A frame-based semantics of the dative alternation in Lexicalized Tree Adjoining Grammars

Laura Kallmeyer and Rainer Osswald*
Heinrich-Heine-Universität Düsseldorf

1. Introduction

It is well-known that the meaning of a verb-based construction does not only depend on the lexical meaning of the verb but also on its specific syntagmatic environment. Lexical meaning interacts with constructional meaning in intricate ways and this interaction is crucial for theories of argument linking and the syntax-semantics interface. These insights have led proponents of Construction Grammar to treating every linguistic expression as a construction (Goldberg 1995). But the influence of the syntagmatic context on the constitution of verb meaning has also been taken into account by lexicalist approaches to argument realization (e.g. Van Valin & LaPolla 1997). The crucial question for any theory of the syntax-semantic interface is how the meaning components are distributed over the lexical and morphosyntactic units of a linguistic expression and how these components combine. In this paper, we describe a grammar model that is sufficiently flexible with respect to the factorization and combination of lexical and constructional units both on the syntactic and the semantic level.

The proposed grammar description framework combines *Lexicalized Tree Adjoining Grammars* (LTAG) with *decompositional frame semantics* and makes use of a constraint-based, 'metagrammatical' specification of the elementary syntactic and semantic structures. The LTAG formalism has the following two key properties (Joshi & Schabes 1997): (i) *Extended domain of locality*: The full argument projection of a lexical item can be represented by a single elementary tree. The domain of locality with respect to dependency is thus larger in LTAG than in grammars based on context-free rules. Elementary trees can have a complex constituent structure. (ii) *Factoring recursion from the domain of dependencies*: Constructions related to iteration and recursion are modeled by the operation of adjunction. Examples are attributive and adverbial modification. Through adjunction, the local dependencies encoded by elementary trees can become long-distance dependencies in the derived trees.

Bangalore & Joshi (2010) subsume the properties (i) and (ii) under the slogan 'complicate locally, simplify globally.' The idea is that basically all linguistic constraints are specified over the local domains represented by elementary trees and, as a consequence, the composition of elementary trees can be expressed by the two general operations substitution and adjunction. This view on the architecture of grammar, which underlies LTAG, has direct consequences for

*The research presented here has been supported by the Collaborative Research Center 991 funded by the German Research Foundation (DFG). We would like to thank Stefan Müller and the reviewers, especially Chris Piñón, for their comments on this paper.

Empirical Issues in Syntax and Semantics 9, ed. Christopher Piñón, pp. 167–184
EISS 9 http://www.cssp.cnrs.fr/eiss9/

semantic representation and computation. Since elementary trees are the basic syntactic building blocks, it is possible to assign complex semantic representations to them without necessarily deriving these representations compositionally from smaller parts of the tree. Hence, there is no need to reproduce the internal structure of an elementary syntactic tree within its associated semantic representation (Kallmeyer & Joshi 2003). In particular, one can employ 'flat' semantic representations along the lines of Copestake et al. (2005). This approach, which supports the underspecified representation of scope ambiguities, has been taken up in LTAG models of quantifier scope and adjunction phenomena (Kallmeyer & Joshi 2003; Gardent & Kallmeyer 2003; Kallmeyer & Romero 2008).

The fact that elementary trees can directly be combined with semantic representations allows for a straightforward treatment of idiomatic expressions and other non-compositional phenomena, much in the way proposed in Construction Grammar. The downside of this 'complicate locally' perspective is that it is more or less unconcerned about the nature of the linguistic constraints encoded by elementary trees and about their underlying regularities. In fact, a good part of the linguistic investigations of the syntax-semantics interface are concerned with argument realization, including argument extension and alternation phenomena (e.g. Van Valin 2005; Levin & Rappaport Hovav 2005; Müller 2006). Simply enumerating all possible realization patterns in terms of elementary trees without exploring the underlying universal and language-specific regularities would be rather unsatisfying from a linguistic point of view.

The mere enumeration of basic constructional patterns is also problematic from the practical perspective of grammar engineering (Xia et al. 2010): The lack of generalization gives rise to redundancy since the components shared by different elementary trees are not recognized as such. This leads to maintenance issues and increases the danger of inconsistencies. A common strategy to overcome these problems is to introduce a tree description language which allows one to specify sets of elementary trees in a systematic and non-redundant way (e.g. Candito 1999; Xia 2001). The linguistic regularities and generalizations of natural languages are then captured on the level of descriptions. Since LTAG regards elementary trees as the basic components of grammar, the system of tree descriptions is often referred to as the *metagrammar*. While the details of the approaches of Candito (1999) and Xia (2001) differ, they both assume canonical or base trees from which alternative constructions are derived by a system of lexical and syntactic rules. Crabbé (2005), by contrast, proposes a purely constraint-based approach to metagrammatical specification (see also Crabbé & Duchier 2005), which does not presume a principle distinction between canonical and derived patterns but generates elementary trees uniformly as minimal models of metagrammatical descriptions. We will adopt the latter approach for our framework because of its clear-cut distinction between the declarative level of grammatical specification and procedural and algorithmic aspects related to the generation of the elementary trees.

Existing metagrammatical approaches in LTAG are primarily concerned with the organization of general valency templates and with syntactic phenomena such as passivization and wh-extraction. The semantic side has not been given much attention up to now. However, there are also important semantic regularities and generalizations to be captured within the domain of elementary constructions. In addition to general semantic constraints on the realization of arguments, this includes also the more specific semantic conditions and effects that go along with argument extension and modification constructions such as resultative and applicative constructions, among others. In order to capture phenomena of this type, the metagrammatical descriptions need to include semantic constraints as well. In other words, analyzing the syntax-semantics interface given by elementary constructions that goes beyond the mere enumeration

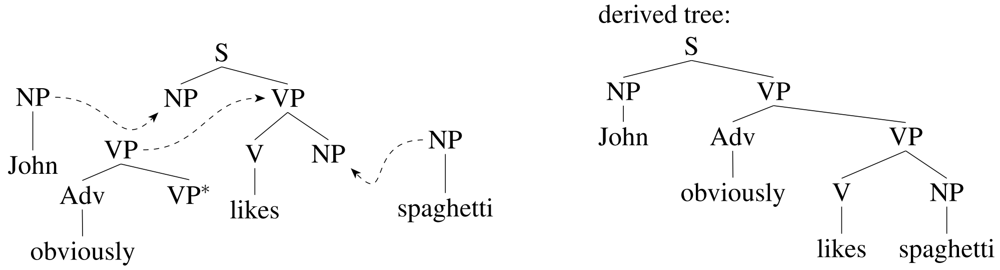

Figure 1: A sample derivation in TAG

of form-meaning pairs calls for a (meta)grammatical system of constraints consisting of, both, syntactic and semantic components. It must be emphasized that this conclusion does not imply a revival of the idea of a direct correspondence between syntactic and semantic (sub)structures, an assumption which LTAG has abandoned for good reasons.

The framework proposed in this paper treats the syntactic and the semantic components of elementary constructions as structured entities, trees, on the hand, and frames, on the other hand, without requiring that there be any structural isomorphism between them. The metagrammar specifies the syntactic and semantic properties of constructional fragments and defines how they can combine to larger constructional fragments. There is no need for a structural isomorphism between syntax and semantics simply because the relation between the syntactic and semantic components is explicitly specified. Below we illustrate this program of decomposing syntactic trees and semantic frames in the metagrammar by a case study of the dative alternation in English, which is is well-known to be sensitive to lexical and constructional meaning components.

