An incremental approach to verb clusters in German

Petter Haugereid
Western Norway University of Applied Sciences

Proceedings of the 27th International Conference on
Head-Driven Phrase Structure Grammar

Online (Berlin/Seattle)

Stefan Müller, Anke Holler (Editors)

2020

CSLI Publications

pages 88–105

http://csli-publications.stanford.edu/HPSG/2020

Keywords: HPSG, German, Verb Clusters, Cross-Serial Dependencies, Incremental Parsing

Haugereid, Petter. 2020. An incremental approach to verb clusters in German. In Müller, Stefan, & Holler, Anke (Eds.), *Proceedings of the 27th International Conference on Head-Driven Phrase Structure Grammar, Online (Berlin/Seattle)*, 88–105. Stanford, CA: CSLI Publications.

Abstract

This paper presents an incremental approach to verb clusters in German which radically differs from standard HPSG accounts. While the common assumption is that the verbs in subordinate clauses form clusters and accumulate all their valence requirements on a SUBCAT list, the assumption in this paper is that the arguments in verb final clauses are encapsulated incrementally into syntactic and semantic structures before the verbs are attached. The proposed analysis is in line with psycholinguistic findings. A grammar fragment of German demonstrating an implementation of the analysis is presented.

1 Verb clusters in German HPSG

A widely studied topic in German syntax is that of verbal clusters, as illustrated in (1).

(1) daß ich den Jungen das Buch holen sah
that I the boy the book fetch saw
'that I saw the boy fetch the book'

The clause has an AcI[1] verb *sehen* 'see' which takes an infinitival complement and takes the subject of the infinitival complement as its direct object *den Jungen* 'the boy'. In Müller (2007a) it is given the SUBCAT value shown in Figure 1. The first element on the SUBCAT list is an NP subject (in (1) *ich* 'I'). The last element on the list is an embedded verb (in (1) *holen* 'fetch') which SUBJ and SUBCAT values ($\boxed{1}$ and $\boxed{2}$) also appear on the SUBCAT list of the AcI verb. This ensures that the arguments of the embedded verb (*den Jungen* 'the boy' and *das Buch* 'the book') end up on the subcat frame of the AcI verb.[2]

$$\begin{bmatrix} \textit{AcI-verb} \\ \textsc{cat}|\textsc{subcat}\ \left\langle \text{NP} \right\rangle \oplus \boxed{1} \oplus \boxed{2} \oplus \left\langle \text{V}\begin{bmatrix}\textsc{subj}\ \boxed{1} \\ \textsc{subcat}\ \boxed{2}\end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 1: AcI verb adapted from Müller (2007a, 279)

The schema in Figure 2 shows how complex predicates are combined (Hinrichs and Nakazawa, 1994). In a clause like (1) the AcI verb and the embedded verb are

[†] I would like to thank two anonymous reviewers and the audience at the HPSG 2020 conference in Berlin, Seattle, Buxtehude, wherever, for very useful comments and suggestions. A special thanks goes to Stefan Müller for his constructive feedback. I also would like to thank the research group Language and Society at Western Norway University of Applied Sciences for its valuable support.

[1] Accusative and Infinitive.

[2] Semantic roles and case are also important parts of the account, but that will not be discussed here.

combined. The AcI verb will then be the head daughter. The last element on its SUBCAT list is unified with the SYNSEM of the embedded verb ($\boxed{2}$). The SUBCAT list of the complex predicate ($\boxed{1}$) is the subcat list of the head daughter, except from the last element.

$$\begin{array}{c}\mathit{head\text{-}cluster\text{-}structure} \Rightarrow \\ \begin{bmatrix} \text{SYNSEM} & \begin{bmatrix}\text{LOC}|\text{CAT}|\text{SUBCAT}\ \boxed{1}\end{bmatrix} \\ \text{HEAD-DTR} & \begin{bmatrix}\text{SYNSEM}|\text{LOC}|\text{CAT}|\text{SUBCAT}\ \boxed{1} \oplus \left\langle \boxed{2} \right\rangle\end{bmatrix} \\ \text{NON-HEAD-DTRS} & \left\langle \begin{bmatrix}\text{SYNSEM}\ \boxed{2}\end{bmatrix} \right\rangle \end{bmatrix}\end{array}$$

Figure 2: Schema for complex predicates (from Müller (2007a, 240))

The combination of the transitive verb *holen* ‘fetch’ and the AcI verb *sehen* ‘see’ in example (1) is shown in Figure 3. The SUBCAT list of the mother is the concatenation of the subject of *sehen* ($\boxed{2}$) and the SUBCAT list of *holen* ($\boxed{3}$).

