Combining Predicate-Argument Structure and Operator Projection: Clause Structure in Role and Reference Grammar

Laura Kallmeyer
Heinrich Heine University
Düsseldorf, Germany
kallmeyer@phil.hhu.de

Rainer Osswald
Heinrich Heine University
Düsseldorf, Germany
osswald@phil.hhu.de

Abstract

This work presented here is motivated by the goal of formalizing the theory of Role and Reference Grammar (RRG; Van Valin and LaPolla 1997; Van Valin 2005). The main contribution of this paper is to show how RRG's rather flat constituent structure and its operator projection, which reflects the scopal properties of functional operators, can be integrated in a single tree. Inspired by Tree Adjoining Grammar (TAG), we model the operator structure by means of feature structures. Furthermore, we develop an architecture that allows us to impose constraints on sister adjunction, which is the mechanism used for adding operators and modifiers, by appropriate edge and node features.

1 Introduction

Role and Reference Grammar (RRG; Van Valin and LaPolla 1997; Van Valin 2005) is a non-transformational linguistic theory whose development has been strongly inspired by typological concerns and in which semantics and pragmatics play significant roles. One of the basic assumptions of RRG is that clauses have a *layered structure* which reflects the distinction between predicates, arguments, and non-arguments. The *core* (CO) layer consists of the *nucleus* (NUC), which specifies the verb, and its arguments. The *clause* (CL) layer contains the core as well as extracted arguments. Each of the layers can have a *periphery* (PERI) for attaching adjuncts. In the following, we refer to the top clause node and its (non-peripheral) clause, core, and nucleus descendants as the *skeleton* of the clause.

Another important aspect of RRG is the representation of *operators*, which are closed-class grammatical categories such as aspect, modality, and tense. Each type of operator is assumed to attach to a specific layer: tense operators attach to the clause, modality to the core, aspect to the nucleus. Moreover, the surface order of the operators reflects their attachment site in that the higher the layer an operator is attached to, the farther away from the nucleus the operator occurs on the surface.

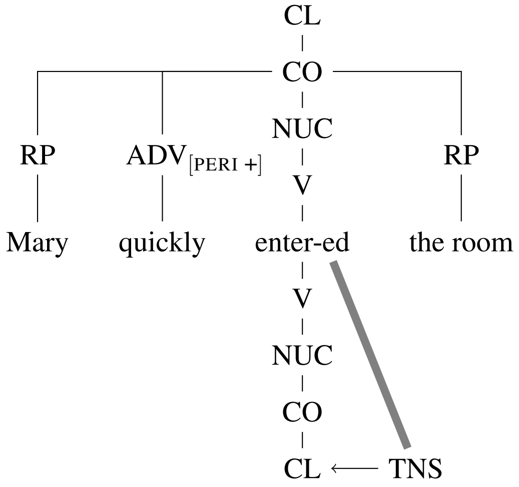


Figure 1: Simplified example of a syntactic representation in RRG with operator projection

While the ordering among the operators is thus systematically correlated with the scope given by their attachment site at the clausal skeleton, the surface order of the operators relative to arguments and adjuncts is much less transparent and often requires crossing branches. For this reason, RRG represents the constituent structure and the operator structure as different *projections* of the clause. The syntactic representation in Fig. 1 illustrates this idea: The tense (TNS) operator, which is a clause-level operator, attaches morphosyntactically to the verb. The example also shows that the peripheral pace adverb *quickly* modifies the core.

Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+13), pages 61–70,
Umeå, Sweden, September 4–6, 2017. © 2017 Association for Computational Linguistics

('RP' stands for 'referential phrase'.)

Johnson (1987) once proposed a formalization of the projection approach which uses two different context free grammars, one for analyzing the sequence consisting of the verb plus arguments and adjuncts, and one for the sequence consisting of the verb plus operators. The two grammars taken together then constitute a "projection grammar". However, Johnson's proposal is purely surface oriented and does not capture the fact that the two projections share basically the same clausal skeleton.

In the present paper, we propose a new approach that conflates the operator projection with the constituent structure, preserves the scope-related ordering constraints of the operators and avoids crossing branches with other constituents. The basic idea is that operators can attach to the clausal skeleton "in situ" and then project their content upwards (or downwards) to their respective scope layer. For instance, a tense operator, whose scope level is the clause, can be attached by sister adjunction to a nuclear node and thereby avoids crossing branches with argument constituents.

The adjunction of the operators needs to be controlled in the following two respects: (i) The adjunction of an operator is *obligatory* if the information conveyed by the operator is required for a sentence to be complete. (ii) The scope-related *ordering* of the operators must be respected. In our approach, these constraints are implemented with the help of feature structures attached to the edges of the trees.

The rest of the paper is organized as follows: Section 2 briefly introduces a proposal for defining tree composition in a way that leads to the kind of flat syntactic representations postulated by RRG. The two operations, (wrapping) substitution and sister adjunction are intended to capture argument structure constructions, including long distance dependencies, and the flat adjunction of modifier expressions. Section 3 presents the core idea of this paper which is to add feature structures to edges (as well as to nodes) for bookkeeping purposes. The approach is then applied in Section 4 for the seamless alignment of the operator projection of RRG with the constituent projection. Section 5 shows how the proposed representation of the operator projection behaves in the case of complex sentences, in which the structure of the clausal skeleton interacts with the scope-taking behavior of the operators in intricate ways.