A long-term goal of the work described in this paper is the development of a grammar engineering framework that allows a seamless integration of lexical and constructional semantics. More specifically, the approach provides Tree Adjoining Grammars with a decompositional lexical and constructional semantics and thereby complements existing proposals which are focused on standard sentence semantics. From a wider perspective, the framework can be seen as a step towards a formal and computational account of some key ideas of Construction Grammar à la Goldberg, since the elementary trees of LTAG combined with semantic frames come close to what is regarded as a construction in such approaches. Frameworks with similar goals are Embodied Construction Grammar (Bergen & Chang 2005) and Sign-Based Construction Grammar (Sag 2012).

2. LTAG and grammatical factorization
2.1. Brief introduction to TAG

Tree Adjoining Grammar (TAG; Joshi & Schabes 1997; Abeillé & Rambow 2000) is a tree-rewriting formalism. A TAG consists of a finite set of *elementary trees*. The nodes of these trees are labelled with non-terminal and terminal symbols, with terminals restricted to leaf nodes. Starting from the elementary trees, larger trees are derived by *substitution* (replacing a leaf with a new tree) and *adjunction* (replacing an internal node with a new tree). Sample elementary trees and a derivation are shown in Fig. 1. In this derivation, the elementary trees for *John* and *spaghetti* substitute into the subject and the object slot of the elementary tree for *likes*, and the *obviously* modifier tree adjoins to the VP node.

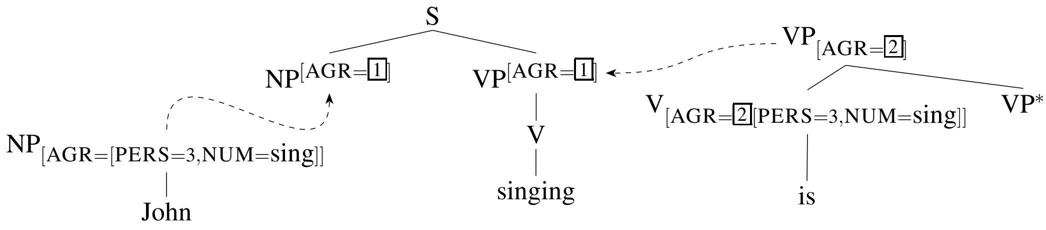

Figure 2: Agreement in FTAG

In case of an adjunction, the tree being adjoined has exactly one leaf that is marked as the *foot node* (marked with an asterisk). Such a tree is called an *auxiliary* tree. To license its adjunction to a node *n*, the root and foot nodes must have the same label as *n*. When adjoining it to *n*, in the resulting tree, the subtree with root *n* from the old tree is attached to the foot node of the auxiliary tree. Non-auxiliary elementary trees are called *initial* trees. A derivation starts with an initial tree. In a final derived tree, all leaves must have terminal labels. In a TAG, one can specify for each node whether adjunction is mandatory and which trees can be adjoined. The subscripts *NA* and *OA* indicate adjunction constraints: *NA* signifies that for this node, adjunction is not allowed while *OA* signifies that adjunction is obligatory.

In order to capture syntactic generalizations in a more satisfying way, the non-terminal node labels in TAG elementary trees are usually enriched with feature structures. The resulting TAG variant is called *feature-structure based TAG* (FTAG; Vijay-Shanker & Joshi 1988). In an FTAG, each node has a top and a bottom feature structure (except substitution nodes that have only a top). Nodes in the same elementary tree can share features. In an FTAG, adjunction constraints are expressed via the feature structures. During substitution and adjunction, the following unifications take place. In a substitution operation, the top of the root of the new initial tree unifies with the top of the substitution node. In an adjunction operation, the top of the root of the new auxiliary tree unifies with the top of the adjunction site and the bottom of the foot of the new tree unifies with the bottom of the adjunction site. Furthermore, in the final derived tree, top and bottom must unify for all nodes. Since nodes in the same elementary tree can share features, constraints among dependent nodes can be more easily expressed than in the original TAG formalism. Fig. 2 shows an example where the top feature structure is notated as a superscript and the bottom feature structure as a subscript of the respective node.

2.2. LTAG elementary trees

The elementary trees of a TAG for natural languages are subject to certain principles (Frank 2002; Abeillé 2002). Firstly, they are lexicalized in that each elementary tree has at least one lexical item, its *lexical anchor*. A *lexicalized* TAG (LTAG) is a TAG that satisfies this condition for every elementary tree. Secondly, each elementary tree associated with a predicate contains 'slots', that is, leaves with non-terminal labels (substitution nodes or foot nodes) for all and only the arguments of the predicate (*elementary tree minimality*). Most argument slots are substitution nodes, in particular the nodes for nominal arguments (see the elementary tree for *likes* in Fig. 1). Sentential arguments are realised by foot nodes in order to allow long-distance dependency constructions such as *Whom does Paul think that Mary likes?*. Such extractions can be obtained by adjoining the embedding clause into the sentential argument (Kroch 1989; Frank 2002).

As for semantic representation and the syntax-semantics interface, we basically build on

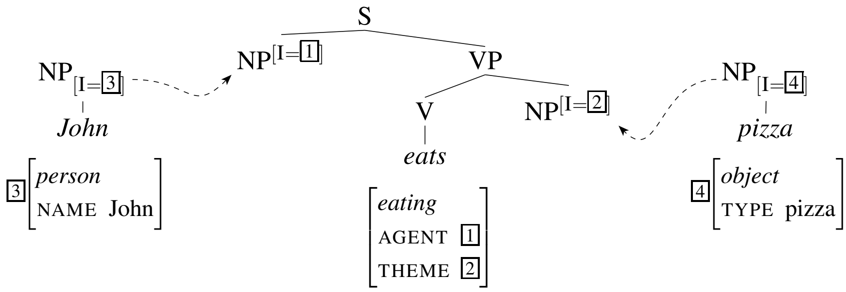

Figure 3: Syntactic and semantic composition for *John eats pizza*

approaches which link a semantic representation to an entire elementary tree (cf. §1) and which model composition by unifications triggered by substitution and adjunction. For example, in Gardent & Kallmeyer (2003) (see also Kallmeyer & Joshi 2003, Kallmeyer & Romero 2008), every elementary tree is paired with a set of typed predicate logical formulas containing metavariables linked to features in FTAG structures. The syntactic composition then triggers unifications that lead to equations between semantic components.

The focus of the present paper is on a decompositional frame semantics for elementary LTAG trees. Fig. 3 illustrates the locality of linking in a frame-semantic approach by a simple example (which does not exploit the decompositional potential of frame representations). The substitutions give rise to unifications between 1 and 3 and between 2 and 4, which leads to an insertion of the corresponding argument frames into the frame of *eats*. Notice that the use of frames does not preclude an approach of the type described above for modeling semantic composition beyond the level of elementary trees, including the effect of logical operators such as quantifiers and other scope taking elements. But the technical details of such a combination remain to be worked out and are beyond the scope of this article.

2.3. Metagrammar and factorization

LTAG allows for a high degree of factorization inside the lexicon, that is, inside the set of lexicalized elementary trees. Firstly, unanchored elementary trees are specified separately from their lexical anchors. The set of unanchored elementary trees is partitioned into *tree families* where each family represents the different realizations of a single subcategorization frame. For transitive verbs such as *hit*, *kiss*, *admire*, etc. there is a tree family (see Fig. 4) containing the patterns for different realizations of the arguments (canonical position, extraction, etc.) in combination with active and passive. The node marked with a diamond is the node that gets filled by the lexical anchor; the 'empty' symbol ε indicates the trace of an extraction.