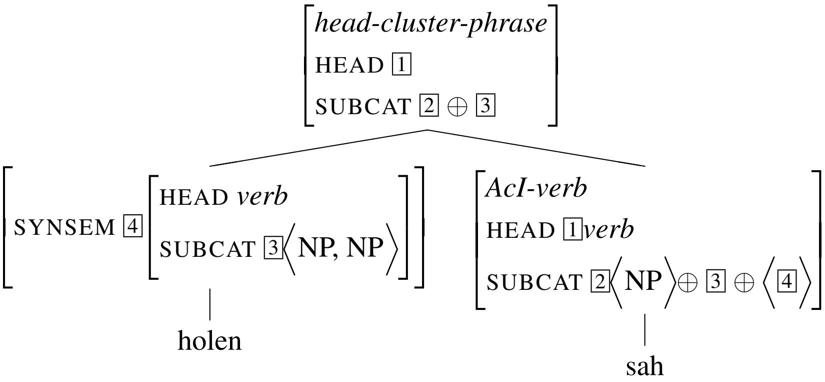

Figure 3: Composition of complex predicate

The arguments are subsequently realized by the Head Argument Schema shown in Figure 4 (Müller, 2007a).

$$\begin{array}{c}\mathit{head\text{-}argument\text{-}phrase} \Rightarrow \\ \begin{bmatrix} \text{CAT}|\text{SUBCAT}\ \boxed{1} \oplus \boxed{3} \\ \text{HEAD-DTR}|\text{CAT}|\text{SUBCAT}\ \boxed{1} \oplus \left\langle \boxed{2} \right\rangle \oplus \boxed{3} \\ \text{NON-HEAD-DTRS}\ \left\langle \boxed{2} \right\rangle \end{bmatrix}\end{array}$$

Figure 4: Head Argument Schema (adapted from Müller (2007a, 79))

This rule attaches the arguments one by one in a binary fashion. The fact that the rule splits the SUBCAT list of the head daughter in three, realizes the middle element ($\boxed{2}$) as the argument, and then concatenates the initial list ($\boxed{1}$) and the final list ($\boxed{3}$) in the SUBCAT of the mother, accounts for the fact that arguments may be permuted. The middle list may contain any of the arguments, since the lengths of list $\boxed{1}$ and $\boxed{3}$ are underspecified.

The HPSG analysis of verb clusters stems from Hinrichs and Nakazawa (1994). While their focus is on the formation of verb clusters and the position of auxiliary verbs, the part of the analysis where the arguments are realized (the Head Argument Schema) is not formalized. Müller (2007a) gives a precise account of the realization of arguments of verb clusters, as illustrated in Figure 4. The use of the concatenation operator in the Head Argument Schema requires arbitrary relational constraints, which are supported by TRALE (Meurers *et al.*, 2002), and not just unification of typed feature structures, which is the case with DELPH-IN resources (Deep Linguistic Processing with HPSG Initiative) like the LKB system (Copestake, 2002).

I would argue that a unification based approach without relational constraints is preferrable to an approach which requires relational constraints of two reasons. The first reason is parsimony. Even though relational constraints allows a grammar writer to write more compact statements, the underlying formalism is more complex and unrestricted. The second reason is the fact that the problems of an approach become more exposed if they are not masked by relational constraints. An example of the latter is the treatment of argument permutations in connection with verb clusters by the German Grammar (Crysmann, 2003), which is implemented with the LKB system, and therefore does not employ relational constraints. It resolves the challenge by assuming different Head Cluster Rules, one for each possible permutation of the arguments. In this way, the argument realization rule does not have to split the SUBCAT list, it just needs to realize the first element. This, however, leads to a large number of combinations of Head Cluster Rules if the number of embedding verbs is larger than one, and it can be said to be a not very elegant brute force approach.

From a processing perspective, there is a second challenge with Hinrichs and Nakazawa's (1994) approach to verb clusters. Given the fact that restrictions on arguments stem from the verb lexical entries, arguments cannot be linked before the verbs have been parsed. The notion of words being incrementally added to an overall syntactic structure one by one (incremental processing) is well established in the psycholinguistic literature, evidenced by studies showing that sentences in head-final languages do not require higher processing than sentences in head-initial languages (Swets *et al.*, 2008). And studies on German show that there is an unmarked order in which arguments are processed (see Kretzschmar *et al.* (2012) and references therein). If an argument is locally ambiguous with regard to nominative or accusative case and it appears first of the arguments, it will typically be interpreted as the subject. If the final verb reveals that it is not the initial argument that is the subject, we get a garden path effect, and the clause will be reanalyzed. This is illustrated in (2) (Kretzschmar *et al.*, 2012).

(2) a. Dass Erich Nachbarinnen stört, hat jeden
that Erich.amb.sg neighbors.amb.pl disturb.sg, has everyone
verwundert.
surprised
'It surprised everyone that Erich disturbs neighbors.'

b. Dass Erich Nachbarinnen stören, ...
that Erich.amb.sg neighbors.amb.pl disturb.acc.pl
'It surprised everyone that neighbors disturb Erich.'

In both (2a) and (2b), the arguments of the subordinate clause are underspecified with regard to nominative or accusative case. In (2a), the verb agrees with the first argument, while in (2b) it agrees with the second argument. Experiments confirm that the marked order results in clearly visible reanalysis costs on the verb. This performance effect is however not explained by the lexicalist approach to verbal clusters in German.[3]

2 An incremental approach to argument realization

In this section I will show how complex predicates with multiple verb embeddings can be analyzed within the framework of Haugereid (2007, 2009).

2.1 Haugereid (2007)

It is a well-known fact that arguments in the German Mittelfeld can permute very freely, and Müller (2006) uses examples from German subordinate clauses (see (3)) to point out problems with the flat structures that are implied by Construction Grammar (Goldberg, 1995). The examples show how the SUBJect, OBJect and OBLique arguments of a clause may be permuted.