2 Tree composition in Role and Reference Grammar

RRG shares some fundamental properties with Tree Adjoining Grammar (TAG, Joshi and Schabes, 1997; Abeillé and Rambow, 2000), notably its extended domain of locality and certain underlying assumptions about the structure of elementary syntactic building blocks. In particular, RRG assumes that a predicate and its arguments are realized within the same elementary tree (cf. Frank, 2002, for similar assumptions in TAG). Therefore RRG can be formalized as a tree-rewriting grammar in the spirit of TAG, albeit with slightly different operations for combining elementary trees.

Previous work on formalizing tree composition in RRG (Kallmeyer et al., 2013; Osswald and Kallmeyer, in press) has identified two operations that are needed in order to cope with the flat structure of RRG trees: An operation called *(wrapping) substitution* that serves to fill argument slots, i.e., to fill substitution nodes, and an operation called *sister adjunction* used to add so-called periphery elements, i.e., modifiers.

2.1 (Wrapping) substitution for argument composition

Simple substitution, as in TAG, consists of replacing a non-terminal leaf with a new tree of the same category. The idea behind wrapping substitution is that a substitution node (i.e., a non-terminal leaf) in the target tree gets filled by adding a subtree from a new tree. More concretely, this new tree gets split at a point where the lower part has the category of the substitution slot and can be inserted there. The higher part is identified with the root of the target tree. It can add material above that root but also new material to the right or left of all the daughters of that root. Potential sites for splitting a tree are indicated by dominance links. In other words, wrapping such a tree around another one means stretching the dominance link in such a way that its upper node merges with the root while the lower node merges with a substitution node in the target tree. Such a wrapping substitution occurs for instance in the derivation of the long-distance dependency (1), given in Fig. 2 (where the dashed edge indicates a dominance relation). The result is the tree in Fig. 3. ('PrCS' stands for 'pre-core slot'.)

(1) What does John think Bill smashed?

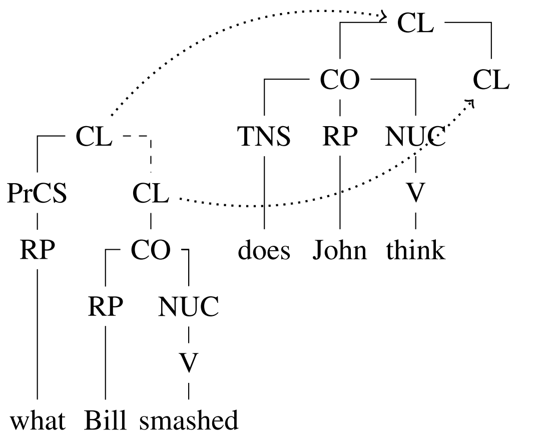


Figure 2: Derivation for (1): wrapping substitution

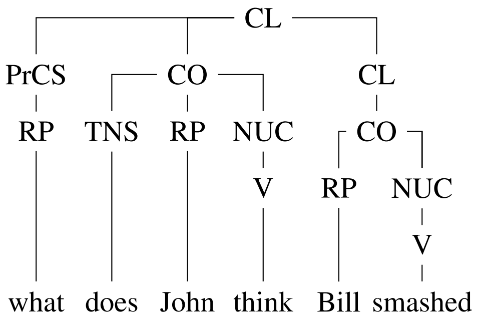


Figure 3: Derived tree for (1)

Note that in a first proposal of how to apply wrapping substitution to tree composition in RRG, Kallmeyer et al. (2013) assumed a more binary structure. The version sketched above goes back to Osswald and Kallmeyer (in press) and is more in line with the flat structures used in RRG. The idea of using wrapping substitution is partly inspired by the operations *subsertion* in D-Tree Grammar (Rambow et al., 1995) and *generalized substitution* in D-Tree Substitution Grammar (Rambow et al., 2001), which, however, are more general. Wrapping substitution shares with subsertion the non-locality: the two nodes targeted by the wrapping substitution (i.e., the substitution node and the root node of the target tree) need not come from the same elementary tree and can be far apart from each other. If the number of wrapping substitutions that stretch across a node in the derived tree is limited by some constant k, it can be shown that an equivalent *simple Context-Free Tree Grammar* (CFTG) (Kanazawa, 2016) of rank k can be constructed, which is in turn equivalent to a well-nested *Linear Context-Free Rewriting System* (LCFRS) (Vijay-Shanker et al., 1987; Seki et al., 1991; Kanazawa, 2009; Gómez-Rodríguez et al., 2010) of fan-out $k+1$ (see Kallmeyer, 2016, for more details on this equivalence).

2.2 Sister adjunction for modification

So-called peripheral elements in RRG are added via sister adjunction. An example is the modifier *quickly* in the example in (2). The corresponding derivation is given in Fig. 4. The root of the modifier tree merges with the target node n of the adjunction and the (necessarily unique) daughter is inserted as a new daughter of n. The categories of the root and n have to be the same.