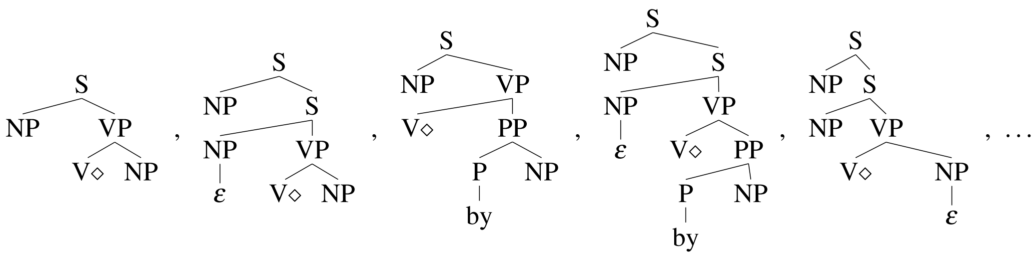

Figure 4: Unanchored tree family for transitive verbs

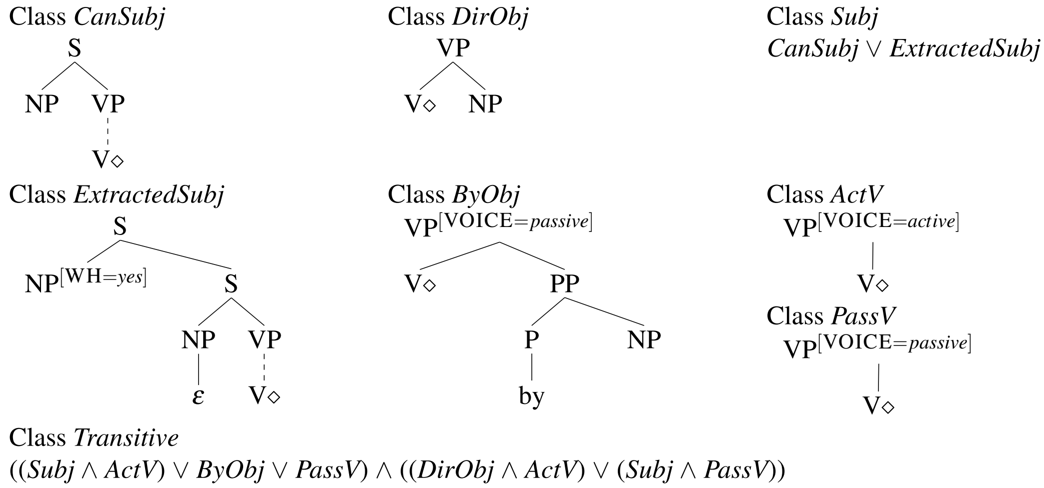

Figure 5: MG fragment for transitive verbs

Secondly, unanchored elementary trees are usually specified by means of a *metagrammar* (Candito 1999; Crabbé & Duchier 2005) which consists of dominance and precedence constraints and category assignments. The elementary trees of the grammar are defined as the *minimal models* of this constraint system. The metagrammar formalism allows for a compact grammar definition and for the formulation of linguistic generalizations. In particular, the metagrammatical specification of a subcategorization frame defines the set of all unanchored elementary trees that realize this frame. Moreover, the formalism allows us to define tree fragments that can be used in different elementary trees and tree families, thereby giving rise to an additional factorization and linguistic generalization. Phenomena that are shared between different tree families such as passivization or the extraction of a subject or an object are specified only once in the metagrammar and these descriptions become part of the descriptions of several tree families.

Let us illustrate this with the small metagrammar fragment given in Fig. 5, which is of course very incomplete in that many tree fragments are missing and features are almost totally omitted. The first two tree fragments describe possible subject realizations: the subject can be in canonical position, immediately preceding the VP, or it can be extracted, with a trace in the canonical subject position. The class *Subj* comprises the different subject realizations. Similar classes exist for the different realizations of the object, while in Fig. 5 only the canonical position class is listed. Furthermore, there is a class for the *by*-PP in a passive construction. This is used only for passive, therefore the tree fragment contains a corresponding feature VOICE = *passive*. Besides these argument classes, our fragment contains two classes for active/passive morphology. Finally, the class *Transitive* specifies for each argument its different grammatical functions: the first argument can be the subject of an active sentence or the *by*-PP of a passive sentence or it can be omitted in a passive sentence.[1] The second argument can be the direct object or it can be promoted to a subject in a passive sentence. If we assume that the metagrammar constraints require the identification of the lexical anchor nodes, then the set of minimal models of this class includes the first four trees in Fig. 4, among others. Note that the difference between canonical subject and extracted subject is factored out in the class *Subj*.

[1]We are computing minimal models, this is why the third possibility in the disjunction signifies that this argument is not realized.

A similar factorization is possible within the semantics. The semantic contribution of unanchored elementary trees, that is, constructions, can be separated from their lexicalization, and the meaning of a construction can be decomposed further into the meaning of fragments of the construction. Due to this factorization, relations between the different parts of a syntactic construction and the components of a semantic representation can be expressed. In the following, we will use the metagrammar factorization of elementary trees in order to decompose the semantics of double object and prepositional object constructions.

3. Frame-based semantics and the dative alternation

3.1. Frame semantics and lexical decomposition

The program of Frame Semantics initiated by Fillmore (1982) aims at capturing the meaning of lexical items in terms of *frames*, which are to be understood as cognitive structures that represent the described situations or state of affairs. In their most basic form, frames specify the type of a situation and the semantic roles of the participants, that is, they correspond to feature structures of the kind used in Fig. 3 for representing eating situations. Frame semantics as currently implemented in the FrameNet project (Fillmore et al. 2003) basically builds on such plain role frames, and it is a central goal of FrameNet to record on a broad empirical basis how the semantic roles are expressed in the morphosyntactic environment of the frame evoking word.

In contrast to pure semantic role approaches to argument realization, many current theories of the syntax-semantics interface are based on predicate decomposition and event structure analysis (cf. Levin & Rappaport Hovav 2005). These theories assume that the morphosyntactic realization of an argument depends crucially on the structural position of the argument within the decomposition. Two simple notational variants of such a decomposition of the causative verb *break* are shown in (1), formulated along the lines of Van Valin & LaPolla (1997) and Rappaport Hovav & Levin (1998).

(1) [[*x* ACT] CAUSE [BECOME [*y BROKEN*]]]

With respect to the goals of our project, a decompositional semantic representation is the natural choice since it allows us to associate specific components of the semantic representation with specific syntactic fragments. We integrate event structure decomposition with frame semantics.[2] That is, we use frames, understood as potentially nested typed feature structures with additional constraints, for representing decompositional templates of the sort shown in (1). Fig. 6a shows a fairly direct translation of this template into a frame representation. Note the different uses of CAUSE in (1) and Fig. 6. In (1), CAUSE expresses the causation relation between the activity and the change of state. In the frame representation, by contrast, the attribute CAUSE describes the 'cause component' of the causation scenario. The graph on the right of Fig. 6 can be regarded either as an equivalent presentation of the frame, or as a minimal model of the structure on the left if the latter is seen as a frame description. It is worth mentioning that there is also a fairly close correspondence of decompositional frame representations to event logical formulas in a neo-Davidsonian style. For if each subframe is interpreted as representing a reified subcomponent of the described event, then the structure shown in Fig. 6 gives rise to a formula like (2),

[2]Koenig & Davis (2006), who make a similar proposal, put emphasis on the fact that the part of the frame relevant for argument linking can be a proper subframe of the semantic representation associated with the expression in question. That is, the 'referential node' of the frame need not coincide with the root of the frame. While we do not exploit this possibility in our analysis, we do not exclude it in principle.