(3) a. daß so grün selbst Jan die Tür nicht streicht
that that green even Jan the door not paints
[OBL SUBJ OBJ V]
'that not even Jan would paint the door that green'

b. daß so grün die Tür selbst Jan nicht streicht
that that green the door even Jan not paints
[OBL OBJ SUBJ V]

[3] The argument I am making here is concerning the processing of an utterance. According to Wasow (2020) HPSG theories are theories of competence, and while they should be possible to incorporate into a theory of performance, they are not themselves theories of performance. As I see it, a lexicalist approach like Hinrichs and Nakazawa (1994) does not show how words are assigned structure and meaning incrementally in a theory of performance. This will have to be accounted for in the theory of performance. On the other hand, the left branching approach I am suggesting in this paper, which like other HPSG theories is a theory of competence, would require far less adaption in order to be incorporated into a theory of performance.

c.	daß	Jan	so	grün	selbst	die	Tür	nicht	streicht
	that	Jan	that	green	even	the	door	not	paints
		[SUBJ		OBL		OBJ			V]

d.	daß	eine	solche	Tür	so	grün	niemand	streicht
	that	a	such	door	that	green	nobody	paints
		[OBJ			OBL		SUBJ	V]

‘that nobody paints such a door that green’

In order to account for the clauses in (3) one would need a construction for each possible order of the argument, and if interspersable adjuncts are also to be accounted for, the flat structures becomes unfeasible, given that the number of constructions needed would be infinite.

Haugereid (2007) shows how a constructional approach is still possible if the assumed flat structures are replaced with binary subconstructions. So instead of employing flat structures that realize all the arguments of a clause at once, arguments are assumed to be realized by five types of valence rules; one type of rules for agent or source arguments (CMP1-rules), one type for patient/theme arguments (CMP2-rules), one type for benefactive or recipient arguments (CMP3-rules), one for resultative or end-of-path arguments (CMP4-rules) and one for antecedents (e.g. instrument arguments) (CMP5-rules). These rules may apply before the verb(s) of the clause are attached. In addition to linking the argument to the predicate of the clause, each valence rule contributes an atomic valence type, and during the parse, the valence types are unified with an argument structure type assigned to the verb. When these types are unified, their greatest lower bound is a construction type. If the types do not have a greatest lower bound, the parse fails. This prevents verbs from being assigned arguments that they are not compatible with. It also prevents combinations of arguments that are not licited by the grammar, even though the verb is not yet parsed. This latter fact makes it possible to account for backward gapping in head-final languages like Japanese, where the verb only appears in the final conjunct (Haugereid, 2019).

The rule for attaching a patient/theme argument is shown in Figure 5. It links the complement to the ARG2 of the key relation of the clause KEYREL ($\boxed{1}$). It also introduces a subconstruction type *arg2+* which will be unified with the other subconstruction types and the argument frame type of the predicate.

$$\begin{bmatrix} \textit{cmp2-phrase} & \\ \text{ARGS} & \left\langle \begin{bmatrix} \text{KEYREL} & \begin{bmatrix} \text{PRED } \textit{arg2+} \\ \text{ARG2 } \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{CMP2 } \boxed{2} \end{bmatrix} \end{bmatrix}, \boxed{2}\begin{bmatrix} \text{INDEX } \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 5: Rule for attaching theme/patient (CMP2) arguments

Given the fact that the valence information of a verb is specified by the position of the argument frame type in the type hierarchy of valence types, and not by means of valence lists, the order of the arguments is not fixed in the lexicon. This opens for permutations of arguments in a way that is not possible with a lexicalist approach, as shown in (4). Here, the arguments are realized in a left-branching manner by the valence rules before the verb is attached. The binary design also allows for interspersable adjuncts.

(4) a. [[[[COMPL CMP4] CMP1] CMP2] V]
b. [[[[COMPL CMP4] CMP2] CMP1] V]
c. [[[[COMPL CMP1] CMP4] CMP2] V]
d. [[[[COMPL CMP2] CMP4] CMP1] V]

2.2 Criticism of Haugereid (2007)

Müller (2007b) points out a problem with the approach taken in Haugereid (2007), namely that there will be a need for a new set of valence rules for each embedding verb (raising verbs and control verbs) in a verbal cluster. The rules assumed in Haugereid (2007) only account for the arguments of the matrix verb. The example in (5) has two embedding verbs (*helfen* 'help' and *läßt* 'let'), and an analysis would require three sets of valence rules, linking at different levels of embedding, as illustrated in Figure 6. This number of embeddings would be multiplied by two since each rule has an extraction variant. Müller (2007b) argues that the number of embeddings in verbal clusters is limited by performance, and that a grammar in principle should allow for an unlimited number of embeddings. This would be unfeasible with the N levels deep linking approach inferred from Haugereid (2007).

(5) weil Hans Cecilia John das Nilpferd füttern helfen läßt.
because Hans Cecilia John the hippo feed help let
'because Hans lets Cecilia help John feed the hippo.'