(2) Mary quickly entered the room

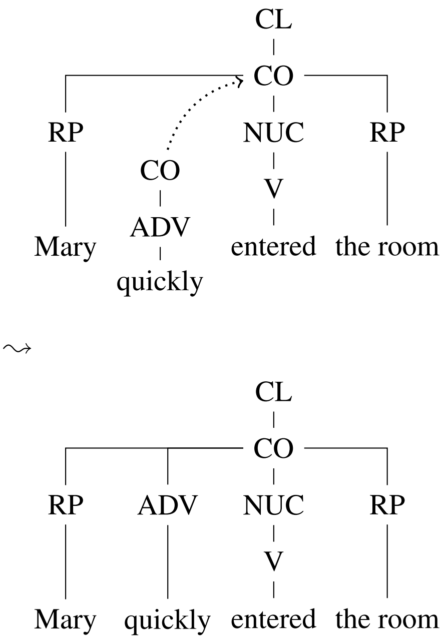


Figure 4: Derivation for (2): modification via sister adjunction

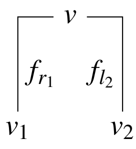
3 Adding feature structures to the syntactic trees

In TAG, internal nodes have top and bottom feature structures. The underlying idea is that the top reflects properties of the node visible from above while the bottom reflects properties of the subtree below the node. These features control the adjunction possibilities at that node. In particular, mismatches between them (i.e., the fact that they do not unify) express an obligatory adjunction constraint.

In RRG, we want to pursue a similar strategy, namely modeling obligatory adjunction via feature mismatches. However, these mismatches cannot be on top and bottom features of the nodes as in TAG. But we can retain the idea that in places where adjunction occurs, two feature structures that are not unifiable get separated. In the case of sister adjunction, we add a new sister between two nodes or, to put it differently, between two edges. Therefore we propose the following for feature-based sister adjunction:

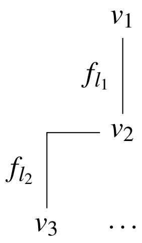
Nodes have just a single feature structure. In contrast, edges have two feature structures, a left one and a right one. In a sister adjunction, the feature structure of the root of the adjoined tree unifies with the feature structure of the node targeted by the adjunction. In the final derived tree, the two feature structures between two neighbouring edges have to unify. (Consequently, if they are not unifiable, this acts as an obligatory adjunction constraint.) Furthermore, features on the leftmost (resp. rightmost) edge percolate upwards, except if there is a substitution node, which blocks feature percolation. More precisely, the following unifications occur in the final derived tree:

1. Whenever there are nodes v, v_1, v_2 with edges $\langle v, v_1 \rangle$ and $\langle v, v_2 \rangle$ such that v_1 immediately precedes v_2, the right feature structure f_{r_1} of $\langle v, v_1 \rangle$ unifies with the left feature structure f_{l_2} of $\langle v, v_2 \rangle$.



$\Rightarrow f_{r_1}$ and f_{l_2} are replaced with $f_{r_1} \sqcup f_{l_2}$

2. Whenever there are nodes v_1, v_2, v_3 with edges $\langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle$ such that v_3 does not have a sister to the left (to the right) and v_2 was not a substitution node, the left (right) feature structure of $\langle v_2, v_3 \rangle$ unifies with the left (right) feature structure of $\langle v_1, v_2 \rangle$.



$\Rightarrow f_{l_1}$ and f_{l_2} are both replaced with $f_{l_1} \sqcup f_{l_2}$ if v_2 was not filled by substitution

For illustration consider the simple example in Fig. 5. The initial tree α carries an obligatory adjunction constraint since the feature structures between the two edges do not unifiy. β can adjoin repeatedly in between and, as a result, we obtain derived trees where the feature structures between neighbouring edges are unifiable.

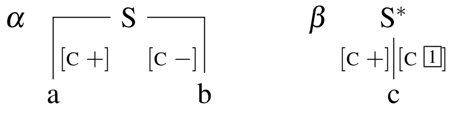


Figure 5: Obligatory sister adjunction

In a substitution, the feature structure at the root of the tree that gets added and the one at the substitution site unify as well. In the more general case of a wrapping substitution with a dominance link from node n_1 to n_2 that gets stretched and a target tree with root n_r and substitution node n_s, the feature structures of n_1 and n_r unify and the ones of n_2 and n_s unify.

The constraint in 2. that v_2 was not a substitution site is motivated by the hypothesis that substitution nodes act as islands concerning operators. An example is (1) where the complement clause is added by wrapping substitution and each of the two clauses requires its own tense operator. This restriction for the feature percolation is a working hypothesis; further examples of (wrapping) substitution need to be examined in order to determine whether this assumption makes the right generalization or whether it is too restrictive.

With the above definition of feature structure unification for edge features, the requirement for a non-finite verbal nucleus to obtain tense from a finite verb can for instance be modeled in a way similar to Fig. 5 via a TNS feature with values $+/-$, as illustrated in Fig. 6. The *sleeping* tree requires that a tense marker adjoins to the core node somewhere to the left of the nucleus, i.e., either preceding or following the subject RP. Besides contributing tense, the finite verb also assigns case to the subject RP and it specifies the agreement features that constrain the subject. This can be modeled via the node features, using features from the XTAG grammar (XTAG Research Group, 2001).