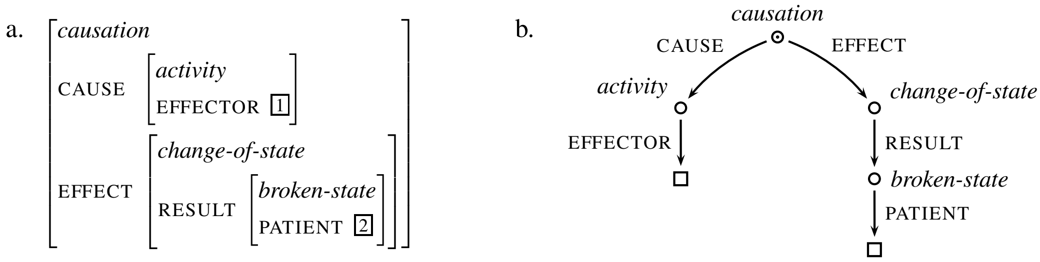

Figure 6: Possible frame representation for template (1)

in which the activity event e' is the cause component of the event e, and so on.

(2) $\exists e \exists e' \exists e'' \exists s\, [\textit{causation}(e) \wedge \textsc{cause}(e,e') \wedge \textsc{effect}(e,e'') \wedge \textit{activity}(e') \wedge \textsc{effector}(e',x) \wedge \textit{change-of-state}(e'') \wedge \textsc{result}(e'',s) \wedge \textit{broken-state}(s) \wedge \textsc{patient}(s,y)]$

Frames allow us to combine two key aspects of template-based decompositions and of logical representations: Like decompositional schemas they are concept-centered and have inherent structural properties and like logical representations they are flexible and can be easily extended by additional subcomponents and constraints.

3.2. Semantic properties of the dative alternation

The English dative alternation is concerned with verbs like *give*, *send*, and *throw* which can occur in both the double object (DO) and the prepositional object (PO) construction as exemplified by (3).

(3) a. John sent Mary the book.
b. John sent the book to Mary.

The two constructions are traditionally associated with a ‘caused possession’ (3a) and ‘caused motion’ (3b) interpretation, respectively. These two interpretations have often been analyzed by decompositional schemas of the type shown in (4a) and (4b), respectively.

(4) a. [[x ACT] CAUSE [y HAVE z]]
b. [[x ACT] CAUSE [z GO TO y]]

In a similar vein, Krifka (2004) uses event logical expressions of the sort shown in (5) for distinguishing the two interpretations.

(5) a. $\exists e \exists s[\mathrm{AGENT}(e,x) \wedge \mathrm{CAUSE}(e,s) \wedge s : \mathrm{HAVE}(y,z)]$
b. $\exists e \exists e'[\mathrm{AGENT}(e,x) \wedge \mathrm{CAUSE}(e,e') \wedge \mathrm{MOVE}(e') \wedge \mathrm{THEME}(e',y) \wedge \mathrm{GOAL}(e',z)]$

Following the general outline sketched in the previous section, (5b) could be translated into the frame representation shown in Fig. 7a. Version 7b, by comparison, is closer to template (4b) if we take [x ACT] to represent the activity subcomponent of the caused motion event. Frame 7c is a further variant based on the caused motion schema (6) taken from Van Valin & LaPolla (1997). In comparison with the first two frame versions, this representation tries to make explicit the resulting change of location of the theme.

(6) $[\mathbf{do}(x,\emptyset)]$ CAUSE [BECOME $\mathbf{be\text{-}at}(y,z)$]

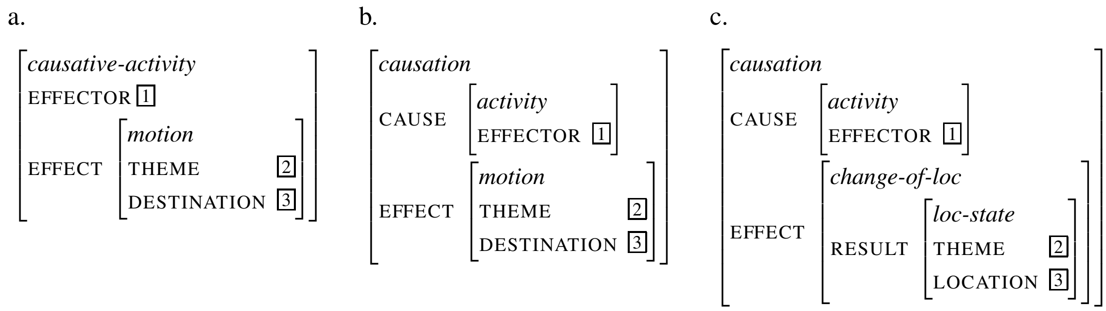

Figure 7: Some frame representation options for caused motion

The difference between the DO and the PO variant and their respective interpretations has been observed to span a wider range of options than those described so far. Rappaport Hovav & Levin (2008) distinguish three types of alternating verbs based on differences in the meaning components they lexicalize: *give*-type (*lend*, *pass*, etc.), *send*-type (*mail*, *ship*, etc.), and *throw*-type verbs (*kick*, *toss*, etc.).[3] They provide evidence that verbs like *give* have a caused possession meaning in both kinds of constructions. The *send* and *throw* verbs, by comparison, lexically entail a change of location and allow both interpretations depending on the construction they occur in. The *send* and *throw* verbs differ in the meaning components they lexicalize: *send* lexicalizes caused motion towards a destination, whereas *throw* encodes the caused initiation of motion and the manner in which this is done. A destination is not lexicalized by *throw* verbs, which accounts for the larger range of directional PPs allowed for these verbs.

Beavers (2011) proposes a formally more explicit explanation of these observations based on a detailed analysis of the different types of results that determine the aspectual behavior of the verbs in question. He identifies four main types of results for ditransitive verbs: loss of possession, possession, leaving, and arrival. These results are associated with two different dimensions or 'scales': The first two results belong to the 'possession scale'; the latter two results are associated with a location or path scale. Only *give* verbs lexicalize actual possession as a result. *Send* verbs and *throw* verbs, by contrast, do not encode actual possession nor do they encode prospective possession when combined with the PO construction. The result condition that makes these verbs telic even if the theme does not arrive at the destination or recipient is the leaving of the theme from the actor. That is, the aspectually relevant result consists in leaving the initial point of the underlying path scale.

With respect to the goals of the present study, the main question is how the constructional meaning interacts with the lexical meaning. The DO construction encodes only *prospective* possession. Actual possession must be contributed by the lexical semantics of the verb. This is the case for *give* verbs, which explains why there is no difference between the DO and the PO constructions for these verbs as far as caused possession is concerned. All other alternating ditransitive verbs show such a difference since only the DO pattern implies prospective possession.[4] Beavers (2011) draws a distinction between different types of caused possession verbs. Verbs such as *give* encode pure caused possession without necessarily motion involved. Verbs

[3] For simplicity, we do not consider verbs of communication (*tell*, *show*, etc.) nor do we take into account differences in modality as between *give* and *offer* (Koenig & Davis 2001).

