furthermore, it requires that they be popped only when they are either the root clause or they are the immediately embedded clause of the previously popped structure. Before extending this approach to the extraposition cases, I will recast it in terms of a closely related automaton, the Bottom-up EPDA (BEPDA)¹. The BEPDA consists of a finite-state control and of a stack of stacks. There are two types of moves: either an input is read and pushed onto a new stack on top of the stack of stacks, or a fixed number of stacks below and above a designated stack on the stack of stacks is removed and a new symbol is pushed on the top of the designated stack, which is now the top stack (an “unwrap” move). The operation of this automaton will be illustrated on the German center-embedded sentence $N_1N_2N_3V_3V_2V_1$ ². The moves of the BEPDA are shown in Table 3. The three nouns are read in, and each is pushed onto a new stack on top of the stack of stacks (steps 1–3). When V_3 is read, it is combined with its nominal argument and replaces it on the top stack (Step 4). The PPI prevents V_3^{*o} from being popped from the automaton, since V_3^{*o} is not the root clause and V_2 has not yet been popped. V_2 is then read and pushed onto a new stack (Step 5a). In the next move (5b), N_2 , V_3^{*o} and V_2 (i.e., V_2 and its nominal and clausal complements) are unwrapped, and the combined V_2^{*o} is placed on top of the new top stack (the one formerly containing V_3^{*o}). A similar move happens in steps 6a and 6b. Now, V_1^{*o} can be popped from the automaton in accordance with the PPI. (Recall that V_1^{*o} contains its clausal argument, V_2^{*o} , which in turn contains its clausal argument, V_3^{*o} , so that at this point all input has been processed.) In summary, the machine operates as follows: it creates a new top stack for each input it reads, and unwraps

¹I am indebted to Yves Schabes for suggesting the use of the BEPDA.

²I will abbreviate the lexemes so that for example sentence (i) will be represented as $N_1N_3V_3V_2V_1$. As in [Joshi 1990], an asterisk (e.g., V_1^*) denotes a verb not lacking any overt nominal complements. In extension to this notation, a circle (e.g., V_1°) denotes a verb not lacking any clausal complements.

1	[N_1
2	[N_1 [N_2
3	[N_1 [N_2 [N_3
4	[N_1 [N_2 [V_3^{*o}
5a	[N_1 [N_2 [V_3^* [V_2
5b	[N_1 [V_2^{*o}
6a	[N_1 [V_2^{*o} [V_1
6b	[V_1^{*o}

Figure 3: BEPDA moves for $N_1N_2N_3V_3V_2V_1$

whenever and as soon as this is possible.

Using a BEPDA rather than an EPDA has two advantages: first, the data-driven bottom-up automaton represents a more intuitive model of human sentence processing than the top-down automaton; second, the grammar that corresponds to the BEPDA analysis is the TAG grammar proposed independently on linguistic grounds, as shown in Figure 4³. The unwrap in move 5a/b corresponds to the adjunction of tree β_2 to tree α_3 at the root node of α_3 (shown by the arrow), and the unwrap in Move 6a/b to the adjunction of tree β_1 to tree β_2 .

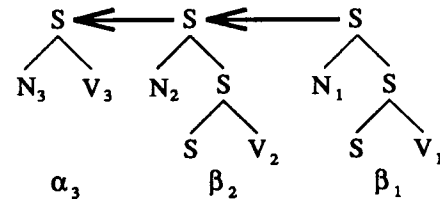


Figure 4: Derivation for German Center-Embedding

Let us consider how the BEPDA account can be extended to the extraposition cases, such as sentence (xxiii), $N_1V_2V_1N_3V_3$. If we simply use the BEPDA for center-embedding described above, we get the sequence of moves in Figure 5. In move 3a, we can unwrap the nominal argument and verb of the matrix clause, which is popped in move 3b in accordance with the PPI. In move 3c, the clause of V_2^* can also be popped. Then, the remaining noun and verb are simply read and popped.

If we use any of the metrics proposed in [Joshi 1990] (such as the sum of the number of moves that input elements are stored in the stack) we predict that sentence

³In the interest of conciseness, VP nodes and empty categories have been omitted.

- 1 $[N_1$
- 2 $[N_1 [V_2^*$
- 3a $[N_1 [V_2^* [V_1$
- 3b $[V_2^* V_1^*$
- 3c $[V_2^*$
- 4 $[N_3$
- 5 $[V_3^*$

Figure 5: BEPDA moves for $N_1 V_2 V_1 N_3 V_3$

(xxiii) is easier to process than sentence (i), which appears to be correct. It is easy to see how this analysis extends to sentence (xvi). Its processing would be predicted to be the easiest possible, and in fact it is the word order by far preferred by German speakers.