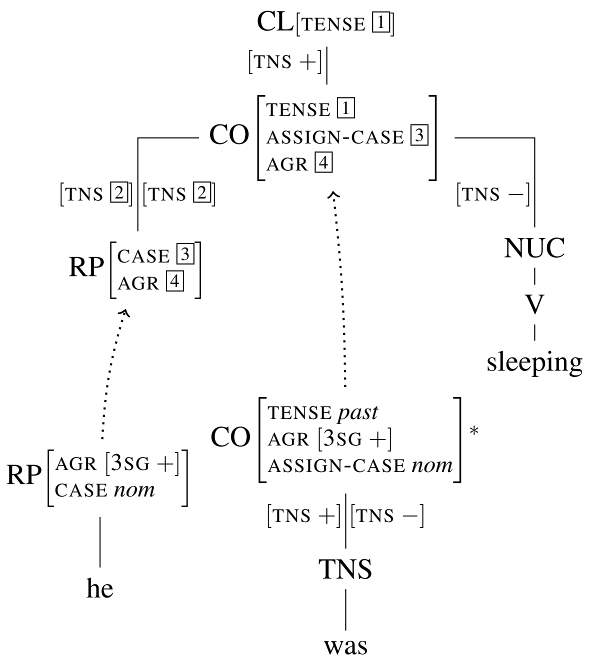

CL[TENSE 1]
[TNS +]
CO [TENSE 1, ASSIGN-CASE 3, AGR 4]
[TNS 2]
[TNS 2]
[TNS −]
RP [CASE 3, AGR 4]
NUC
V
sleeping
CO [TENSE past, AGR [3SG +], ASSIGN-CASE nom] *
RP [AGR [3SG +], CASE nom]
[TNS +]
[TNS −]
TNS
he
was

Figure 6: Obligatory adjunction of a tense marker.

4 Modeling the operator projection with features

We have seen that, in line with Van Valin (2005), we treat operators as modifiers that are added by sister adjunction. Moreover, we have illustrated in Fig. 6 how features can be used to enforce the adjunction of certain operators.

What is missing is a modeling of the operator projection of RRG. Each operator belongs to a certain level of the layered structure (see Fig. 7). The mapping from operators to levels of the layered structure explains (i) the scope behavior of operators, since structurally higher operators take scope over lower ones, and (ii) surface order constraints for operators; higher operators are further away from the nucleus of the structure.

Layer	operators
Nucleus	Aspect
	Negation
	Directionals
Core	Directionals
	Event quantification
	Modality
	Negation
Clause	Status
	Tense
	Evidentials
	Illocutionary Force

Figure 7: Operators in the layered structure of the clause (cf. Van Valin, 2005, p. 9)

The problem is that the constituent and the operator structure are not completely parallel, i.e., one can have structures where an operator belonging to a specific layer is, on the surface structure, surrounded by elements belonging to a lower layer in the constituent structure. Examples are (3) and the Turkish example in (4) (taken from Van Valin, 2005, p.10), where a clause-level tense operator is embedded in the core. In (3), we have the RP *John* and the NUC *sleeping* that form the CO constituent. In between, two operators are added, namely the nucleus-level aspectual operator *been* and the clause-level tense operator *has*. The former can attach at the NUC node, consistent with its operator level. The latter, however, cannot adjoin to the clause node, except if crossing branches are allowed. Within the constituent structure, it is part of the CO constituent while its operator level is higher.

(3) John has$_{\text{TNS}}$ been$_{\text{ASP}}$ sleeping.

(4) Gel-emi-yebil-ir-im.
come-ABLE.NEG$_{\text{MOD}}$-PSBL$_{\text{STA}}$-AOR$_{\text{TNS}}$-1SG
'I may be unable to come'

Similarly, in the Turkish example in (4), the clause-level status and tense operators occur between the verb and the pronominal affix, which is part of the core.

Even though the constituent structure and the operator structure are not fully aligned, they depend on each other. Their hierarchical order is the same and the existence of a layer in the operator projection requires that this layer also exists in the constituent structure. For instance, one can only have clause-level operators if a clause node exists in the constituent structure. In the following we will show that the feature structure-based definition of tree rewriting with sister adjunction proposed above allows us to model the operator projection within the features while attaching the operators at their surface position. In other words, operators sometimes attach lower than their position in the operator structure. The features capture the constraints mentioned above.

Let us illustrate the feature architecture for operators by the analysis of (3) shown in Fig. 8.[1] We

[1] To keep things simple, the analysis does not take into ac-

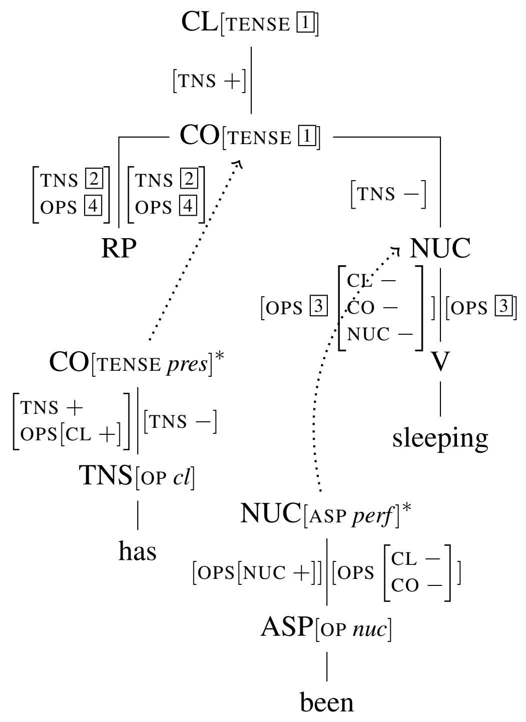


Figure 8: Trees for (3)

assume a feature OPS (for operator structure) used on the edges that specifies which operator projection layer(s) have been reached so far. Its value is a feature structure with features CL, CO and NUC for the three layers, each with possible values + or −. This feature is used in such a way as to guarantee that nuclear, core and clausal operators have to appear in this order when moving outwards in the sentence, starting from the nuclear predicate. For instance, a nuclear operator such as *been* that adjoins to the left of the predicate has a requirement to the right that the levels CL and CO have values −, i.e., are not reached yet. To the left, it just gives the information NUC +. On the other hand, a clausal operator such as the tense operator *has*, when adjoining to the left of the nucleus, does not have any requirement for the operator level that has already been reached (hence there is no OPS feature specified on the right of the top-most edge) but it states to the left of the edge that now CL has value +. In addition to the OPS feature, the preterminal nodes of the operator trees have a feature OP indicating the operator level that the tree targets, which can be different from the root node category of the tree that specifies the constituency level, i.e., the surface position. For example, *has* is a clause-level operator that adjoins at the core node.

The OPS feature can also be used to make sure that operator levels are licensed by corresponding nodes in the constituent structure of the targeted elementary tree. If, for instance, the core layer is the highest layer of the predicative elementary tree, then the OPS features on the left of the leftmost edge and on the right of the rightmost edge immediately below the core node will have features CL −, which means that clausal operators are not allowed within this core.

Fig. 9 shows the derived tree for sentence (3). The final feature unifications between neighbouring edges and between leftmost/rightmost edges below a node and the left feature structure/right feature structure of the edge to the mother lead to the following: The NUC + information is passed from the NUC–ASP edge to the left of the CO–NUC edge and it gets unified with the feature structure on the right of the CO–TNS edge. Furthermore, the feature structures between CO–RP and CO–TNS unify and the resulting values of TNS and OPS are passed to the left of the CO–RP edge and from there to the left of the CL–CO edge.

Note that the root category of the operator tree (in our example CO in the case of *has* and NUC in the case of *been*) determines the attachment site of the operator in the constituent structure. A possible constraint on these elementary trees, which remains to be verified empirically, is that with respect to RRG's layered structure, the root node label of an operator tree has to be lower or equal to the OP value at the operator node (i.e., at the daughter of that root). This means for instance that there cannot be an operator with [OP *nuc*] and root category CO or CL.[2]

Since the preterminal nodes of the operators specify which operator projection layer the operator belongs to, we can deterministically map a derived tree to the standard RRG structure where the constituent structure and the operator projection are separated. The two structures for our example are given in Fig. 10.

Note that a single lexical item can also contribute more than one operator. The operator *had* in (5), for instance, contributes aspect (nuclear

count the progressive aspect of the present perfect progressive construction in (3). A more thorough analysis would derive *perfect* from the auxiliary choice ‘have’ and the past participle morphology of ‘be’, while *progressive* derives from ‘be’ and the ‘-ing’ form of the main verb.

[2] Such constraints can be implemented in a principled way within a metagrammatical representation using for instance XMG (Crabbé et al., 2013).

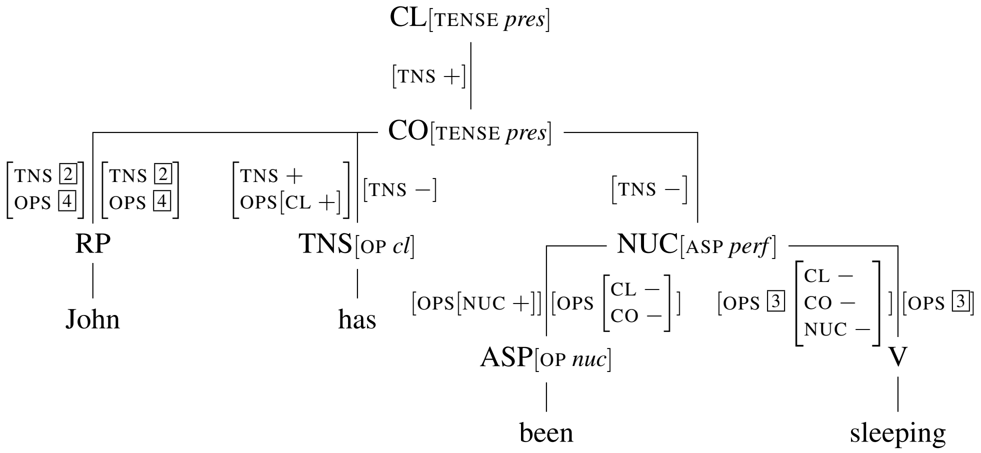


Figure 9: Derived tree for (3) (before final edge feature unification)

level) and tense (clause level).