[4] The story is a bit more complicated: If the destination of the PO construction is human or human-like (e.g. an institution), there seems to be a conventional implicature that the (prospective) destination is also a (prospective) recipient, that is, (prospective) possession seems to be entailed in cases like *send the package to London*.

	lexical meaning					PO pattern	DO pattern
	#args	result	punct.	manner	motion	(◇arrive)	(◇receive)
give	3	receive	yes	no	no	receive (arrive)	receive
hand	3	receive	yes	yes	yes	receive (arrive)	receive
send	3	leave ◇arrive	yes	no	yes	◇arrive	◇receive
throw	2	leave	yes	yes	yes	◇arrive	◇receive
bring	3	arrive	no	no	yes	arrive	receive

Table 1: Semantic classes of verbs in interaction with the DO and PO patterns

like *hand* and *pass*, by comparison, imply actual possession but also arrival of the theme via motion. The possession scale is 'two-point' or 'simplex' in that its only values are non-possession and possession. It follows that verbs which lexicalize caused possession are necessarily punctual since there are no intermediate 'points' on this scale. In contrast to *send* and *throw*, verbs like *bring* and *take* do encode arrival of the theme at the destination (Beavers 2011). That is, for these verbs of accompanied motion, the arrival is actual and not only prospective, and this property can be regarded as lexicalized since the verbs in question are basically three-place predicates. Verbs like *carry* and *pull*, which lexicalize a 'continuous imparting of force', behave differently (Krifka 2004). They are basically two-argument verbs, that is, they do not lexicalize a destination, and they are usually regarded as being incompatible with the DO pattern.[5]

In sum, the DO and PO constructions strongly interact with the lexical semantics of the verb.[6] Table 1, which builds on Beavers' analysis, gives an overview of the contribution of the lexicon and the constructions. Prospectivity is indicated by '◇'. For some of the verbs listed in the table, possible frame semantic representations are given in Fig. 8. Consider the frame for *send*. The change of location subframe is meant to encode motion towards the destination without necessarily implying arrival. Actual arrival would be encoded by a resulting location state as in Fig. 7c, that is, in analogy to the representation of actual possession in the entry for *give*. The representation for *throw* differs from that for *send* in that *throw* lexicalizes a certain type of activity, here simply encoded by a subtype *throw-activity* of *activity*. Moreover, it is inherent in the given representation that the destination of the entity thrown is not part of the lexical meaning of *throw*.[7]

[5]Krifka (2004) explains this fact by pointing out that the continuous imparting of force is a 'manner' component that is not compatible with a caused possession interpretation. The strict exclusion of the DO pattern for verbs accompanied motion like *carry* has been called into question by Bresnan & Nikitina (2010) on the basis of corpus evidence. Building on Krifka's approach, Beavers (2011:46f) explains the low frequency of the DO pattern by distinguishing between the different kinds of 'have' relations involved: the 'have' of control by the actor during the imparting of force and the final 'have' of possession by the recipient. He proposes a 'naturalness constraint' which largely, but not totally, excludes caused possession in cases where the actor has control on the theme at the final point of the event. Conditions of this type would naturally go into a detailed frame-semantic analysis elaborating on the ones given in this paper.

[6]The DO construction with caused possession interpretation also occurs for creation verbs with a benefactive extension as in *bake her a cake*. The PO pattern requires a *for*-PP in theses cases, which will not be taken into account in the following.

[7]In Fig. 8, there is no indication of the different types of causation involved. For instance, the initiation of motion encoded by *throw* could be captured by a subtype *onset-causation* of *causation*; cf. Kallmeyer & Osswald 2012.

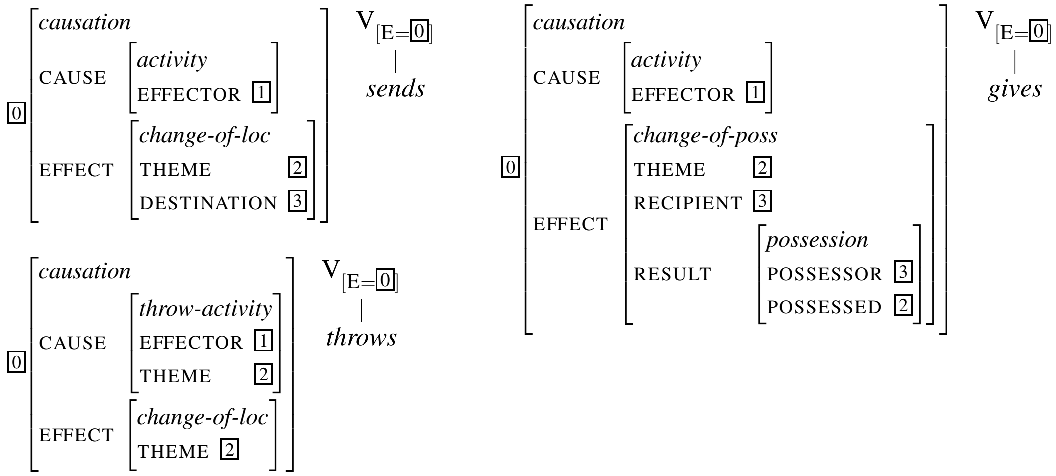

Figure 8: Possible frame representations for some of the lexical items in Table 1

It goes without saying that a full account of the dative alternation has to cope with a lot more phenomena than the distinction between caused motion and caused possession interpretations and their sensitivity to the lexical semantics of the head verb. The distribution of the DO and PO variants of the alternation is known to be influenced by various other factors, including discourse structure effects, heaviness constraints, and the definiteness, pronominality, and animacy of recipient and theme (cf. Bresnan & Ford 2010). Correspondingly, a full grammar model would have an information structure component, ordering constraints which are sensitive to constituent length, and so on, and, in addition, would allow for defeasible and probabilistic constraints. While we think that our grammar framework is well-suited for implementing requirements of this sort, they are beyond the scope of the present study, which is primarily concerned with modeling the influence of narrow verb classes on constructional form and meaning.

4. Analysis of DO versus PO constructions

Modeling the data above in our approach calls for a sufficiently detailed decomposition of the semantics of verbs and constructions using frames represented as typed feature structures. Moreover, the semantic frames and their subcomponents are to be associated with morphosyntactic trees and tree fragments. Note that this paper does not deal with idiomatic expressions like *give somebody the creeps*. Such expressions are not decomposed into a DO construction meaning and a lexical item meaning. Instead, idiomatic lexicalized elementary trees have multiple anchors, here *give* and *creeps*, and they can be associated with specific meanings (cf. Abeillé & Schabes 1989).

4.1. Unanchored elementary trees

Concerning the form of the syntactic elementary trees, we partly follow the choices made in the TAG grammar of English developed by the XTAG group (XTAG Research Group 2001). This grammar employs a tree family for ditransitive verbs with two NPs and a tree family for verbs selecting for an object NP and a PP. In the PO construction we are interested in, the PP has to be a directional PP. Some verbs are more restricted than others concerning the choice of the preposition due to the interplay of the properties of the event participant determined by the

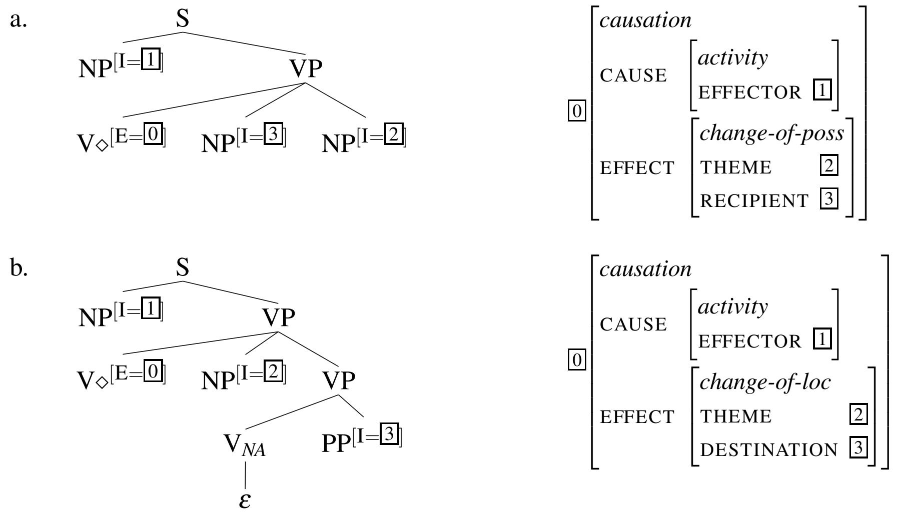

Figure 9: Unanchored elementary trees and semantics of the DO and PO constructions

verb and the properties determined by the preposition. For instance, if the verb *give* is combined with an *into*-PP, then the meaning of the preposition implies that the NP embedded in the PP has to be some kind of container, which is difficult to reconcile with a change-of-possession interpretation of the verb. We leave the exact frame-based modeling of such restrictions for future research. Notice also that there are PO constructions in which a specific preposition is treated as a coanchor of the elementary tree. This is the case for phrasal verbs like *believe in*, where the preposition is lexicalized with the verb and does not add any separate semantic information.

The base trees of the DO and PO families are depicted on the left side of Fig. 9. The lower VP node in the PO tree is inspired by the XTAG proposal and allows for the adjunction of adverbial modifiers between the direct object and the PP object.[8] The semantics of the DO construction is a (prospective) caused possession meaning which gets further constrained when linking it to a specific lexical anchor. Fig. 9a shows how the unanchored tree is linked to its semantic frame. The identities between the I features in the syntactic tree (which keep track of the denoted individuals) and the thematic roles in the semantic frame provide the correct argument linking. The semantics of the PO construction differs in that it triggers a caused motion instead of a caused possession interpretation; see Fig. 9b. The E feature of the V node describes an event; its value is the frame of the elementary tree. When anchoring the tree with a lexical item, this feature unifies with the E feature of the lexical item and thereby guarantees unification of the lexical and the constructional frame.

4.2. Metagrammar decomposition

The unanchored trees for the two constructions and their associated semantic frames can be further decomposed in the metagrammar. Some of the tree fragments in the metagrammar are used by both constructions, some are specific to one of them. In the following, we restrict ourselves to the base trees when explaining the syntactic and semantic decomposition. Of course,

[8]The empty V tree below this additional VP carries a *NA* (null adjunction) constraint, that is, this node does not allow for adjunction.

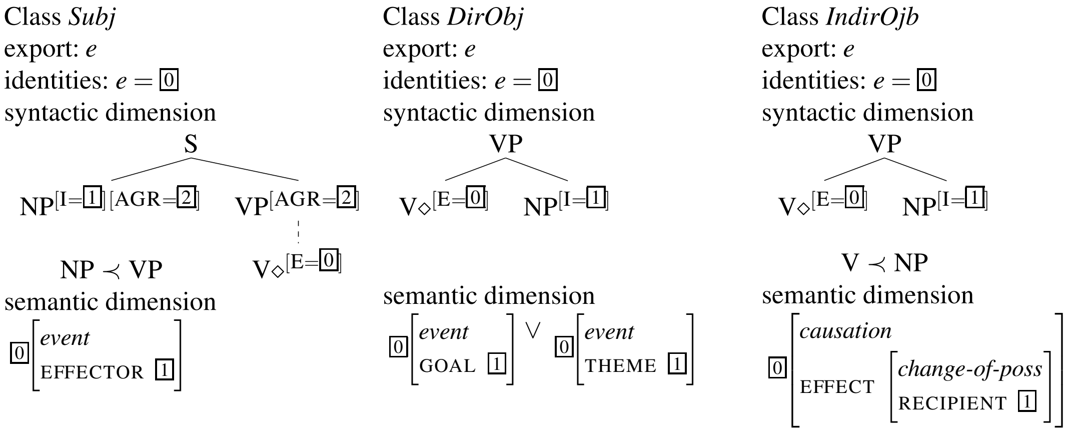

Figure 10: MG classes for subject, direct object, and indirect object

other argument realizations are possible as well and should be taken into account in the metagrammar classes. For instance, the subject NP class *Subj* should not only contain the base subject realization shown on the left of Fig. 10 but also a tree fragment for an extraposed subject, for a wh-extracted subject, for a relativized subject, etc. Some of these tree fragments will contribute different aspects to the semantics. We leave this aside for the moment, since the focus of this paper is on the dative alternation and its semantics. In this paper, we treat only the active base case, assuming that other cases can be captured along the lines sketched in Fig. 5.

Let us first consider the classes needed for the DO construction. Some of the classes are just small tree fragments that do not use any other class. These are, for instance, the ones for the different arguments, namely, for the subject NP, the direct object NP, and the indirect object NP. The first two are fairly general; they occur in many of the elementary trees and do not constrain the semantics. The three argument classes are shown in Fig. 10. Only the roles relevant for our constructions are given, there are of course more possible roles for these arguments. Each class has a name, a declaration of variables that one can refer to when using this class (the export variables), a list of equations, and a syntactic dimension and a semantic dimension. The syntactic dimension contains a tree description that is depicted in the usual way in the figure. That is, solid lines indicate immediate dominance, dotted lines indicate dominance and the order of sisters indicates linear precedence (but not necessarily immediate linear precedence). Furthermore, $\prec$ denotes immediate linear precedence. In the class *Subj*, for instance, the tree description tells us that there are three nodes n_1, n_2, n_3 with labels S, NP, and VP such that n_2 has a top feature I with value $\boxed{1}$. Furthermore, n_1 immediately dominates n_2 and n_3 (depicted by the edges of the tree) and n_2 immediately precedes n_3. The representation is a bit sloppy since it mixes node variables with node categories. The realization of the third argument as an NP (i.e. the use of the class *IndirObj*) is responsible for the caused possession meaning. Therefore this class contributes a frame fragment in its semantics that tells us that the meaning is a causation whose effect is a change of possession where the argument contributed by this class denotes the recipient.[9]

Concerning the semantic dimension, we assume this to be a description of a typed feature structure. When we say 'unification', speaking of combining frames in the metagrammar, we

[9]Again, this is of course not the only way this syntactic fragment can be used; other semantic contributions of indirect objects must be specified in the metagrammar as well.

Class *VSpine*
syntactic dimension

$$\begin{array}{c} \mathrm{VP}_{[\mathrm{AGR}=\boxed{1}]} \\ | \\ \mathrm{V}_{\diamond}{}^{[\mathrm{AGR}=\boxed{1}]} \end{array}$$

Class *n0V*
export: e
use classes V_1 =VSpine, N_1 =Subj
identities: $V_1.V = N_1.V$, $e = N_1.e$

Class *n0Vn1*
export: e
use classes V_1 =n0V
N_2 =DirObj
identities: $V_1.V = N_2.V$, $e = N_2.e$

Class *DOConstr*
use classes V_1 =n0Vn1, N_3 =IndirOjb
identities: $V_1.V = N_3.V$
semantic dimension

$$V_1.e \begin{bmatrix} \textit{causation} & \\ \text{EFFECTOR} & \boxed{1} \\ \text{THEME} & \boxed{2} \\ \text{GOAL} & \boxed{3} \\ \text{CAUSE} & \begin{bmatrix} \textit{activity} & \\ \text{EFFECTOR} & \boxed{1} \end{bmatrix} \\ \text{EFFECT} & \begin{bmatrix} \textit{change-of-poss} & \\ \text{THEME} & \boxed{2} \\ \text{RECIPIENT} & \boxed{3} \end{bmatrix} \end{bmatrix}$$

Figure 11: MG classes for transitive verbs and the DO construction

actually mean conjunction and feature value equation. So far, our impression is that we need only a simple feature logic without quantification or negation.

Now we combine our small tree fragments into larger ones, thereby defining further MG classes. We add a class *VSpine* that takes care of the percolation of features (for instance AGR) along the verbal spine. This class combines with the subject class into the *n0V* class that in turn combines with classes for further arguments. The definition of the class for active transitive verbs is shown in Fig. 11.[10] What is still missing here is a fully elaborate linking theory that determines the possible combinations of semantic roles for a given unanchored syntactic tree, for example, along the lines of Van Valin (2005). We leave this issue for future research.

The further combination with the class for the indirect object is shown in Fig. 11. The minimal model of *DOConstr* is the unanchored tree from Fig. 9a. In addition to the frame shown in Fig. 9a, we include a specification of the thematic roles on the top level of the frame that serves to obtain the correct identifications of participants when unifying with the frame of the lexical anchor. We will come back to this when treating lexical anchoring in §5.

Now let us consider the PO construction. Here, the *n0Vn1* class is used again. For the third argument, we use the class *DirPrepObj* for a directional PP-argument. The PP contributes the goal of some change of location. The higher class *POConstr* arises from a combination of the *n0Vn1* class and the class for the directional PP (Fig. 12). The change of location frame contributed by the PP is embedded under the EFFECT attribute of the frame of the verb and it is enriched with a role THEME that is the event participant contributed by the direct object. Finally, we can define a class *DAltConstr* as the disjunction of *DOConstr* and *POConstr*. This way, we obtain a single tree family containing trees for both constructions. Depending on whether we have a PP or a direct object, only the corresponding part of the family can be selected. The minimal referent of the class *DAltConstr* contains the two trees from Fig. 9.[11]

[10] Note that we assume that whenever we use a class, its meta-variables ($\boxed{0}$, $\boxed{1}$, etc.) get instantiated with fresh values. This avoids unintended unifications.

[11] As mentioned above, the classes corresponding to elementary tree families usually have more than one minimal referent since all possible realizations of an argument (topicalization, extraposition, relativization, etc.) have to be taken into account.

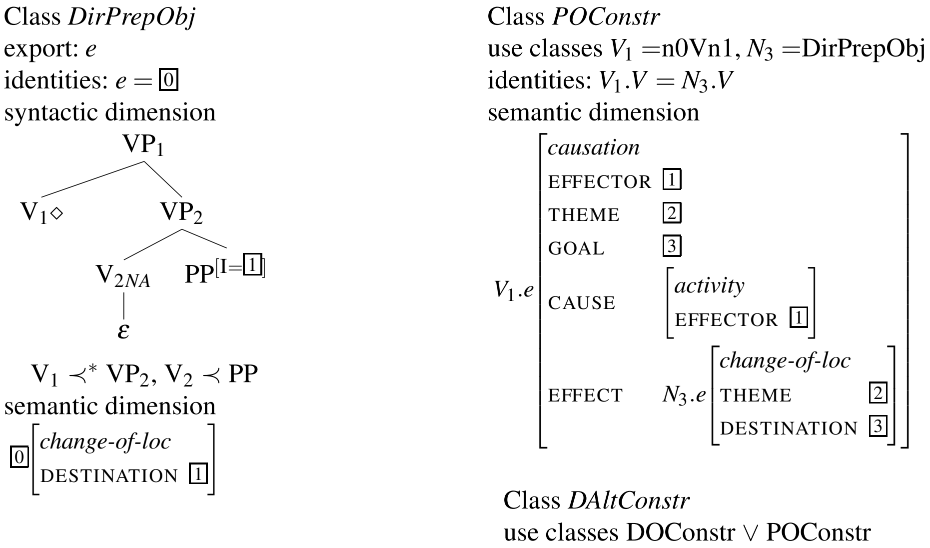

Figure 12: MG classes for the PO construction

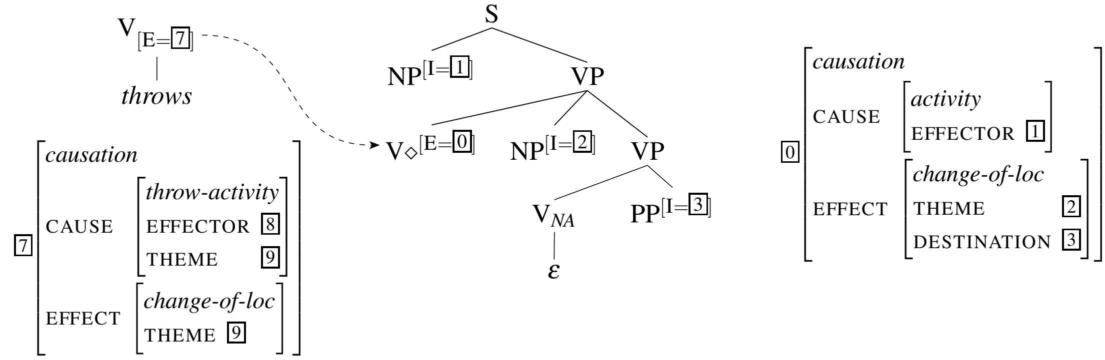

Figure 13: Lexical selection of the elementary tree for *throws* in the PO construction

5. Lexical anchoring for DO and PO constructions

Once the unanchored tree families are computed via compilation of the corresponding MG classes, these trees are anchored by lexical items. The lexical anchor that is substituted into the anchor node contributes parts of a semantic frame (see Fig. 8 above for some lexical items and their semantic frames). Because of the unifications of the syntactic E features on the V nodes, the frames of the unanchored tree and of the lexical anchor unify. The example in Fig. 13 shows the lexical anchoring of the PO construction with the anchor *throws* (with top level roles omitted for reasons of space). The resulting anchored elementary tree has a semantic frame that is the unification of the frames 7 and 0. (Recall that *throw-activity* is a subtype of *activity*.)