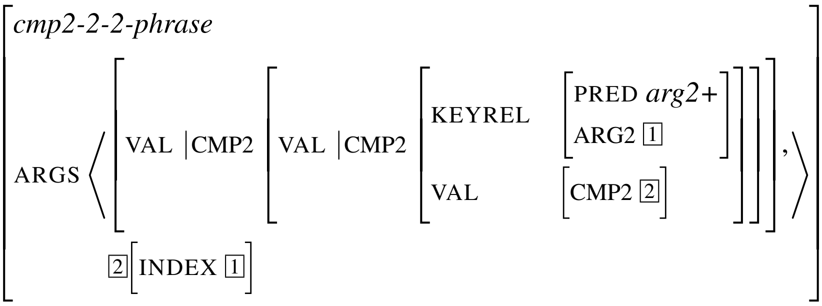

Figure 6: Hypothesized rule for linking theme/patient arguments two levels deep

2.3 Analysis of embedded structures in German subordinate clauses

The problem with the asserted N levels deep linking approach can be solved by means of three unary embedding rules, one for linking the subject of the embedded clause to the subject of the matrix clause (subject raising/control), one for linking the subject of the embedded clause to the indirect object of the matrix clause (object control), and one for linking the subject of the embedded clause to the direct object of the clause (AcI verbs). Figure 7 shows the rule for object control.

$$\left[\begin{array}{l} \textit{unary-obj-control-phrase} \\ \text{SYNSEM} \quad \boxed{1} \left[\begin{array}{ll} \text{SUBJ} & \left[\begin{array}{ll}\text{INDEX} & \boxed{2}\end{array}\right] \\ \text{LTOP} & \boxed{3} \\ \text{INDEX} & \textit{infin}\end{array}\right] \\ \text{STACK} \quad \left\langle \boxed{4} \right\rangle \\ \text{ARGS} \left\langle \left[\begin{array}{l} \text{SYNSEM} \ \boxed{4} \left[\begin{array}{ll} \text{VAL} & \left[\begin{array}{l} \text{ARGFRAME } \textit{arg123} \\ \text{CMP2 } \boxed{1}\end{array}\right] \\ \text{KEYREL} & \left[\begin{array}{l}\text{ARG2 } \boxed{3} \\ \text{ARG3 } \boxed{2}\end{array}\right]\end{array}\right]\end{array}\right]\right\rangle \end{array}\right]$$

Figure 7: Rule for entering embedded structures with object control in German

The rule takes as input a structure, and outputs a structure embedded in the initial structure. The SYNSEM of the input structure is put on a STACK. The rule constrains the argument frame type of the input structure (the matrix clause) to be of type *arg123*, which means that it should have three arguments (an agent, a patient/theme, and a benefactive). The ARG2 of the input structure is linked to the label of the output (the embedded clause). The ARG3 of the input structure is linked to the subject of the embedded clause.

The rule for entering AcI structures is shown in Figure 8. It is similar to the object control rule, except from the fact that it says that the infinitival clause is the CMP4 and not the CMP2, the ARGFRAME value is *arg124*, and not *arg123*, and the matrix structure ARG2 is linked to the subject of the infinitival clause while in the object control rule the matrix structure ARG3 is linked to the subject of the infinitival rule.

Once the embedded structure has been entered, the valence rules can be employed in a regular fashion. There is principally no limit to how many times the unary embedding rule can be used, and so the linking of arguments embedded two levels deep is no longer a problem.

In addition to the unary embedding rules, the grammar also has a unary popping rule, which pops out of embedded structures (see Figure 9).

$$\begin{bmatrix} \textit{AcI-phrase} & \\ \text{SYNSEM} & \boxed{1}\begin{bmatrix}\text{SUBJ} & \begin{bmatrix}\text{INDEX} & \boxed{2}\end{bmatrix} \\ \text{LTOP} & \boxed{3} \\ \text{INDEX} & \textit{infin}\end{bmatrix} \\ \text{STACK} & \left\langle \boxed{4} \right\rangle \\ \text{ARGS} & \left\langle \begin{bmatrix} \text{SYNSEM}\ \boxed{4} \begin{bmatrix} \text{VAL} & \begin{bmatrix}\text{ARGFRAME}\ \textit{arg124} \\ \text{CMP4}\ \boxed{1}\end{bmatrix} \\ \text{KEYREL} & \begin{bmatrix}\text{ARG2}\ \boxed{2} \\ \text{ARG4}\ \boxed{3}\end{bmatrix}\end{bmatrix}\end{bmatrix}\right\rangle \end{bmatrix}$$

Figure 8: Rule for entering embedded AcI structures in German

$$\begin{bmatrix} \textit{popping-rule} & \\ \text{SYNSEM} & \boxed{1} \\ \text{STACK} & \boxed{2} \\ \text{ARGS} & \left\langle \begin{bmatrix}\text{STACK} & \left\langle \boxed{1} \right\rangle \oplus \boxed{2}\end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 9: Rule for exiting embedded structures

The embedding and popping rules works in tandem with the valence rules, creating a left branching tree structure. It is important to note that these left branching structures are not constituent trees, but parse trees. The stacking and popping is a way to navigate the constituent tree. So when an embedding rule works, the parser enters one level of embedding. And when the popping rule works, the parser exits that level of embedding. In this way, linking can be done at various levels during parsing.

The assumed constituent tree structure for the sentence in (5) is shown in Figure 10. The structure is fairly flat, and while this would be a challenge in an approach where the parse tree and the constituent tree is the same, it is not a problem in the present approach given the division between parse trees and constituent trees. (This division is explained in more depth in Haugereid and Morey (2012).)

The tree in Figure 11 shows how the embedding rules and popping rules work during parsing of the sentence in (5).[4] The parse starts in the bottom left corner with the complementizer *weil*. First the subject *Hans*, and the indirect object *Cecilia* are attached (and linked). Then the AcI rule works. It enters the SYNSEM of the AVM parsed so far, onto a STACK in the mother. Now, the second indirect

[4] In Figure 11, linking of the arguments is left out for expository reasons.

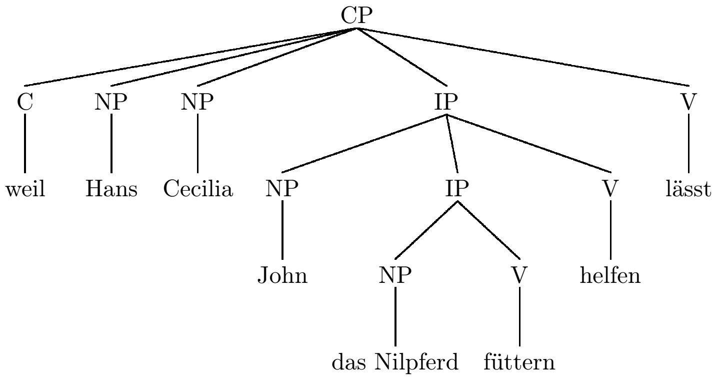

Figure 10: Constituent analysis of German subordinate clause with two embeddings

object *John* is attached (and linked). Note that it is the same type of *cmp3-rule* attaching both *Cecilia* and *John*. No extra valence rule is required, even though the two arguments are at different levels of embedding. The next step is to enter another level of embedding (an object control structure, see Figure 7) before the final argument *das Nilpferd* is attached (and linked). At this point there are two elements on the STACK list, showing the level of embedding. After the arguments are attached, the verbs are attached at the appropriate levels of embedding.[5,6]

The resulting AVM is shown in Figure 12. It shows how the relations of the verbs are linked to their arguments, how the embedded verbs are linked to their matrix verbs (see $\boxed{2}$ and $\boxed{4}$), and how the indirect objects of the control verbs (ARG3) are linked to the subjects of the embedded verbs ($\boxed{3}$ and $\boxed{5}$)

The tree in Figure 13 illustrates that the approach also accounts for permutations. The unary embedding rule works twice in order to allow the object of *füttern* (*das Nilpferd*) to be linked at the correct level of embedding ($\boxed{2}$), before the other arguments. Then two popping rules apply in order to let the subject and the indirect object of the matrix clause be linked (*Hans* and *Cecilia*). Then the embedding rule applies again in order to link the object of *helfen* ($\boxed{1}$). The embedding rule applies over again in order to attach the verb *füttern* at the right level ($\boxed{2}$). The AVM resulting from the analysis in Figure 13 is the same as the AVM resulting from the analysis in Figure 10.

As shown in Figure 13, the embedding and popping mechanism allows the

[5] The left-branching parse trees are, in addition to the incremental nature of the left-branching structures, motivated by the fact that verbs and complementizers in some languages reflect whether they are on the extraction path. In the approach presented in this paper, verbs and complementizers have local access to the extraction path, so the reflection of the extraction path can easily be accounted for. However, in a regular HPSG grammar, this becomes a challenge, especially with regards to extracted adjuncts (Haugereid, 2009, Chapter 6.9).

[6] The approach has similarities with the parsing approach in Güngördü (1997, Chapter 6).

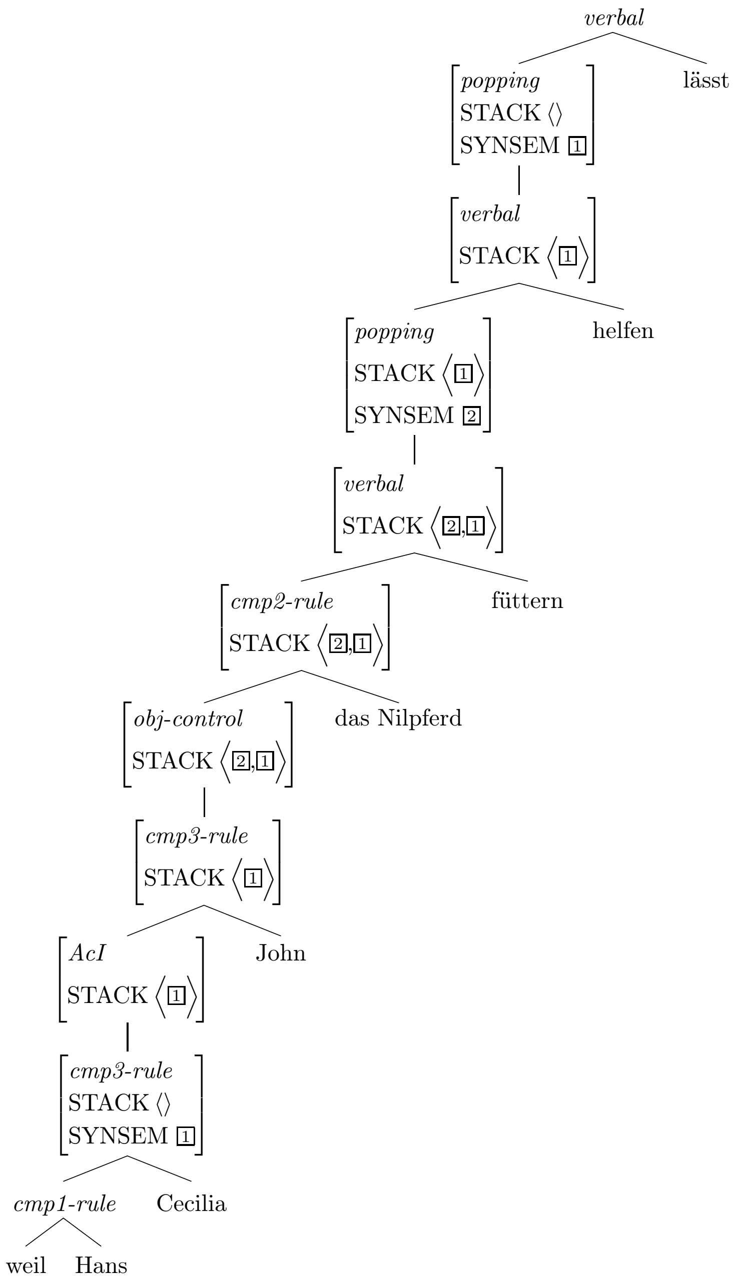

Figure 11: Left-branching analysis of German subordinate clause with two embeddings

parser to enter an embedding, leave it, and then entering it again, adding more specific constraints. The hierarchy of construction types ensures that one is forced down the same embedding if one has exited an embedding and is forced down an embedding again, as illustrated by the tags in Figure 13. So, if the object control

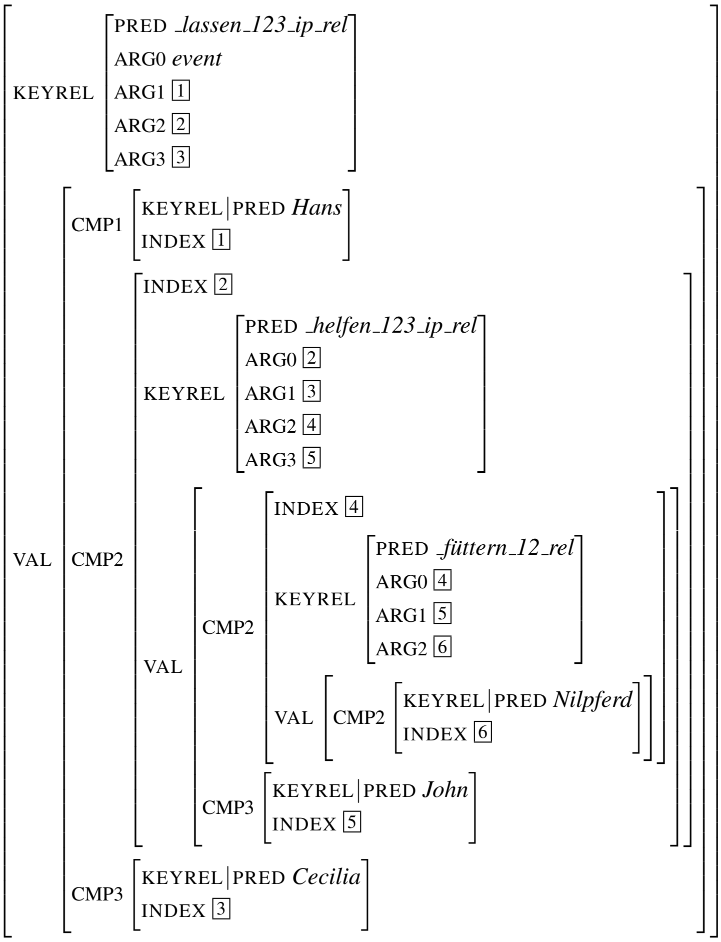

Figure 12: AVM of German clause with two Avl embeddings

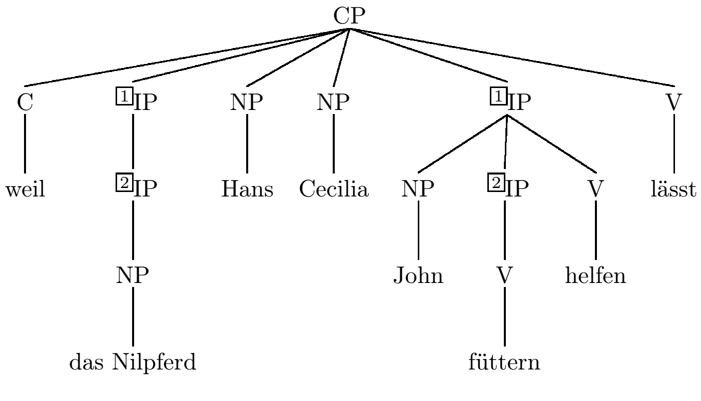

Figure 13: Constituent analysis of German subordinate with two embeddings and permutations

rule in Figure 7 has worked at a certain level earlier in the parse, the same rule will have to work again if there is a need to enter the embedding again. If one tries to employ another embedding rule, like the AcI rule in Figure 8, the ARGFRAME values would not be compatible *arg123* and *arg124*, so the rule would not be applicable.

2.5 Analysis of cross-serial dependencies in Swiss-German

The analysis can also be applied to Swiss-German, which, compared to German, has the verbs in opposite order at the end of the subordinate clause, illustrated in (6) (from Shieber (1985)).

(6)	... mer	d'chind	em Hans	es huus	lönd	hälfe	aastriiche
	we	the children.ACC	Hans.DAT	the house.ACC	let	help	paint

'... we let the children help Hans paint the house.'

In the analysis shown in Figure 14, the arguments of the verbs are attached first, and then the verbs are attached. Since the matrix verb comes before the embedded verbs, the parser pops out to the matrix level before it is attached. Then the parser proceeds to attach verbs at increasing levels of depth. This ensures that the case requirements of the verbs at different depths match the case of their arguments, and the predicates of the verbs are unified with the subconstruction types provided by the subconstructions that attached their arguments. If an argument is attached by a rule with a subconstruction type that is not compatible with the predicate of the verb at that level of embedding, the types will not unify, and the analysis fails.

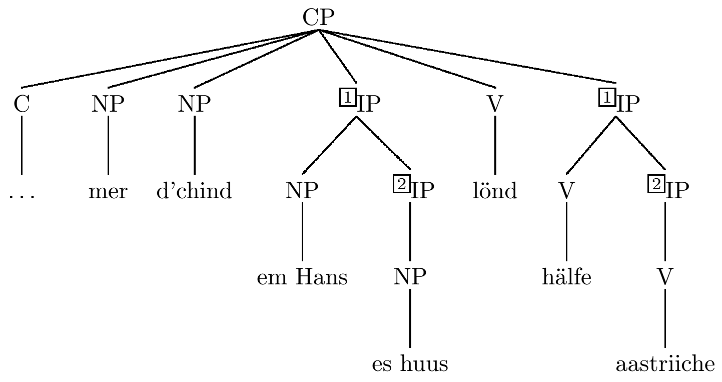

Figure 14: Constituent analysis of Swiss-German subordinate with two embeddings

2.6 Intersective modifier attachment

In addition to permutations of arguments and cross-serial dependencies, the approach also lends itself to an account of intersective modifier attachment at different levels of embedding, before the verbs are attached (see Egg and Lebeth (1995); Crysmann (2004)).

Pütz (1982, 340) shows that the intersective modifier may attach at different levels of embedding in a clause with an embedding verb. In (7) the PP *im Laboratorium* can modify either *blitzen* or *sehen*.

(7) Peter hat es im Laboratorium blitzen sehen.
Peter has it in.the lab flash see
'Peter saw some flashes/lightning in the lab.'

Crysmann (2004, 308) shows that in a subordinate clause, the modifier may permute with the arguments of the verb (see (8)). Still it is just as ambiguous as the sentence in (7).

(8) weil im Labor Peter es blitzen sah
because in.the lab Peter it flash saw
'because Peter saw some flashes/lightning in the lab'

Egg and Lebeth (1995) shows that the sentence in (9) has three readings resulting from different attachments of the modifier *im Märtz*. The modifier can attach to the verb *sollen*, the verb *machen*, as well as the noun *Termin*, even though it is not adjacent to any of them.

(9) Sollen wir im März noch einen Termin ausmachen?
shall we in March an appointment make
'Should we schedule a meeting in March?'

Both Egg and Lebeth (1995) and Crysmann (2004) suggest analyses of the modifiers in (7)–(9) where the semantic attachment is underspecified.

In the present approach it would be possible to link the modifiers directly, during parsing, as shown in Figures 15 and 16.

In Figure 15, the PP is realized under the CP node in the constituent tree, and it therefore modifies the verb *sah*. In Figure 16 on the other hand, the embedding rule is employed before the PP is attached, so that it ends up modifying the verb of the IP, namely *blitzen*. The approach could also account for attachment of adjuncts to nouns, as illustrated in example (9), where one of the readings is that the modifier *im März* modifies the noun *Termin*.[8]

[8] The structure of NPs is not at topic of this paper, and so it is not discussed further here.

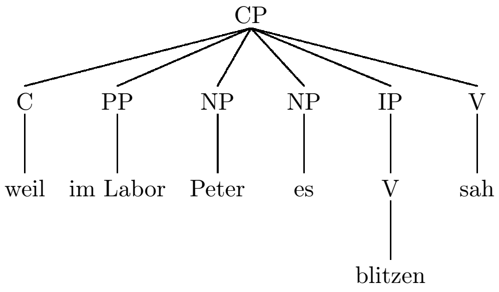

Figure 15: *Im Labor* attaches to *sah*

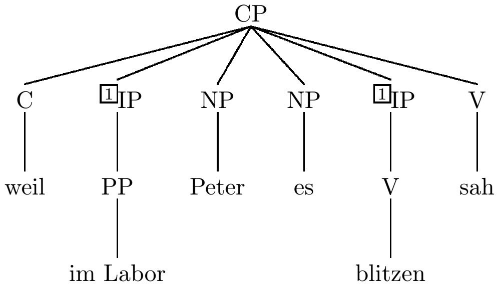

Figure 16: *Im Labor* attaches to *blitzen*

3 Implementation and discussion

The analysis is implemented with the LKB system (Copestake, 2002) in a German demo grammar (Haugereid, 2009, 308-313) based on the Norwegian HPSG grammar Norsyg (Haugereid, 2009). Apart from the lexicon, only slight alterations are made in order to account for the basic clause structures in German.[9] It successfully analyses the examples in (1) and (5) and produces proper semantic representations. The implementation demonstrates that the analysis works, and the grammar analyzes verb-final clauses with multiple embeddings like example (5).

Currently, the implementation only opens for scrambling locally, that is, at the same level of embedding. In order to allow for scrambling between embeddings, allowing for example *das Nilpferd* in (5) to come before the other arguments, as shown in Figure 13, the embedding rules need to be less constrained, that is, they will have to be applicable before all arguments at a level of embedding are realized.[10] This loosening of constraints is not feasible, since the embedding rule then could take itself as input, and the LKB system does not have a way to explore one

[9] In addition to the changes described in Haugereid (2009, 308–310), a unary version is made of the object control rule, and both the object control rule and the AcI rule (which already was unary) were allowed to apply before the verb.