(5) John had slept.

We therefore modify the representation slightly in that we replace non-terminal operator categories such as TNS, ASP, etc. by a more general category OP, and the single feature OP by a complex feature structure that lists all the features contributed on the different operator layers. In the case of the operator in (5), this would yield the feature structure [NUC [ASP *perf*], CLAUSE [TENSE *past*]]. Likewise, the node label $\text{ASP}_{[\text{OP}\ \mathit{nuc}]}$ in the upper tree of Fig. 10 is to be replaced by the label $\text{OP}_{[\text{NUC}\ [\text{ASP}\ \mathit{perf}]]}$ under the new convention.

5 Operators in complex sentences

A crucial assumption of RRG concerning the structure of complex sentences is the distinction between embedded and non-embedded dependent structures. Embedded dependent structures correpsond to *subordinations*. By contrast, non-embedded dependent structures, which are referred to as *cosubordination* structures, have basically the form $[[\ ]_X\ [\ ]_X]_X$. It is characteristic of this type of construction that operators that apply to category X are realized only once but have scope over both constituents. Cosubordination differs from the *coordination* of two independent structures in that the latter type of construction has the form $[[\ ]_X\ [\ ]_X]_Y$, where Y is a higher-level category than X.

The Turkish sentence in (6) (taken from Van Valin, 2005, p. 201) is an example of a core cosubordination construction (see also Bohnemeyer and Van Valin, 2017, p. 155f). On the surface, the deontic modal operator *-meli* ('should, ought to') is embedded in the second core, but it takes scope over the entire complex core. ('LM' stands for 'linkage marker'.)

(6) [[Gid-ip]$_{\text{CO}}$ [gör-meli-yiz]$_{\text{CO}}$]$_{\text{CO}}$.
go-LM see-MOD-1PL
'We ought to go and see.'

In the following, we leave aside the question of how to derive cosubordination structures. As a tentative working hypothesis, we may assume that the second embedded core in (6) is not added by substitution but, rather, that the first core is added to the second core by sister adjunction, due to the possible iteration of the construction. The focus of the present paper is on the adjunction of the operator and the construction of the operator projection from the constituent structure. The modal operator in (6) adjoins to the second embedded core node and it carries a feature indicating that it is a core operator. The result is the derived structure in Fig. 11 (cf. Van Valin, 2005, p. 204). In the case of a cosubordination as in Fig. 11, an operator embedded in one part of the complex structure generally takes scope over the larger category. Accordingly, in all elementary trees for cosubordination configurations, the relevant features (here MOD) are shared between the lower and the higher category in question (here the two CO nodes). This is taken to be a general property of cosubordination structures. Corresponding to this, we assume that when mapping our derived structure to the standard RRG structure, the operator targets the highest corresponding node, as long as there is no higher operator level and no substitution node in

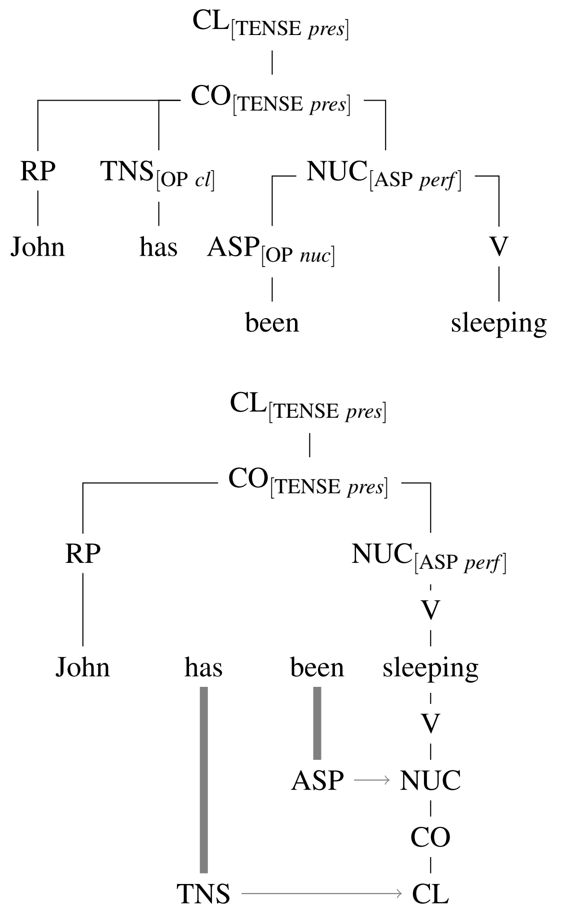


Figure 10: Derived tree for (3) (without edge features) and corresponding RRG structure

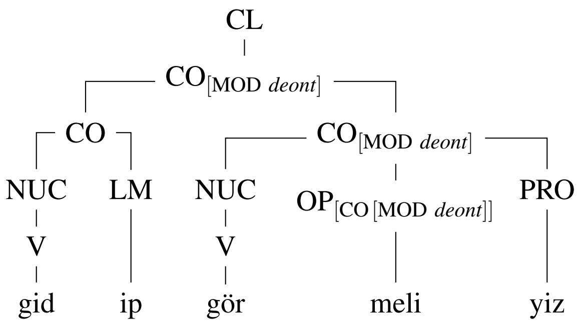


Figure 11: Derived tree for (6)

between. In the case of Fig. 11, this is the core of the entire sentence.