Now let us turn to the third construction cases. If we assume the PPI, the only way for a simple TAG to derive the relevant word orders (e.g., $N_1 N_2 V_1 V_2$) is by an analysis corresponding to verb raising as employed in Dutch. In Section 2, I mentioned linguistic evidence against a verb-raising analysis for German. Processing considerations also speak against this approach: we would have to postulate that German speakers can either use the German center-embedding strategy, or the Dutch verb-raising strategy. This would mean that German speakers should be as good at cross-serial dependencies as at center-embedding. However, in German at levels of embedding beyond 2, the center-embedding construction is clearly preferred. We are left with the conclusion that we must go beyond simple TAGs, as was in fact proposed in Section 2. Therefore, a simple BEPDA will not handle such cases either, and we will need an extension of the automaton. This extension will be explained by way of an example, sentence (iv).

N_1 , N_3 , V_2 and V_3 are read in and placed on new top stacks (moves 1 – 4a). (Popping V_2^* would violate the PPI.) Now we unwrap V_2^* and combine it with V_3^* . This yields V_2° : while formerly V_2^* did not lack any nominal arguments (since it has none of its own), V_2° now has its clausal complement, but it is lacking a nominal complement (namely V_3 's)⁴. The reason why N_3 and V_3 can't be unwrapped around V_2 is that V_3 does not subcategorize for a clausal complement. We then unwrap N_3 around V_2° and get $V_2^{*\circ}$ in step 4c. We can then unwrap and pop the matrix clause, and then pop $V_2^{*\circ}$ in the usual manner. The grammar corresponding to the BEPDA of Figure 6 is shown in Figure 7 (the arrows again show the sequence of adjunctions): we see that the deferred incorporation of N_3 corresponds to the use of a tree set for the clause of V_3 .

Finally, let us consider the extremely marginal sentence (xxv), $N_1 N_3 V_2 V_1 V_3$. Here, the automaton as defined so far would simply read in the input elements and push them on separate stacks. At no point can a clause be unwrapped (because both verb/noun pairs are too far apart), and the extension proposed to handle the third construction, the deferred incorporation of nominal arguments, cannot apply,

⁴This operation can be likened to the operation of function composition in a categorial framework.

- 1 $[N_1$
- 2 $[N_1 [N_3$
- 3 $[N_1 [N_3 [V_2^*$
- 4a $[N_1 [N_3 [V_2^* [V_3^{\circ}$
- 4b $[N_1 [N_3 [V_2^{\circ}$
- 4c $[N_1 [V_2^{*\circ}$
- 5 $[V_2^{*\circ}$

Figure 6: BEPDA moves for $N_1 N_3 V_2 V_3 V_1$

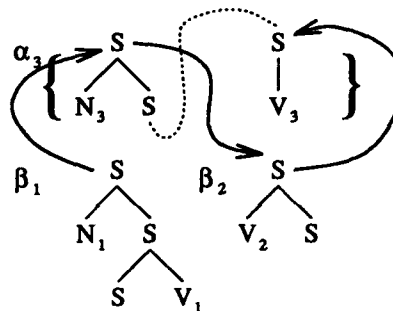


Figure 7: Derivation for $N_1 N_3 V_2 V_3 V_1$

either. The automaton rejects the string, as desired.

4 Current and Future Work

In summary, the linguistic analysis correctly predicts which sentences are ungrammatical, and the processing analysis shows promise for correctly ruling out the extremely marginal sentences, and for accounting for the differences in acceptability among the remaining sentences. Immediate further goals include testing the coverage of this approach, and exploring the relation between the proposed extension to the BEPDA and the form of the SI-TAG grammar.

References

- [Besten and Rutten 1989] Besten, Hans den and Rutten, Jean, 1989. On verb raising, extraposition and free word order in Dutch. In Jaspers, Dany (editor), *Sentential complementation and the lexicon*, pages 41–56. Foris, Dordrecht.
- [Joshi 1990] Joshi, Aravind K., 1990. Processing Crossed and Nested Dependencies: an Automaton Perspective on the Psycholinguistic Results. *Language and Cognitive Processes*.
- [Lee 1991] Lee, Young-Suk, 1991. Scrambling and the Adjoined Argument Hypothesis. Thesis Proposal, University of Pennsylvania.
- [Santorini and Kroch 1990] Santorini, Beatrice and Kroch, Anthony, 1990. Remnant Extraposition in German. Unpublished Paper, University of Pennsylvania.
- [Weir 1988] Weir, David J., 1988. *Characterizing Mildly Context-Sensitive Grammar Formalisms*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania.