Towards a Formalization of Role and Reference Grammar

Rainer Osswald & Laura Kallmeyer
Heinrich-Heine-Universität Düsseldorf

Abstract

We present an outline of a formalization of Role and Reference Grammar (RRG) which attempts to give full consideration to the overall architecture of RRG as a theory of grammar. The proposed formalization draws a clear distinction between declarative and procedural elements and puts emphasis on syntactic and semantic compositionality. Building on ideas from the theory of Tree Adjoining Grammars, we describe a modular way of specifying syntactic templates and introduce three general modes of syntactic composition: substitution, adjunction, and wrapping. Moreover, we sketch how to formalize the semantic representations of RRG in terms of decompositional frames and briefly discuss the issue of linking in constructional schemas.

1 Introduction

Role and Reference Grammar (RRG) has been developed as a theory of grammar which covers typologically distinct languages and which is able to capture the interaction between syntax, semantics, and pragmatics. The design of RRG was not driven by specific formal considerations. In particular, there is no formal core that plays a crucial role in RRG, as, for example, the theory of feature structures does in Head-Driven Phrase Structure Grammar (HPSG; Pollard & Sag 1994). This is not to say that the standard formulation of RRG as presented in Van Valin (2005) has no formal elements. There are the syntactic templates of the syntactic inventory, the Dowty-inspired semantic *Aktionsart* representations, the focus domains of information structure, and, last but not least, the linking algorithm. A *formalized* theory in the sense of the present paper, however, puts stronger emphasis on mathematical and logical rigor than RRG does in its current presentation. For instance, no precise specification of the possible universal and language-specific syntactic templates is given; nor is there a formal definition of how the templates can be combined to larger syntactic structures.

The working typologist, who uses RRG as her or his framework for linguistic analysis, may not regard a thorough formalization as particularly important. In fact, one of the appeals RRG has for field linguistics is that it does not come with an overly heavy theoretical load but keeps a good balance between a rich and elaborate set of notions and explanatory mechanisms, and a semi-formal, intuitive presentation. A formalization can help to identify and eliminate possible gaps and inconsistencies of the theory and, thereby, to improve the theory. In this respect, a formalization has positive impact on every user of RRG irrespective of whether she or he is interested in formal aspects or not. Moreover, a formalization can serve as a basis for computational implementations of RRG. While a thorough formalization may not be absolutely necessary for a computational treatment from an engineering perspective (e.g. Guest 2008), it can contribute to implementations that give full consideration to the overall architecture of RRG as a theory of grammar. The type of implementation we have in mind comes with a grammar development workbench which allows one to test, for instance, whether grammar fragments of language-specific phenomena provide the desired analyses and interact properly with the universal components of RRG.

In the following, we present an outline of how such a formal framework for RRG could look like. The main focus of this article is on the syntactic templates and their composition but we also discuss briefly the formal treatment of the semantic representations and the linking in constructional schemas.

2 The organization of RRG (re)analyzed

The core component of RRG is the bidirectional linking algorithm which captures how syntactic and semantic representations are related to each other (cf. Van Valin 2005, 2010). The two directions of the algorithm, from semantics to syntax and from syntax to semantics, are formulated in a procedural way as sequences of conditional instructions (*if-do* statements) embedded in an elaborate system of case distinctions. Consider the linking from semantics to syntax (for simple sentences), which consists of five main steps. Step one of the algorithm is assumed to build the complete semantic representation of what is to be expressed by the intended utterance. The semantic representation is constructed from the logical structures associated with the selected lexical units (cf. Van Valin 2006: 281), and it also integrates parts of the communicative intentions of the speaker. The second step of the algorithm is concerned with assigning the actor and undergoer roles to arguments of the predicator. In a third step, the morphosyntactic encoding of the arguments is determined, which includes the selection of the privileged syntactic argument, and the assignment of case markers, adpositions, and agreement marking. In step four, the syntactic templates for the sentence in question are selected. The final step anchors the arguments to positions in the syntactic representation of the sentence. A number of alternatives need to be considered here, such as choosing the precore slot, and the case of wh-words needs special treatment.

Kowalski (1979) succinctly characterizes an algorithm as a combination of *logic* and *control*. The logic component is about the representational structures, the compositional relations between them, and the inferential dependencies among properties of them. The control component, by contrast, defines the strategies for sequential processing and for investigating alternatives. Applied to the present context, this distinction means to keep apart the grammatical principles from their application in language production and understanding. From this perspective, the logical, declarative part of RRG is about the principles that constrain the grammatical structures and specify how these structures are related to each other. In which way the principles are put to practice is a question of how, when, and on which grounds they are applied to morphological, syntactic, or semantic information. That is, procedural elements such as assigning grammatical features to linguistic objects or selecting and combining syntactic templates come into play when the principles are applied – be it by the human cognitive system during language production and understanding, given the proposed principles are cognitively relevant, or by a field linguist who consciously employs them for analyzing her or his language data. In the current overall organization of RRG, the principles and constraints of the grammar show up only insofar as they are referred to by the linking algorithm. Examples are the macrorole assignment principle, the syntactic template selection principle, and the completeness constraint. We therefore suggest to concede the principles and constraints a more prominent place in the overall architecture by regarding them as a separate component of the grammar which is independent from procedural aspects.

Another, somewhat related issue of the current architecture is the lack of an integrative perspective on syntactic and semantic composition. The semantics-to-syntax linking algorithm assumes the semantic representation of the utterance to be fully constructed at the very beginning, based solely on the meaning of the selected words and without taking into account any of their morphosyntactic properties. Likewise, the syntax-to-semantics linking algorithm presumes that the parser, which is treated as a separate component in the architecture, is applied to the input before the linking sets in. That is, in the language comprehension process "the parser would take the input and produce a structured syntactic representation of it, identifying the elements of the layered structure of the clause and the cases, adpositions and other grammatically relevant elements in the sentence" (Van Valin 2005: 129). What kind of knowledge about grammatical principles and structures needs the parser to build on, in addition to the input sequence of words, in order to produce the required syntactic representation? Obviously, the parser needs to know about the morphosyntactic properties of the words as well as about the syntactic templates available for the language in question and the possible modes of composition. From an information processing perspective it is not very compelling to assume that the semantic contributions of the words and the constructional schemas involved are ignored in this step. In fact, the discussion in Van Valin (2006: 283ff) about RRG and language comprehension shows that RRG does not exclude more flexible processing options and, for example, supports the use of complex templates for frequent syntactic-semantic patterns. For these reasons, we propose a view on the processing of linguistic information that allows for a stronger interlacing of syntactic and semantic composition than possible within the current architectural restrictions of RRG.

The program of formalizing of RRG starts with a formal account of the representational structures involved. In Section 3, we will do this in some detail for the syntactic representations, the compositional relations between them, and the templates of the syntactic inventory. We introduce three modes of composition, (simple) substitution, (sister) adjunction, and wrapping (substitution), which seem to be sufficient for compositionally deriving all kinds of complex syntactic constructions. The syntactic templates of the syntactic inventory, on the other hand, will be specified by means of a tree description language. In Section 4, we argue for formalizing the semantic representations of RRG in terms of decompositional frames, which are defined as base-labeled feature structures with types and relations. Section 5 presents a brief case study of how the framework developed so far can be employed for formalizing the linking of syntax and semantics in constructional schemas.

3 Syntactic structure

In RRG it is assumed that the syntactic representations are composed of syntactic templates stored in the syntactic inventory. Figure 1a shows a simple example of a syntactic representation; possible candidates of syntactic templates are shown in Figure 1b.

For a formalization we first need to decide on what kind of formal structures to use. Keeping close to the representations of RRG, we use tree structures for that purpose, where nodes can carry additional features besides the category labels (Section 3.1). The second task is to define the modes by which the syntactic representations are composed from the members of the syntactic inventory (Section 3.2). Next we need to specify the templates available in the inventory, and we need to do this in a way that allows us to capture generalizations among them within and across languages (Section 3.4).