A similar example from English is given in (7a) (cf. Van Valin, 2005, p.203) where we have a core consisting of three embedded core constituents where the first contains the modal operator *must*. This operator takes scope over the entire large core. By contrast, in (7b) we have a structure consisting of several cores which constitute a clause. I.e., we have a core coordination and not a core cosubordination. In this case, as correctly predicted by our analysis, the modal embedded in the first core scopes only over this one and not over both cores.

(7) a. [[Kim must$_{\mathit{MOD}}$ go]$_{\text{CO}}$ [to try]$_{\text{CO}}$ [to wash the car]$_{\text{CO}}$]$_{\text{CO}}$
b. [[Kim must$_{\mathit{MOD}}$ ask Pat]$_{\text{CO}}$ [to wash the car]$_{\text{CO}}$]$_{\text{CL}}$

The shared operator scope in (7a) is a standard criterion for distinguishing cosubordinate from coordinate constructions. Another diagnostic is the independent accessibility of the embedded cores by time-positional adverbials, which are analyzed as core-level modifiers (cf. Bohnemeyer and Van Valin, 2017). While (7b) does allow independent time-positional modification, as in *Kim must ask Pat now to wash the car tomorrow*, this is not an option for (7a): Both, #*Kim must go now to try to wash the car tomorrow* and #*Kim must go to try now to wash the car tomorrow* are excluded.

Finally, let us consider a case of subordination in which the same layer category occurs twice on a path in the tree but an embedded operator targets only the lower of the two.

(8) Kim told Pat that she will arrive late.

The example in (8) (adapted from Van Valin, 2005, p. 200) involves substituting a clausal argument into the tree anchored by *told*. This substitution step is shown in Fig. 12.[3] The operator *will* in the embedded clause is a clausal operator that contributes tense. In the resulting tree, it will be dominated by the CL node of the embedded complement clause and, dominating this one, the CL node of the entire sentence. The latter, however, is not available as possible scope of this operator because there is a substitution node between this node and the operator. Consequently, we correctly predict that the tense operator only scopes over the embedded clause. Concering features, this would be reflected in the elementary tree for *told* by the fact that the two CL nodes would *not* share the TENSE feature.

[3] Note that even in a long-distance dependency such as *Who did Kim tell John that Mary likes?*, the composition operation is wrapping substitution and not adjunction.

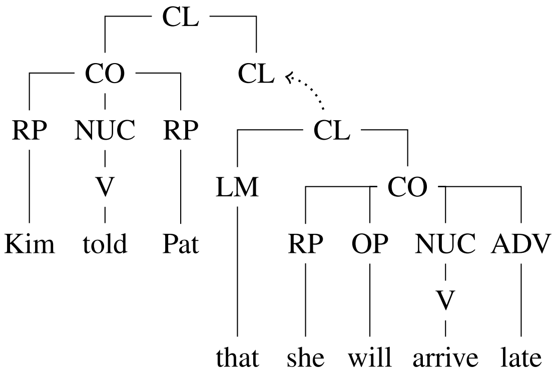


Figure 12: Derivation for (8)

6 Conclusion

The work presented in this paper is part of a larger project of formalizing the theory of Role and Reference Grammar (Foley and Van Valin, 1984; Van Valin, 2005). Based on extensive typological research, RRG assumes a rather flat constituent structure that is interleaved with an operator projection which reflects the scopal properties of functional elements. We offered a formalization of this approach that is inspired from Tree Adjoining Grammars (TAG) and that integrates both structures in one tree, modeling the operator structure within appropriate feature structures. Furthermore, due to the flat constituency structure, modifiers are added via sister adjunction and adjunction constraints are modeled via features attached to edges. The resulting architecture shares central assumptions about elementary trees with TAG but adopts a flat structure. Therefore, instead of using the standard TAG top and bottom feature structure of syntactic nodes, we proposed to use left and right features of edges in order to express adjunction constraints. Due to the features, additional projections such as the operator projection, that are not completely parallel to the constituency structure, can be captured as well.

We assume that the scopal structure of periphery modifiers can be modeled in a similar way as the one of the operators. Their scope order also depends on their order with respect to the nuclear predicate: modifiers that are more outwards scope over modifiers that are closer to the nucleus. At the same time, the surface position of a modifier does not always correspond to its scope. For instance, a modifier scoping only over the nucleus of a clause can be separated from the verb by a core constituent. An example is the aspectual adverb *completely* in (9) (Van Valin, 2005, pp. 19f).