[10] Currently, they are constrained to apply after the arguments at the matrix level are realized.

level of embedding at a time and stop when it arrives (or does not arrive) at an analysis. This would however be an interesting path to pursue, as it would be in line with psycholinguistic findings of garden-path effects, involving backtracking and reanalysis. Whenever the parser has to backtrack from attempting to parse the unmarked order of the arguments of a sentence, the effort on the parser would increase, just like the human processing efforts are increasing when attempting to process a garden path sentence.

The division between a parse tree and a constituent tree demonstrated in this paper allows for linking of arguments during parsing, and it is shown that by retaining a constituent tree, one is able to let the parser enter the same level of embedding more than once, and in this way allow for cross-serial dependencies and modifier attachment at different level of embedding.

References

Copestake, A. (2002). *Implementing Typed Feature Structure Grammars*. CSLI publications.

Crysmann, B. (2003). On the efficient implementation of German verb placement in HPSG. In *Proceedings of RANLP 2003*, pages 112–116, Borovets, Bulgaria.

Crysmann, B. (2004). Underspecification of intersective modifier attachment: Some arguments from german. In S. Müller, editor, *Proceedings of the HPSG-2004 Conference, Center for Computational Linguistics, Katholieke Universiteit Leuven*, pages 378–392. CSLI Publications, Stanford.

Egg, M. and Lebeth, K. (1995). Semantic underspecification and modifier attachment ambiguities. In J. Kilbury and R. Wiese, editors, *Integrative Ansätze in der Computerlinguistik*, pages 19–24, Düsseldorf.

Goldberg, A. (1995). *Constructions: A Construction Grammar Approach to Argument Structure*. The University of Chicago Press.

Güngördü, Z. (1997). *Incremental constraint-based parsing: An efficient approach for head-final languages*. Ph.D. thesis.

Haugereid, P. (2007). Decomposed phrasal constructions. In S. Müller, editor, *The Proceedings of the 14th International Conference on Head-Driven Phrase Structure Grammar*, pages 120–129, Stanford. CSLI Publications.

Haugereid, P. (2009). *Phrasal subconstructions: A constructionalist grammar design, exemplified with Norwegian and English*. Ph.D. thesis, Norwegian University of Science and Technology.

Haugereid, P. (2019). An incremental approach to gapping in japanese. In S. Müller and P. Osenova, editors, *Proceedings of the 26th International Conference on Head-Driven Phrase Structure Grammar, University of Bucharest*, pages 42–57, Stanford, CA. CSLI Publications.

Haugereid, P. and Morey, M. (2012). A left-branching grammar design for incremental parsing. In S. Müller, editor, *Proceedings of the 19th International Conference on Head-Driven Phrase Structure Grammar, Chungnam National University Daejeon*, pages 181–194.

Hinrichs, E. and Nakazawa, T. (1994). Linearizing AUXs in German verbal complexes. In J. Nerbonne, K. Netter, and C. J. Pollard, editors, *German in Head-Driven Phrase Structure Grammar*, number 46 in CSLI Lecture Notes, pages 11–37. CSLI Publications, Stanford University.

Kretzschmar, F., Bornkessel-Schlesewsky, I., Staub, A., Roehm, D., and Schlesewsky, M. (2012). Prominence facilitates ambiguity resolution: On the interaction between referentiality, thematic roles and word order in syntactic reanalysis. In *Case, word order and prominence*, pages 239–271. Springer.

Meurers, D., Penn, G., and Richter, F. (2002). A web-based instructional platform for contraint-based grammar formalisms and parsing. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*, pages 19–26.

Müller, S. (2006). Phrasal or lexical constructions? *Language*, **82**(4), 850–883.

Müller, S. (2007a). *Head-Driven Phrase Structure Grammar: Eine Einführung*. Number 17 in Stauffenburg Einführungen. Stauffenburg Verlag, Tübingen.

Müller, S. (2007b). Phrasal or lexical constructions: Some comments on underspecification of constituent order, compositionality, and control. In S. Müller, editor, *The Proceedings of the 14th International Conference on Head-Driven Phrase Structure Grammar*, pages 373–393, Stanford. CSLI Publications.

Pütz, H. (1982). Objektsprädikate. In W. Abraham, editor, *Satzglieder im Deutschen – Vorschläge zur syntaktischen, semantischen und pragmatischen Fundierung*, number 15 in Studien zur deutschen Grammatik, pages 331–367. Gunter Narr Verlag, Tübingen.

Shieber, S. M. (1985). Evidence against the context-freeness of natural language. In *Philosophy, Language, and Artificial Intelligence*, pages 79–89. Springer.

Swets, B., Desmet, T., Clifton Jr., C., and Ferreira, F. (2008). Underspecification of syntactic ambiguities: Evidence from self-paced reading. *Memory & Cognition*, **36**(1), 201–216.

Wasow, T. (2020). Processing. In S. Müller, A. Abeillé, R. D. Borsley, and J. Koenig, editors, *HeadDriven Phrase Structure Grammar: The handbook (Empirically Oriented Theoretical Morphology and Syntax)*. Language Science Press, Berlin. Prepublished version.