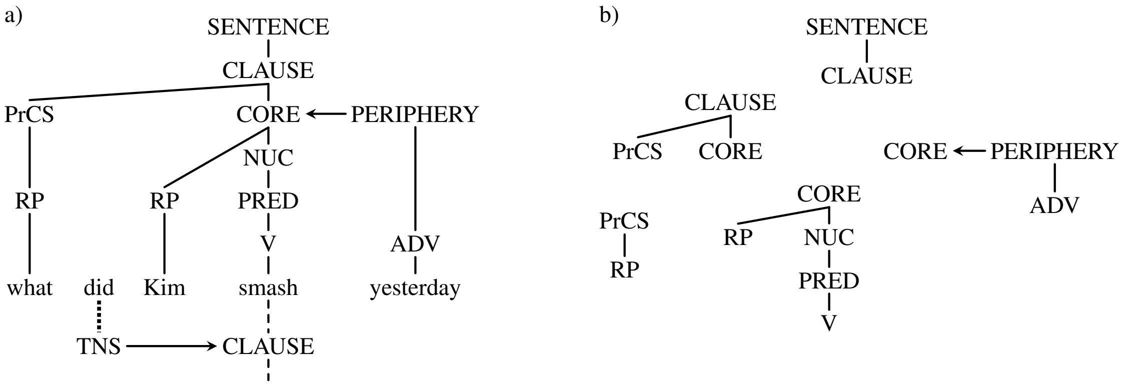


Fig. 1: Examples of syntactic representation and syntactic templates in RRG

3.1 Syntactic representations

A key component of RRG's approach to syntactic analysis is the layered structure of the clause: sentences are assumed to have an internal structural layering consisting of clause, core, and nucleus. The different layers serve as attachment sites for different types of operators: tense operators attach to the clause, modality to the core, aspect to the nucleus, etc. The core level is also the default attachment site for arguments. In the following, we will refer to the subtree of a syntactic representation consisting of the root and its non-peripheral clause, core, nucleus, and predicating descendants as the *clausal skeleton* of the representation.

The syntactic structures in RRG are basically labeled trees, and there are good reasons to use tree structures in a formalization as well. Trees provide the most natural way to analyze syntactic structures since they build on the basic relations *immediate dominance* and *linear precedence*.[1] Having decided on using trees, we need to cope with the more specific aspects of RRG's syntactic presentations such as the operator projection and the arrow notation for peripheral constituents (cf. Figure 1).

Representing the operator structure as a separate projection was originally proposed by Johnson (1987), based on the observation that the ordering among the operators is systematically correlated with their scope given by their attachment site at the clausal skeleton, whereas the surface order of the operators relative to arguments and adjuncts is much less transparent and often requires crossing branches. Johnson's idea was to use two different context free grammars for analyzing the sequence consisting of the verb plus arguments and adjuncts on

[1] There are other options for modeling constituent structures. In HPSG, for instance, constituent structures are modeled by *feature structures*, and Nolan (2004) suggests the same for RRG. But employing feature structures comes at the price of introducing features, or attributes, by which the subconstituents can be accessed. This can either be done by reconstructing tree structures as feature structure based on formal features such as FIRST and REST, or by employing functional notions like SUBJECT, DIRECT-OBJECT, etc. However, in RRG, configurational syntactic notions are *derived* concepts at best and, therefore, should not be part of the underlying modeling.

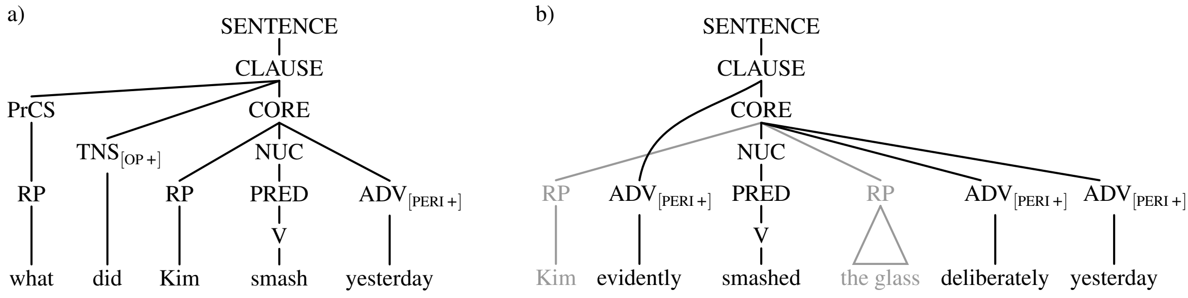


Fig. 2: Periphery and operator marking by features

the one hand, and the sequence of the verb plus operators on the other hand. The two grammars taken together then constitute a *projection grammar*. Hence the names *constituent projection* and *operator projection*. While it seems reasonable to distinguish between constituent structure and operator structure, the proposal of Johnson has the problem of being purely surface oriented. As a consequence, it does not enforce matching clausal skeletons in the two projections. However, corresponding clausal skeletons in both projections are taken for granted in the syntactic representations of RRG. For this reason, our formalization assumes a single syntactic structure in which operator components are distinguished by the feature [OP +] (cf. Figure 2a). We will see below that this representation, together with the approach to operator adjunction presented in Section 3.2.2, is sufficient to capture the scope-related ordering among the operators. The operator projection can then be defined as the subtree consisting of the clausal skeleton plus the components marked by [OP +].

In RRG's syntactic representations, peripheral structures and clause linkage markers are attached to the clausal skeleton by arrows, which is a further non-standard element to be dealt with. In our proposal, we do not posit separate PERIPHERY nodes but mark peripheral structures by a feature as indicated in Figure 2. Similar to the operators, peripheral elements are subject to the iconicity principle that their relative ordering respects the layering of their attachment sites (see Figure 2b for an illustration). The modes of combining peripheral structures with the clausal skeleton are similar to those used for operators (Section 3.2.2). As to clause linkage markers, they can be treated along similar lines. However, since they behave more like prepositions and oblique case markers than like peripheral elements or operators, we regard their proper syntactic representation as an issue that calls for further clarification before coming up with a definite proposal of their formalization.

To sum up, we formalize the syntactic representations of RRG as labeled trees, where the labels can have additional features. Since the node labels can be regarded as feature values, too, e.g., category labels as values of the feature CAT, we can assume without restriction of generality that node labels are sets of attribute-value specifications. From this perspective, $\text{ADV}_{[\text{PERI}\ +]}$ is short for [CAT *adv*, PERI +]. Introducing features allowed us to get rid of the PERIPHERY nodes, whose only purpose is to mark their daughters as peripheral. By the same line of reasoning we can eliminate the PRED nodes, whose only purpose is to mark their daughters as predicating (cf. Van Valin 2005: 13). The system of node attributes and the constraints on their values are still under development at the present stage of the formalization, and we will accordingly make a rather liberal use of features without specifying the full feature set of a node and without spelling out in detail the feature co-occurrence restrictions and related constraints.

3.2 The composition of syntactic structures

The standard presentation of RRG gives only an informal description of how syntactic templates are combined to more complex syntactic structures. Van Valin & La Polla (1997: 654, note 34) suggest that a formal account of the modes of composition may show some similarity to Tree Adjoining Grammars (cf. Joshi & Schabes 1997). Our approach to formalizing the syntactic side of RRG basically follows this hint.[2]

3.2.1 Substitution

The most basic mode of composition for syntactic templates is *substitution*. In the following, a *tree with label* X is meant to be a tree whose root carries the category label X. A tree β with label X can be substituted for a leaf node labeled with X (the *substitution node*) of a tree α by "identifying" the root node of β with the substitution node (cf. Figure 4a). More generally, if the nodes are labeled by feature structures, then the two feature structures must be compatible, and the node of the resulting tree is labeled by the *unification* of the two feature structures.

Substitution is the main mode of composition for expanding argument nodes by the syntactic representations of specific argument realizations. In Figure 3a, the two RP structures associated with *Kim* and *the glass* are substituted at the RP nodes of an argument structure template. The template in turn is selected by the predicating lexical element *smash(ed)*, where selection means substitution of the lexical tree at the predication leaf of the argument structure template (similar to lexical anchoring in Lexicalized Tree Adjoining Grammars). Finally, the substitution of the CLAUSE node at the top turns the clause into a complete sentence.

[2] Van Valin & La Polla (*ibid.*) rightly point out that TAG is a *grammar formalism* and not a *linguistic theory* in its own right. That is, TAG *per se* does not make any commitments about what kind of categories and what kind of syntactic configurations are appropriate for linguistic analysis. However, there is one caveat: the standard adjunction operation of TAG is not well suited for modifying flat structures, which is one of the reasons why our formalization employs slightly different modes of composition.

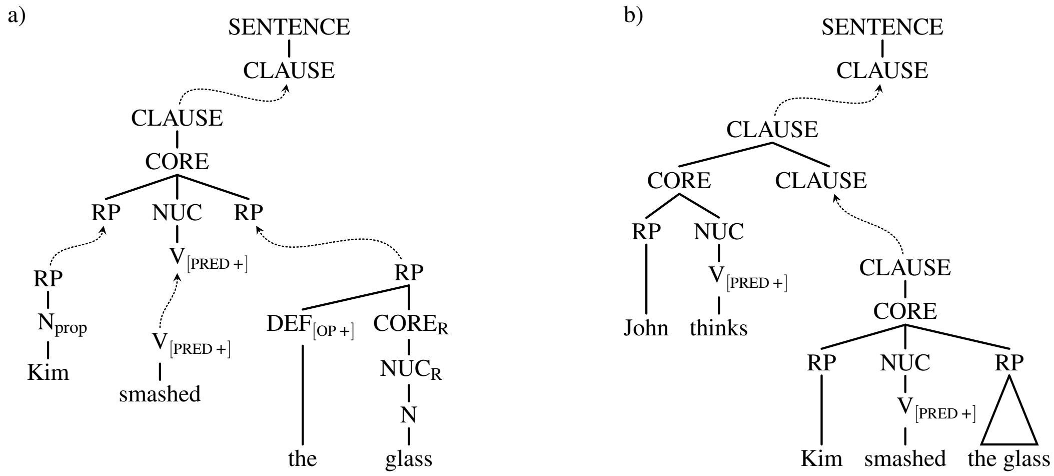


Fig. 3: Examples of substitution

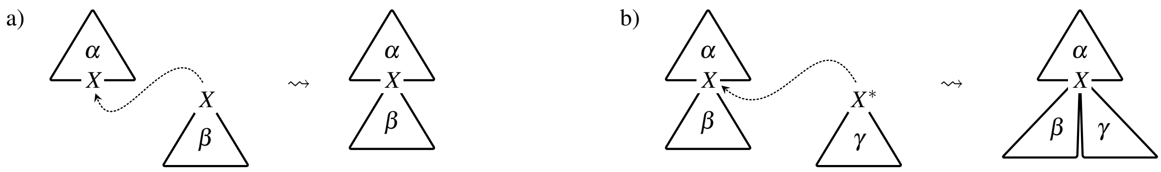


Fig. 4: Schematic sketch of substitution a) and adjunction (to the right) b)

Figure 3b shows an example in which the same clause is substituted as the complement of a clausal subordination construction associated with the verb *think*.

3.2.2 Adjunction

Peripheral structures cannot be added by substitution since they do not attach to leaves but to internal nodes in general, and the same holds for the operators; cf. Figures 1 and 2. The mode of composition we propose for these cases is *(sister) adjunction* (see also Kallmeyer *et al.* 2013).[3] As with substitution, we assume that the templates available for adjunction have a root label which coincides with the label of the target node (cf. Figure 4b). For convenience, the root of an adjunction tree is marked by an asterisk in the graphical presentations. Figure 5 sketches how the operator and periphery elements of the syntactic representations in Figure 2 can be added by adjunction. Notice that the root label of an adjunction tree specifies the attachment site at the phrasal skeleton. For example, the adjunction template selected by a temporal adverb like *yesterday* specifies the core as its layer of attachment. We also allow the substitution or the adjunction of trees at adjunction trees. For instance, the adjunctions of the

[3] Sister adjunction has been introduced as a composition operation on so-called *d-trees* by Rambow *et al.* (1995). Notice that sister adjunction differs from adjunction in TAG. In fact, the latter operation is more akin to the wrapping operation described in Section 3.2.3 below.

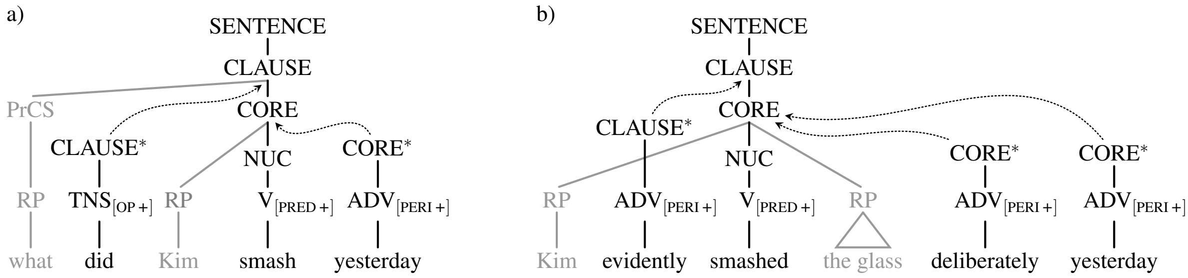


Fig. 5: Attachment of periphery and operator by (sister) adjunction

'deliberately' tree and the 'yesterday' tree to the core node in Figure 5b can be reduced to a single adjunction to the core node by first adjoining the 'yesterday' tree (to the right) of the 'deliberately' tree (or vice versa).

The examples in Figure 5 show a fairly arbitrary ordering of the adjoined substructures with respect to their sister nodes; even discontinuities can arise as shown by the placement of *evidently* within the core. However, within the operator and periphery projections, adjunction reduces to adjunction to the left or to the right of the daughters of the target node. It then remains to merge the projections with the constituent structure subject to additional ordering constraints, which are often language-specific. For instance, the tense operator must be placed between the precore slot and the core in Figure 5a, and the peripheral adverb *evidently* occurs between the first RP and the nucleus in Figure 5b. The position of the tense operator and its realization paradigm (here, by an auxiliary) is determined by the constructional schema of wh-questions in English. The placement of the peripheral adverbs is less restricted as long as the layer-related iconicity within the periphery projection is respected. For English, additional language-specific constraints would ensure that adverbials are not inserted between the verb and the following core RP argument and that phrasal adverbials go systematically to the right of the clausal skeleton.

3.2.3 Wrapping

Control constructions and extraction from complements pose a problem for the modes of composition presented so far. Consider the examples of wh-extraction in (1).

(1) a. What does John think Kim smashed?
    b. What does John think Mary claimed Kim smashed?

Clearly, it would not be appropriate to assume a separate complex template for each of these constructions. The syntactic representations are to be composed of basic argument structure templates in a systematic way. There are several options for achieving this goal, depending on

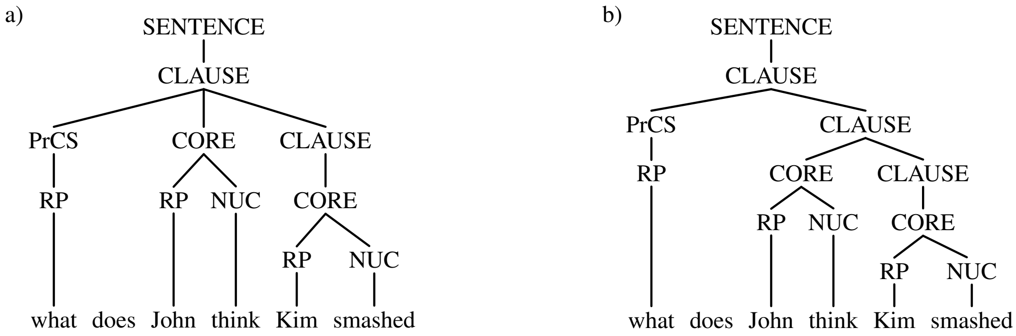


Fig. 6: Two possible syntactic representations of wh-extraction from complements

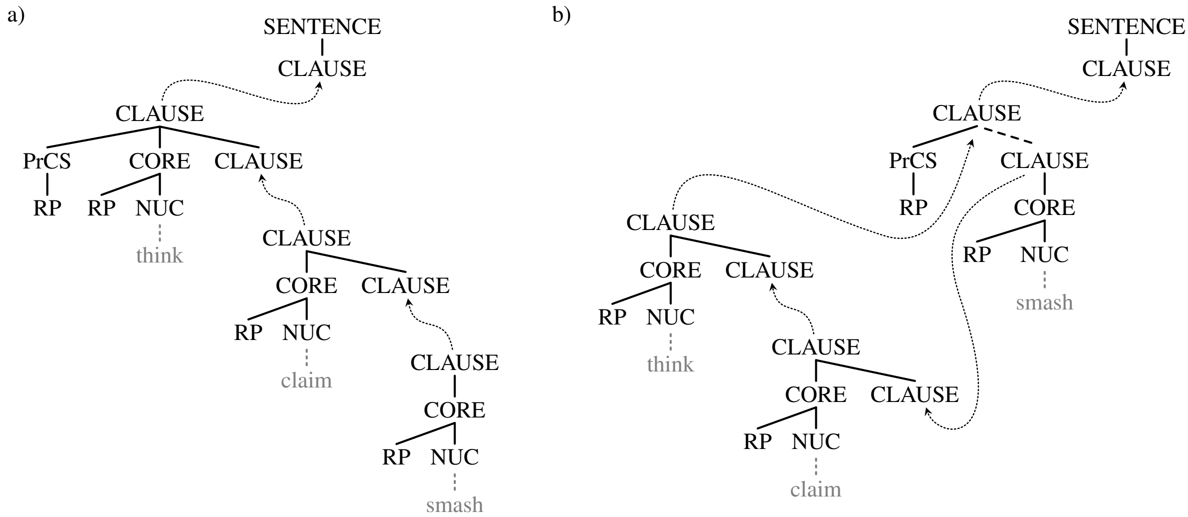


Fig. 7: Wh-extraction via simple substitution a) and wrapping substitution b)

the presumed inventory of elementary templates. First we need to decide on the proper syntactic representations of the examples in (1). While sentences of this type are discussed in the context of island constraints in Van Valin & La Polla (1997: 615) and Van Valin (2005: 273), no structural analysis is provided there. Due to the nature of the embedding constructions, the basic binary precore slot pattern [$_{\text{CLAUSE}}$ [$_{\text{PrCS}}$ ...][$_{\text{CORE}}$ ...]] shown in Figure 1 and Figure 5a does not apply to the present case. Figure 6 shows two possible alternatives. The analysis in Figure 6a assumes a precore variant of the clausal subordination pattern [$_{\text{CLAUSE}}$ [$_{\text{CORE}}$ ...][$_{\text{CLAUSE}}$ ...]]. The structure in Figure 6b, by contrast, assumes an additional clause node, and the precore slot pattern is [$_{\text{CLAUSE}}$ [$_{\text{PrCS}}$ ...][$_{\text{CLAUSE}}$ ...]]. In the following, we restrict the discussion to the first option since we regard it as difficult to come up with an independent motivation for the additional clausal node in Figure 6b.

Figure 7 sketches two ways of composing the syntactic representation of example (1b). Figure 7a employs substitution only, but at the price of assuming a special elementary template associated with ‘think’ that has a precore slot in addition to its normal argument slots.

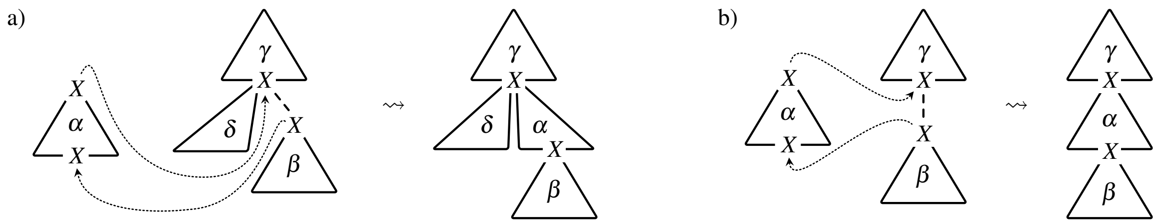


Fig. 8: Wrapping substitution (special cases)

Assuming such a template would raise the further problem of providing information about which of the arguments within the embedded clauses is referred to by the referent phrase in the precore slot. The templates in Figure 7b are more straightforward in this respect since they represent proper argument structure templates, that is, the precore slot is locally connected to the core from which the wh-word is extracted. In fact, the long-distance dependency comes about by the insertion of material between the precore slot and the corresponding core. On the downside, more flexible templates and a more complicated tree composition mechanism are required in this case. The clause node of the 'smash' structure is "split in two" and its upper part is unified with the upper clause node of the 'think' structure, thereby keeping the precore slot on the left side of the structure (similar to the adjunction mechanism introduced in the previous section); the lower part of the split node is substituted at the lower clause node of the 'claim' structure. The mode of composition just described is schematically depicted in Figure 8a and will be referred to as *wrapping substitution*.[4]

Let us evaluate the different tree composition options with respect to their applicability for linking, especially with respect to template selection. As pointed out above, it is not an option to assume that the syntactic inventory provides a single template for the complex syntactic structure of (1b). The structure has to be composed from argument structure templates which in turn are selected by the chosen lexical entries. The composition in Figure 7a has the disadvantages that the template selected by 'think' is not an argument structure template but has to be stipulated for embedded wh-questions, and that an advanced mechanism for coreference is needed. The composition scheme in Figure 7b, by comparison, has the advantage that the precore slot can be immediately linked when the template is selected. Here, the underlying assumption is that the syntactic inventory provides argument structure templates for wh-fronting.

Multiply embedded control and matrix-coding constructions may serve as another touchstone for template composition. They pose a number of challenges for RRG's syntactic analysis of such constructions as presented in Van Valin & La Polla (1997) and Van Valin (2005).

[4] Wrapping substitution is related to the concept of *flexible composition* proposed in Joshi *et al.* (2008), which allows one to interpret TAG adjunction as a wrapping operation.

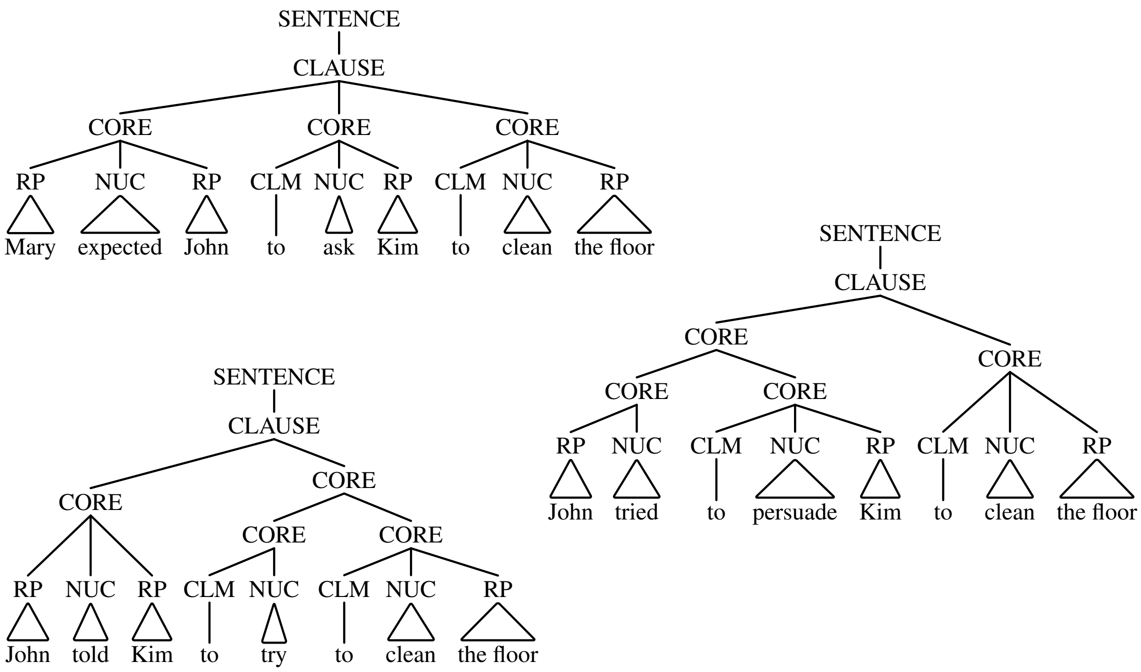


Fig. 9: Syntactic representations of the examples in (2)

Consider the examples in (2) and their syntactic representations in Figure 9.

(2) a. Mary expected John to ask Kim to clean the floor.
b. John tried to persuade Kim to clean the floor.
c. John told Kim to try to clean the floor.

Structures like the ones in Figure 9 show a mismatch between syntax and semantics in the following sense: semantically, the infinitival complements are arguments of the respective control predicates, but syntactically, they do not behave like core arguments. Example (2a) is instructive in showing how multiply embedded constructions can give rise to coordinating syntactic configurations. Notice that the clausal complements discussed in the previous section show also a mismatch between syntax and semantics in that they do not attach to the core but to the clause. But in contrast to the present case, multiple embeddings of clausal complements nevertheless correspond to embeddings on the syntactic side, though at the clause level and not at the core level. For this reason, simple substitution is sufficient for clausal complementation. This is different for the coordination structure of example (2a). Simple substitution cannot be applied here if we assume that the verbs *expect* and *ask* select elementary templates that contain a core daughter of the clause for the infinitival complement. Figure 10a sketches a possible solution that uses wrapping substitution, where we allow split nodes to carry differing top and bottom categories. Figure 10b shows that the same compositional mechanisms

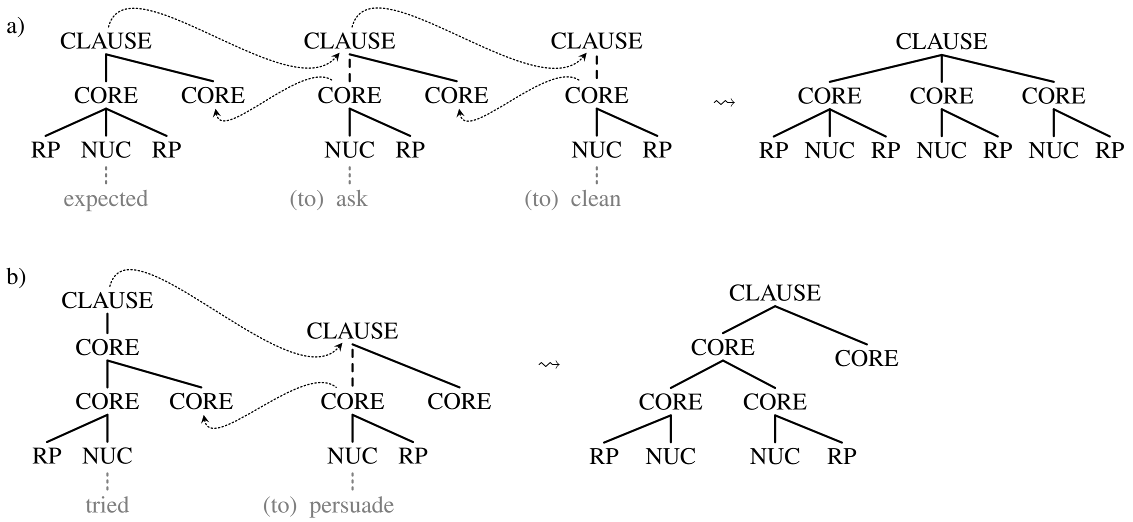


Fig. 10: Composition of templates by wrapping substitution for (2a) and (2b)

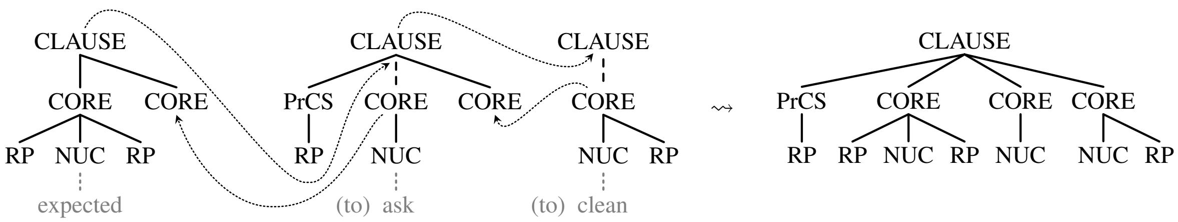


Fig. 11: Composition of the syntactic representation of (3a)

works for control verbs such as *try* which call for a core cosubordination template. Moreover, the treatment of wh-fronting introduced above nicely extends to embedded control and matrix-coding constructions like those in (3).

(3) a. Whom did Mary expect John to ask to clean the floor?

b. What did Mary expect John to ask Kim to clean?

For example, the syntactic representation of (3a) can be composed of elementary argument structure templates as illustrated in Figure 11. The general form of wrapping substitution used in this case in schematically depicted by Figure 12.

3.3 Extended domain of locality and the factoring of recursion

Consider again the composition scheme in Figure 7a. According to the traditional template selection principle of RRG, we can assume that the structure associated with 'smash' is available in the syntactic inventory. The number of core slots is reduced by one compared to the default core template of *smash* since one of the arguments occurs in precore position. As discussed above, this approach has the disadvantage that argument linking is not local; the wh-marked

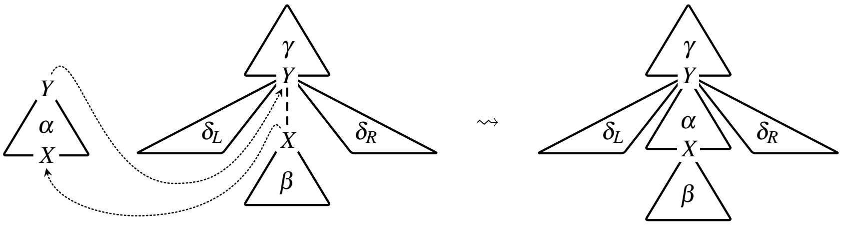


Fig. 12: Wrapping substitution (general case)

argument of *smash* can only be linked after the precore slot template is attached. But at this point of the composition, the referent of the wh-word is not locally accessible anymore because of the intervening clausal subordination structures. A possible solution is to have the information about the wh-marked argument percolate to the top of the tree. A constraint-based formalization of this percolation process could roughly work as follows: The core node of the reduced template for 'smash' carries a (set-valued) feature which contains the referential index of the participant not locally realized as well as its wh-marking. A general constraint then ensures that clause and sentence nodes collect the non-realized indices of their subordinate clauses and cores minus the indices that are realized in precore slot daughters and the like. This way of bookkeeping for modeling long distance dependencies is in fact closely related to the use of 'slash' or 'gap' features in the approaches of Sag & Wasow (1999) and Ginzburg & Sag (2001), among many others. To sum up, the approach just discussed can get along with simple substitution at the price of a considerable amount of bookkeeping. By comparison, the preferred approach exemplified in Figures 7b allows a fully local argument linking but requires a more complex method of tree composition. The former strategy is employed in GPSG/HPSG-related frameworks while the latter strategy is characteristic of approaches in the line of Lexicalized Tree Adjoining Grammars (LTAG; cf. Kroch 1987).

Two important characteristics of the LTAG formalism are the *extended domain of locality* of elementary trees and the *factoring of recursion from the domain of dependencies* by means of TAG-adjunction (Joshi & Schabes 1997: 95f). The first property means that elementary trees represent full argument projections and that they can have a complex constituent structure. The usual example given in this context is that of elementary argument structure trees associated with transitive verbs which contain an internal VP constituent. Since in RRG, the basic argument structure templates are flat, the availability of deeper hierarchical structures is less relevant in the first place. This observation hints at a crucial difference between RRG and the way LTAG is usually applied to linguistic analysis. In the LTAG literature, a standard example of the factoring of recursion by adjunction is the insertion of adverbials. Put briefly, an adverbial is added by adjoining a so-called auxiliary tree, which leads to an additional branching. That is, if a VP node or some other internal node is required as the landing site of the adverbial auxiliary tree, then the structure gets deeper by adjunction and the verb and the direct object may become farer apart.

From an RRG point of view the described effect may be regarded as an artifact of the structural assumptions. If argument structure templates are basically flat and if modifiers are added as sister structures then there is no increase of structural depth in this case. In other words, the structure of the clausal skeleton is not affected by modification. The situation is different in the case of long distance dependencies across clausal complements. Clausal complements increase the depth of the clausal skeleton. It is this case where the notions of an "extended domain of locality" and of "factoring recursion from the domain of locality" directly apply to RRG, at least if we presume elementary argument structure templates with precore slots as illustrated in Figure 7b. The multiple coordination structure of Figure 11 is another type of example to which the factoring of recursion from the domain of locality applies. In this case, the clausal skeleton does not grow in depth but in breadth.

3.4 The syntactic inventory

3.4.1 Syntactic templates

The syntactic templates of RRG are introduced as the basic building blocks of the syntactic representations. No attempt has been made up to now to specify exactly the templates of the syntactic inventory. The discussion of the two approaches to wh-fronting in Section 3.2.3 shows that the question of which templates need to be available in the inventory is not independent of the modes of composition employed. If extraction structures are fully linked locally, as in Figure 7b, then there is no need for a precore slot template to be attached separately. In our formalization proposal, we assume that argument structure templates are clause templates and have slots for each of the arguments occurring in the clause. Let us call this the *full clause projection* assumption.[5]

For example, the templates in Figure 13 represent alternative realization patterns of transitive verbs in English. Notice the two templates for the passive, one in which only the undergoer occurs as an argument, and a second template which includes the realization of the actor by a peripheral *by*-phrase. Standard RRG would probably regard the peripheral *by*-phrase as a separate template that can be adjoined to the passive core. In the context of the framework developed so far, the question is whether the *by*-phrase should be added by (sister) adjunction like an adverbial. By listing the two realization patterns for the passive as elementary templates in Figure 13, we opt against adjunction in this case. The main disadvantage of

[5] A similar constraint on elementary trees is often proposed in TAG-based approaches to linguistic analysis; cf., e.g., Frank (2002).

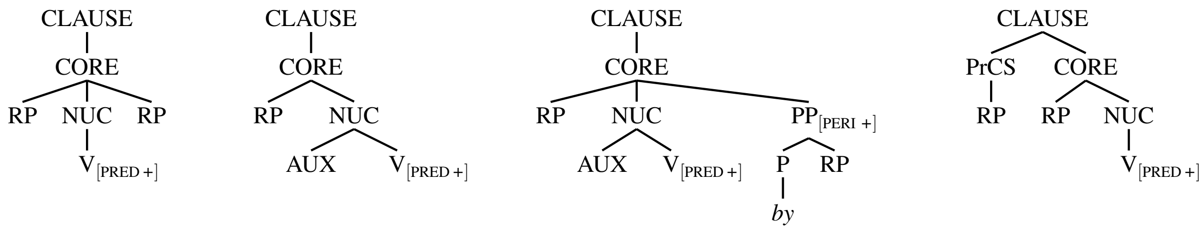


Fig. 13: Basic transitive predication template for English with variants

modeling the *by*-phrase by adjunction would lie in the constraints that need to be imposed, since adding a peripheral *by*-phrase that encodes the actor is restricted to specific circumstances. The optional peripheral *by*-phrase is part of the description of how passive voice is realized in English. Of course, a mere enumeration of the two passive templates in Figure 13 is not satisfactory for a theory of grammar. At some point, the theory should state explicitly that it is the addition of the *by*-phrase which relates the second template to the first. In our framework, this relation is encoded at the level of template *specifications* (cf. Section 3.4.2).

The rightmost template in Figure 13 represents an elementary precore slot argument structure template. This differs from the template system informally suggested in Van Valin (2005: 15), which regards the structure in question as composed of a reduced core template and a precore slot-clause fragment (cf. Figure 1b). Again, we propose that compositions of this type are best modeled at the level of template specifications. It is worth mentioning that the precore slot template in Figure 13 covers actually two different types of argument realization patterns: fronted elements in declarative sentences (as in *Bean soup I don't like*), which go along with focal stress, and fronted wh-words in interrogative sentences of the kind discussed above. The latter pattern differs from the former in the way tense is realized. The commonalities between the two templates would then again be expressed on the specification level.

3.4.2 Template specification by tree descriptions

In the previous section, we raised the question of how to characterize the (universal and language-specific) syntactic templates in a systematic way. The templates proposed above as elementary are more complex than the fragmentary templates usually assumed in RRG. The key advantage of our approach is that the composition of syntactic representations can be reduced to the three modes of composition presented in Section 3.2, namely substitution, adjunction, and wrapping. However, this leaves us with the problem of how to describe in which way the templates are built from more elementary components, and how they are related to each other. As mentioned before, it would be rather unsatisfactory if we had to regard the two

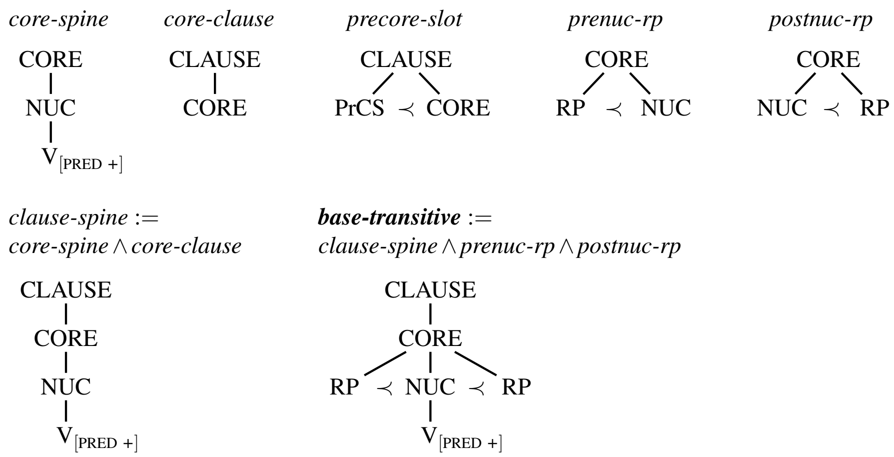


Fig. 14: Example specifications of syntactic fragments

passive templates in Figure 13 as independent units without being able to make explicit the relation between them.

The proposed solution is to treat syntactic templates as *minimal models* of tree descriptions. Relations between templates can then be captured by relating the respective descriptions. For instance, the specification of the passive template with *by*-PP consists of the specification of the simple passive template plus a specification of the *by*-PP and constraints on its position. Our use of tree descriptions for specifying syntactic templates in a modular way is inspired by the *metagrammar* approach of Crabbé & Duchier (2005), where a metagrammar is basically a system of tree descriptions that defines the syntactic inventory as the set of the corresponding minimal models.

Tree descriptions in this sense consist of dominance and precedence constraints as well as category and feature assignments. Consider the example specifications in Figure 14, which are depicted in tree-like diagrams but are to be read as tree descriptions. For instance, the specification with the name *precore-slot* says that there are three distinct nodes n_0, n_1 and n_2 labeled respectively by CLAUSE, PrCS, and CORE, where n_1 and n_2 are daughters of n_0 and n_1 immediately precedes n_2 (expressed by $\prec$). Figure 14 illustrates how the basic transitive template of Figure 13 can be defined by a piecewise combination of such specifications. The precore slot template of Figure 13 can be likewise defined by conjoining the specifications *clause-spine*, *precore-slot*, and *prenuc-rp*. In this way, common components of elementary templates are made accessible in the metagrammar.

4 From logical structures to decompositional frames

There are various ways to introduce a formal account of RRG's logical structures. Probably the most straightforward option is to regard them as expressions of a formal language, which means to give an inductive definition of these expressions by specifying their atomic components and the modes of combining them to larger expressions.[6] However, this approach has the following two disadvantages: First, there are a number of additional semantic components used within RRG's logical structures whose formal syntax seems not fully fixed yet. Among them are the internal variables (Van Valin & La Polla 1997: 117), the qualia structures, the representation of the operator semantics, and the representation of the semantics of the intra-clausal connectives. All of these representational elements would affect the syntactic make-up of the logical structures and should thus be part of their formal definition. A second issue is the interpretation of the logical structures, which is discussed in RRG expositions only at an intuitive level. From a formal point of view, the logical structures are meaningless expressions built in accordance with certain formation rules. While not absolutely necessary for a formalization of RRG as is, we think it desirable to also have a formal account of the semantics represented by the logical structures. A logically more explicit formulation of RRG's semantic representations is needed anyway, for instance, for drawing inferences.[7]

Instead of working out the formal syntax of the logical structures, we propose a slightly different approach for representing the semantic information they encode. The goal of our formalization is to keep the key properties of RRG's decompositional system without necessarily preserving the specific form of the logical structures. Moreover, we aim at a *formal semantics* account, which does not only address the form of the semantic representations but also their interpretation. For these reasons, among others, we make use of *decompositional frames* for the semantic representation. As we will see below, semantic frames, understood as feature structures, allow a fairly straightforward reconstruction of the logical structures of RRG.[8] The theory of feature structures come with a well-explored logic and a model-theoretic semantics; another advantage is the availability of *unification (under constraints)* as a universal mode of composition (cf. Kallmeyer & Osswald 2013 and the references therein).

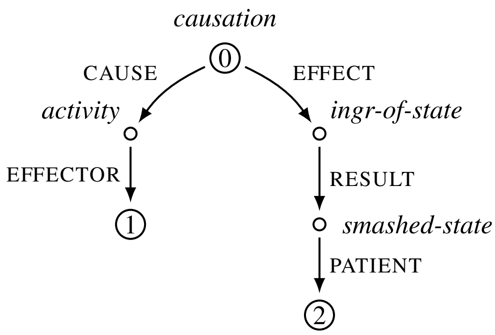
A central purpose of the decompositional system that underlies the logical structures of RRG is the representation of *Aktionsart* types. The representational system allows the distinction between states, activities, changes of state, causative scenarios, etc., and it seems reasonable to assume that the decompositional structures are intended to reflect the internal

[6] This is what Dowty (1979: 352ff) does for his "translation language", which contains the decompositional elements Foley & Van Valin (1984) originally build on. In contrast to RRG's logical structures, Dowty's language is a full-fledged logical language with lambda abstraction which has a model-theoretic semantics.

[7] Cf. Periñan & Mairal (2009) for an argument in this direction.

[8] Cf. Nolan (2004) for a related proposal.

frame / feature structure



description in attribute-value logic

$\boxed{0} : \textit{causation} \wedge \boxed{0} \cdot \text{CAUSE} : \textit{activity} \wedge \boxed{0} \cdot \text{CAUSE EFFECTOR} \triangleq \boxed{1} \wedge$
$\boxed{0} \cdot \text{EFFECT} : \textit{ingr-of-state} \wedge \boxed{0} \cdot \text{EFFECT RESULT} : \textit{smashed-state} \wedge$
$\boxed{0} \cdot \text{EFFECT RESULT PATIENT} \triangleq \boxed{2}$

$$\exists e' \exists e'' \exists s\,(\textit{causation}(\boxed{0}) \wedge \text{CAUSE}(\boxed{0}, e') \wedge \text{EFFECT}(\boxed{0}, e'') \wedge \textit{activity}(e') \wedge \text{EFFECTOR}(e', \boxed{1}) \wedge \textit{ingr-of-state}(e'') \wedge \text{RESULT}(e'', s) \wedge \textit{smashed-state}(s) \wedge \text{PATIENT}(s, \boxed{2}))$$

description in predicate logic

$$\boxed{0}\begin{bmatrix} \textit{causation} & \\ \text{CAUSE} & \begin{bmatrix} \textit{activity} \\ \text{EFFECTOR } \boxed{1} \end{bmatrix} \\ \text{EFFECT} & \begin{bmatrix} \textit{ingr-of-state} & \\ \text{RESULT} & \begin{bmatrix} \textit{smashed-state} \\ \text{PATIENT } \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

attribute-value matrix notation

Fig. 15: Frame representation corresponding to (4)

structure of the described event or state of affairs at some level of conceptual representation. Our frame-based formalization tries to make explicit the reference to events and their subcomponents (see also Osswald & Van Valin 2014). Consider, for example, the decompositional structure shown in (4), which is associated with the causative reading of the English verb *smash*.

(4) [**do'**(x, ∅)] CAUSE [INGR **smashed'**(y)]

The logical structure basically says that an event denoted by transitive *smash* consists of an activity of someone or something *x* which causes a sudden change of state of something *y*, namely *y* getting into the state of being smashed. Put differently, the event in question is of type *causation* and has as its CAUSE component an (unspecified) activity of *x* and as its EFFECT component an ingression of a state which results in *y*'s state of being smashed.[9] This means that the event conceptualization associated with the transitive use of the verb *smash* consists of certain subcomponents (activities, changes of state, etc.), the participants involved, and various relations between these entities. The left side of Figure 15 shows a graphical representation of the structure for the example under discussion. (The numbers are labels for referring to specific components; they will become relevant below in the context of linking.) An essential property of this relational structure is that all relations are *functional*:

9 Note that the use of CAUSE here differs from the one in (4); CAUSE denotes a functional relation that relates an event of type causation to its cause component. In (4), by contrast, CAUSE is to be interpreted as a relation between an activity and a change of state, that is, as the causation relation between events.

it is *the* cause component of the causation event, *the* effector of the activity, etc. Moreover, all components of the structure are "accessible" by functional relations from a distinguished set of *base* elements. In the example, every node is accessible from the base element labeled by 0, which, in this case, is the *root* of the structure. We refer to structure of the kind just described as *feature structures* or *frames*. More precisely, frames are defined as *base-labeled feature structures with types and relations*; see Kallmeyer & Osswald (2013: 280ff) for a formal definition.

By treating decompositional semantic representations as frame structures, we take a clear position with respect to the distinction between logical descriptions and described models. Frame structures are models; they are relational structures of a specific kind and they can be easily characterized in terms of first-order predicate logic. For instance, the frame structure depicted on the upper left of Figure 15 is a generic minimal model of the formula shown on the lower left of the Figure, where the boxed labels are used as constants and the predicates CAUSE, EFFECT, EFFECTOR, etc. are assumed to be functional. Logical representations of this kind are closely related to Neo-Davidsonian approaches to event semantics as proposed in Parsons (1990), among others. It is also possible to characterize frame structures by means of special purpose logics, so-called *feature* or *attribute-value logics*, which can be seen as variable-free fragments of first-order predicate logic. For the example under discussion, a possible attribute-value description is shown on the upper right of Figure 15. The more familiar attribute-value matrix below can be seen as a special syntactic device for writing attribute-value descriptions.

5 Linking and constructional schemas

A thorough formalization of the RRG linking system is far beyond the scope of the present paper. For this reason, and because of lack of space, we restrict the discussion on linking and constructional schemas to a case study of the English adjectival resultative construction (*kick open*, *wipe clean*, etc.). A schema for this construction can be found in Van Valin (2005: 239). The idea of constructional schemas in RRG is that "[o]nly the idiosyncratic, language-specific features of constructions are represented in constructional schemas. Hence constructional schemas, by virtue of their reference to general principles, permit the capturing of cross-linguistic generalizations, while at the same time expressing language-particular properties of grammars" (Van Valin 2005: 132).

In English (and many other languages), adjectival resultative constructions are nuclear cosubordination structures. The two predicates form a complex predicate, which means that argument linking works basically in the same way as for simple sentences (Van Valin 2005:

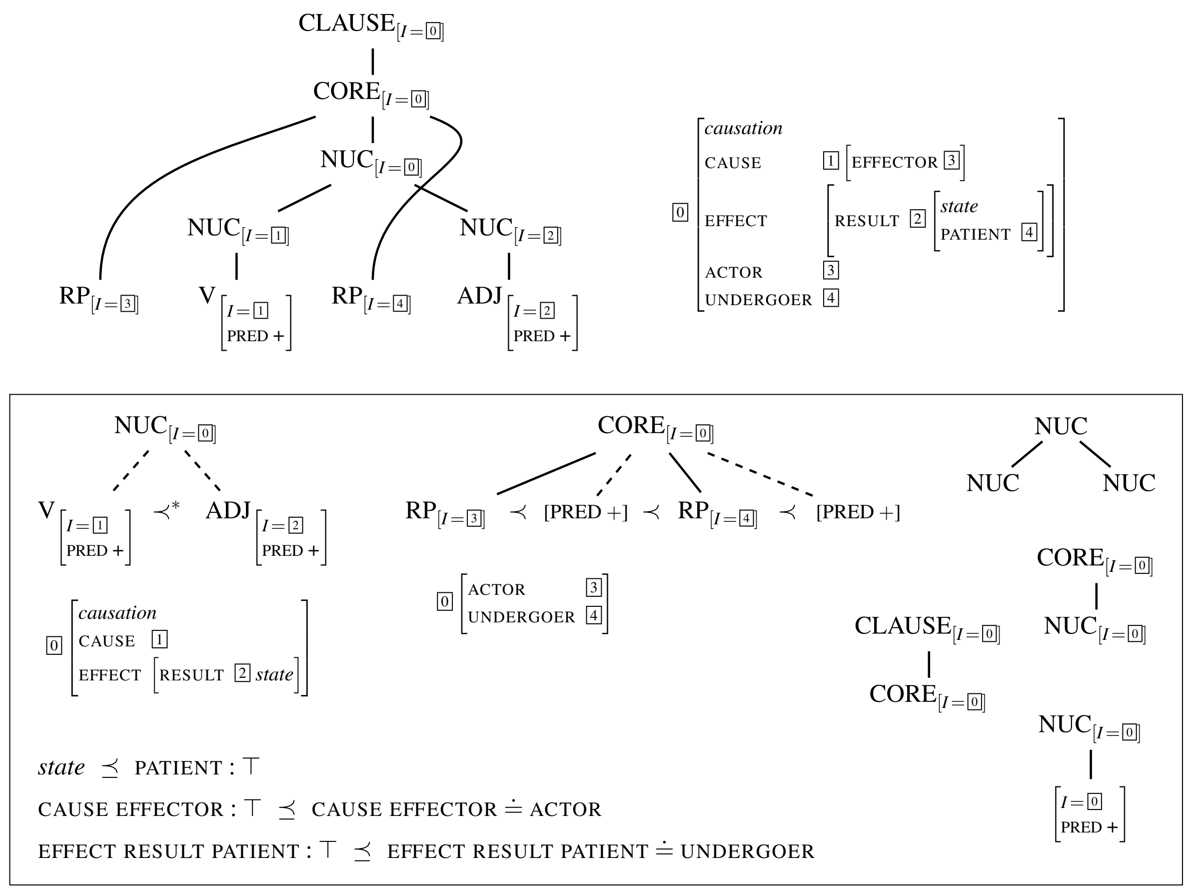


Fig. 16: Decomposition of the English adjectival resultative schema

225). The upper row of Figure 16 shows the corresponding argument structure template together with its associated causative change of state frame, which is enriched by the assignment of the macroroles. (The *I* feature of a constituent is the referential index that refers to the denoted component in the semantic frame.) We can now try to decompose this template and its associated semantic frame into general and language-specific components. Notice that this has to be done at the level of template and frame specifications since the template in question is an elementary argument structure template and not the result of compositions in the sense of Section 3.2. The lower row of Figure 16 provides a sketch of the possible components involved. On the right, we have specifications of general layer fragments and of the bare nuclear cosubordination structure. The component in the middle of the box specifies a transitive serializing multi-predicate core structure with associated macrorole assignment. (Dashed lines indicate dominance while solid lines stand for immediate dominance.) The specification on the left, which is the most language-specific component, says that a predicative verb followed by a predicative adjective can form a nucleus that has a causative change of state interpretation, where the verb denotes the cause and the adjective denotes the result state. The constraints at the bottom of the box are responsible for the correct assignment of the macroroles.[10] For instance, the second to last constraint says that if there is an effector of a causing event then it is assigned the actor macrorole.

Although a number of details have still to be spelled out, the given example highlights a possible route to formalizing constructional schemas in a way that is compatible with the original motivation for introducing them into the architecture of RRG.

6 Conclusion

In this article, we presented a first outline of how a formalization of RRG could look like which attempts to give full consideration to all aspects of the theory. We proposed two additional desiderata with respect to the current organization of RRG: A formalization should draw a clearer distinction between declarative and procedural elements and it should put more emphasis on syntactic and semantic compositionality. The larger part of the article was concerned with the formalization of the syntactic side of RRG. To this end, we introduced three modes of composition for syntactic structures: (simple) substitution, (sister) adjunction, and wrapping (substitution). In particular, wrapping substitution has turned out to be an appropriate mode for composing embedded control and matrix-coding constructions, which have not drawn much attention in the RRG literature before. For formalizing the semantic side of RRG, we proposed the use of decompositional frames. The English resultative construction has served as an example for showing how semantic frames can be integrated with syntactic templates in a flexible way. Of course, many issues remain to be explored. For instance, the formal representation of information structure has been completely neglected in this article.

Lack of space has prevented us from saying more about how the formalism developed so far can be employed for parsing. We refer the reader to Kallmeyer *et al.* (2013) for a formal parsing scheme developed for a slightly simplified version of the above modes of composition. Moreover, Lichte & Petitjean describe ongoing work on adding a frame-semantic component to the XMG grammar development system (Crabbé *et al.* 2013), which can be seen as a first step towards the kind of grammar workbench envisaged in the introduction.

References

Crabbé, Benoit & Duchier, Denys. 2005. Metagrammar redux. In *Constraint Solving and Language Processing*. H. Christiansen, P. R. Skadhauge & J. Villadsen (eds), Lecture Notes in Computer Science 3438, 32–47. Berlin: Springer.

[10] See Kallmeyer & Osswald (2013: 287ff.) for a thorough introduction of the formal notations.

Crabbé, Benoit, Duchier, Denys, Gardent, Claire, Le Roux, Joseph & Parmentier, Yannick. 2013. XMG: eXtensible MetaGrammar. *Computational Linguistics*, 39(3): 591–629.

Dowty, David. 1979. *Word Meaning and Montague Grammar*. Dordrecht: D. Reidel.

Foley, William A. & Van Valin, Robert D., Jr. 1984. *Functional Syntax and Universal Grammar*. Cambridge: CUP.

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

Ginzburg, Jonathan & Sag, Ivan A. 2001. *Interrogative Investigations: The Form, Meaning and Use of English Interrogatives*. Stanford, CA: CSLI Publications.

Guest, Elizabeth. 2008. Parsing for Role and Reference Grammar. In *Investigations of the Syntax-Semantics-Pragmatics Interface.* R. D. Van Valin Jr. (ed.), 435–454. Amsterdam: John Benjamins.

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

Joshi, Aravind K., Kallmeyer, Laura & Romero, Maribel. 2008. Flexible composition in LTAG: Quantifier scope and inverse linking. In *Computing Meaning*. H. Bunt & R. Muskens (eds), vol. 3, 233–256. Berlin: Springer.

Joshi, Aravind K. & Schabes, Yves. 1997. Tree-adjoining grammars. In *Handbook of Formal Languages. Vol. 3: Beyond Words*. G. Rozenberg & A. Salomaa (eds), 69–123. Berlin: Springer.

Kallmeyer, Laura & Osswald, Rainer. 2013. Syntax-driven semantic frame composition in Lexicalized Tree Adjoining Grammars. *Journal of Language Modelling*, 1(2): 267–330.

Kallmeyer, Laura, Osswald, Rainer & Van Valin, Robert D., Jr. 2013. Tree wrapping for Role and Reference Grammar. In *Formal Grammar (FG 2012/2013)*. G. Morrill & M.-J. Nederhof (eds), Lecture Notes in Computer Science 8036, 175–190. Springer.

Kowalski, Robert. 1979. Algorithm = logic + control. *Communications of the ACM*, 22(7): 424–436.

Kroch, Anthony. 1987. Unbounded dependencies and subjacency in a Tree Adjoining Grammar. In *The Mathematics of Language*. A. Manaster-Ramer (ed.), 134–172. Philadelphia: John Benjamins.

Lichte, Timm & Petitjean, Simon. 2015. Implementing semantic frames as typed feature structures with XMG. *Journal of Language Modelling*, 3(1): 185–228.

Nolan, Brian. 2004. First steps toward a computational RRG. In *Proceedings of the International Role and Reference Grammar Conference 2004*. B. Nolan (ed.), 196–223. Dublin: Institute of Technology Blanchardstown.

Osswald, Rainer & Van Valin, Robert D., Jr. 2014. FrameNet, frame structure, and the syntax-semantics interface. In *Frames and Concept Types*. T. Gamerschlag, D. Gerland, R. Osswald & W. Petersen (eds), 125–156. Dordrecht: Springer.

Parsons, Terence. 1990. *Events in the Semantics of English.* Cambridge, MA: MIT Press.

Periñán, Carlos & Mairal, Ricardo. 2009. Bringing Role and Reference Grammar to natural language understanding. *Procesamiento del Lenguaje Natural*, 43: 265–273.

Pollard, Carl J. & Sag, Ivan A. 1994. *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press.

Rambow, Owen, Vijay-Shanker, K. & Weir, David. 1995. D-tree grammars. In *Proceedings of the 33rd Annual Meeting*, 151–158. Association for Computational Linguistics.

Sag, Ivan A. & Wasow, Thomas. 1999. *Syntactic Theory: A Formal Introduction.* Stanford, CA: CSLI Publications.

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

—. 2006. Semantic macroroles and language processing. In *Semantic Role Universals and Argument Linking. Theoretical, Typological, and Psycholinguistic Perspectives*. I. Bornkessel, M. Schlesewsky, B. Comrie & A. D. Friederici (eds), 263–301. Berlin: Mouton de Gruyter.

—. 2010. Role and Reference Grammar as a framework for linguistic analysis. In *The Oxford Handbook of Linguistic Analysis*. B. Heine & H. Narrog (eds), 703–738. Oxford: OUP.

Van Valin, Robert D., Jr. & LaPolla, Randy J. 1997. *Syntax: Structure, Meaning, and Function*. Cambridge: CUP.