(9) Leslie immersed herself completely in the new language.

This leads to a periphery projection that can be modeled via features in a way similar to the treatment of operators proposed in this paper.

One of the next steps towards a complete formalization of RRG will be a detailed analysis of the different types of complex sentences (subordination, cosubordination and coordination at the levels of the different layers) with respect to the composition operations and the elementary trees involved. At that point, our hypothesis that substitution nodes block the feature passing on the edges and act as islands for the operator projection will be tested again.

Another topic to be investigated concerns a binarization of RRG's flat structures. We have argued in the beginning of this paper that the left and right feature structures between edges, which are used in the flat RRG structures to constrain sister adjunction, correspond to the top and bottom feature structures on the nodes in the more binary standard TAG trees. A question is then whether we can actually define a binarization transformation for RRG that turns the features into top and bottom on the nodes and that can be used for instance for feature-based RRG parsing.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. The work presented in this paper has been supported by the Collaborative Research Centre 991 “The Structure of Representations in Language, Cognition, and Science” funded by the German Research Foundation (DFG).

References

Anne Abeillé and Owen Rambow. 2000. Tree Adjoining Grammar: An Overview. In Anne Abeillé and Owen Rambow, editors, *Tree Adjoining Grammars: Formalisms, Linguistic Analysis and Processing*, CSLI, pages 1–68.

Jürgen Bohnemeyer and Robert D. Van Valin, Jr. 2017. The macro-event property and the layered structure of the clause. *Studies in Language* 41(1):142–197.

Benoit Crabbé, Denys Duchier, Claire Gardent, Joseph Le Roux, and Yannick Parmentier. 2013. XMG: eXtensible MetaGrammar. *Computational Linguistics* 39(3):1–66.

William A. Foley and Robert D. Van Valin, Jr. 1984. *Functional syntax and universal grammar*. Cambridge University Press, Cambridge.

Robert Frank. 2002. *Phrase Structure Composition and Syntactic Dependencies*. MIT Press, Cambridge, Mass.

Carlos Gómez-Rodríguez, Marco Kuhlmann, and Giorgio Satta. 2010. Efficient parsing of well-nested linear context-free rewriting systems. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Los Angeles, California, pages 276–284. http://www.aclweb.org/anthology/N10-1035.

Mark Johnson. 1987. A new approach to clause structure in Role and Reference Grammar. In *Davis Working Papers in Linguistics 2*, University of California, Davis, CA, pages 55–59.

Aravind K. Joshi and Yves Schabes. 1997. Tree-Adjoning Grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, Springer, Berlin, pages 69–123.

Laura Kallmeyer. 2016. On the mild context-sensitivity of *k*-tree wrapping grammar. In Annie Foret, Glyn Morrill, Reinhard Muskens, Rainer Osswald, and Sylvain Pogodalla, editors, *Formal Grammar. 20th and 21st International Conferences, FG 2015, Barcelona, Spain, August 2015, Revised Selected Papers. FG 2016, Bozen, Italy, August 2016, Proceedings*. Springer, volume 9804 of *Lecture Notes in Computer Science*, pages 77–93.

Laura Kallmeyer, Rainer Osswald, and Robert D. Van Valin, Jr. 2013. Tree wrapping for Role and Reference Grammar. In Glyn Morrill and Mark-Jan Nederhof, editors, *Formal Grammar 2012/2013*. Springer, volume 8036 of *LNCS*, pages 175–190.

Makoto Kanazawa. 2009. The pumping lemma for well-nested Multiple Context-Free Languages. In V. Diekert and D. Nowotka, editors, *DLT 2009*. Springer, Berlin Heidelberg, volume 5583 of *LNCS*, pages 312–325.

Makoto Kanazawa. 2016. Multidimensional trees and a Chomsky-Schützenberger-Weir representation theorem for simple context-free tree grammars. *Journal of Logic and Computation* 26(5):1469–1516.

Rainer Osswald and Laura Kallmeyer. in press. Towards a formalization of Role and Reference Grammar. In R. Kailuweit, E. Staudinger, and L. Künkel, editors, *Applying and Expanding Role and Reference Grammar*.

Owen Rambow, K. Vijay-Shanker, and David Weir. 1995. D-Tree Grammars. In *Proceedings of ACL*.

Owen Rambow, K. Vijay-Shanker, and David Weir. 2001. D-Tree Substitution Grammars. *Computational Linguistics* .

Hiroyuki Seki, Takahashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. On multiple context-free grammars. *Theoretical Computer Science* 88(2):191–229.

Robert D. Van Valin, Jr. 2005. *Exploring the Syntax-Semantics Interface*. Cambridge University Press.

Robert D. Van Valin, Jr. and Randy LaPolla. 1997. *Syntax: Structure, meaning and function*. Cambridge University Press.

K. Vijay-Shanker, David J. Weir, and Aravind K. Joshi. 1987. Characterizing structural descriptions produced by various grammatical formalisms. In *Proceedings of ACL*. Stanford.

XTAG Research Group. 2001. A Lexicalized Tree Adjoining Grammar for English. Technical report, Institute for Research in Cognitive Science, Philadelphia.