The idea is, of course, that if the lexical anchor frame and the construction frame are not compatible with each other, then unification fails. However, in some cases where standard unification fails we actually want the two frames to unify. An example is the unification of the frame of *sends* that represents a caused change of location and the frame of the DO construction which represents a caused change of possession. The two frames are given in Fig. 14. Even though they do not unify we want them to combine. The meaning of the combined frame (i.e. of the

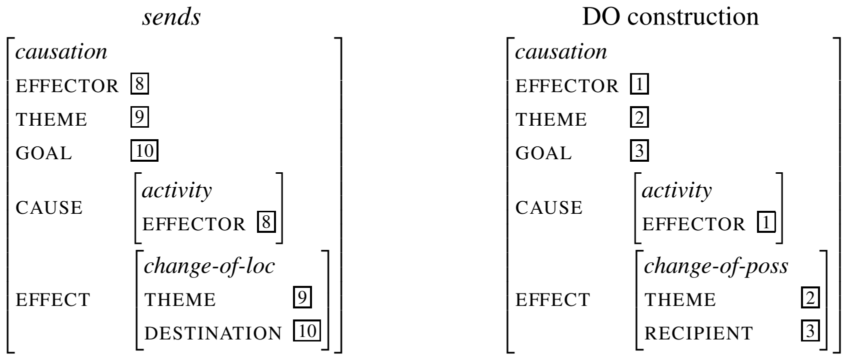

Figure 14: Lexical frame and construction frame of *sends* and the DO construction

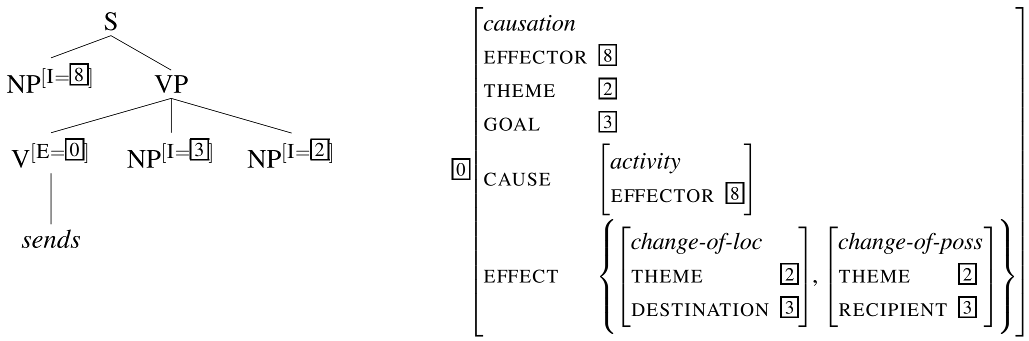

Figure 15: Anchored tree for *sends* with the DO construction

DO construction anchored with *sends*) is, roughly, a causation with effects along different dimensions or 'scales': there is a change of location of the theme and at the same time the theme undergoes also a change of possession.

There are different ways to avoid the mismatch between the two frames. The possibility we choose in this paper is to use set-valued attributes and to assume a special set unification for these. In our case, the attribute EFFECT would have a set of changes as value. When unifying two such sets, the following strategy is adopted: for two elements belonging to the respective sets, if they are of the same type or one is of a subtype of the other, they must unify and the result is part of the resulting set. Otherwise, we take the two elements to describe different aspects that should be considered as a conjunction. We therefore add each of them to the resulting set of frames. In our example, this would lead to the anchored tree in Fig. 15. Note that, in order to obtain the intended identifications between participants of events, we need the top level roles here. They make sure the destination of the change of location is identified with the recipient of the change of possession since both are co-indexed to the GOAL roles of their frames.[12]

6. Conclusion

LTAG is a lexicalized tree grammar formalism with an extended domain of locality and rich possibilities for factorizing syntactic and semantic information on a metagrammatical level. In

[12] An alternative approach, which does not require set-valued attributes, would be to treat the different changes as two different perspectives on the effect of the causation event, represented by two different attributes of the frame. But the details and the consequences of this solution have to be left to future research.

this paper, we propose to exploit this for an implementation of a detailed syntax-related semantic decomposition of both constructional and lexical meaning components. As a case study we have described a model for the dative alternation in English. Our LTAG analysis separates the lexical meaning contribution from the contribution of the construction, taking advantage of LTAG's separation between unanchored elementary trees and lexical anchors. Furthermore, we have factorized the two constructions (double object and prepositional object) into smaller fragments, some of which are shared between the two constructions.

Our analyses have demonstrated that below the level of lexicalized elementary trees and their semantic representations, the metagrammar formalism in LTAG allows us to identify those fragments of syntactic structure that are the potential carriers of meaning. This is partly due to the abstraction from surface structure that comes with LTAG's adjunction operation and the resulting extended domain of locality. As semantic representations we have used decompositional frames represented as typed feature structures, which encode rich semantic information. So far, it seems that the metagrammar descriptions of trees and frames can be rather simple in the sense of being first order tree or feature logics without quantification and negation. The formal properties of our framework need to be further investigated examining a larger range of semantic phenomena. Moreover, we aim not only at theoretically modeling certain linguistic phenomena but also at implementing corresponding grammar fragments. The tools for implementing and testing LTAG grammars are already available, though they need to be adapted to our needs concerning the feature logic we choose.[13]

References

Abeillé, Anne. 2002. Une grammaire électronique du français. CNRS Editions.

Abeillé, Anne, and Owen Rambow. 2000. Tree Adjoining Grammar: An Overview. Tree adjoining grammars: Formalisms, linguistic analyses and processing, ed. by Anne Abeillé and Owen Rambow, 1–68. CSLI Publications.

Abeillé, Anne, and Yves Schabes. 1989. Parsing idioms in tree adjoining grammars. Proceedings of EACL 1989.

Bangalore, Srinivas, and Aravind K. Joshi. 2010. Introduction. Supertagging: Using complex lexical descriptions in natural language processing, ed. by Srinivas Bangalore and Aravind K. Joshi, 1–31. MIT Press.

Beavers, John. 2011. An aspectual analysis of ditransitive verbs of caused possession in English. Journal of Semantics 28.1–54.

Bergen, Benjamin K., and Nancy Chang. 2005. Embodied Construction Grammar in simulation-based language understanding. Construction grammars. Cognitive grounding and theoretical extensions, ed. by Jan-Ola Östman and Mirjam Fried, 147–190. John Benjamins.

Bresnan, Joan, and Marilyn Ford. 2010. Predicting syntax: Processing dative constructions in American and Australian varieties of English. Language 86.186–213.

Bresnan, Joan, and Tatiana Nikitina. 2010. The gradience of the dative alternation. Reality exploration and discovery: Pattern interaction in language and life, ed. by Linda Uyechi und Lian-Hee Wee, 161–184. CSLI Publications.

Candito, Marie-Hélène. 1999. Organisation modulaire et paramétrable de grammaires électroniques lexicalisées. Application au français et à l'italien. Université Paris 7 dissertation.

Copestake, Ann, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal recursion semantics: An introduction. Research on Language and Computation 3.281–332.

[13] We will use the metagrammar compiler XMG (`https://sourcesup.cru.fr/xmg/`) and the TAG parser TuLiPA (`https://sourcesup.cru.fr/tulipa/`).

Crabbé, Benoit. 2005. Grammatical development with XMG. Proceedings of LACL 2005, Lecture Notes in Artificial Intelligence 3492, 84–100. Springer.

Crabbé, Benoit, and Denys Duchier. 2005. Metagrammar redux. Constraint solving and language processing, ed. by Henning Christiansen, Peter Rossen Skadhauge, and Jørgen Villadsen, LNCS 3438, 32–47. Springer.

Fillmore, Charles J. 1982. Frame semantics. Linguistics in the morning calm, ed. by The Linguistic Society of Korea, 111–137. Hanshin Publishing Co.

Fillmore, Charles J., Christopher R. Johnson, and Miriam R. L. Petruck. 2003. Background to FrameNet. International Journal of Lexicography 16.235–250.

Frank, Robert. 2002. Phrase structure composition and syntactic dependencies. MIT Press.

Gardent, Claire, and Laura Kallmeyer. 2003. Semantic Construction in FTAG. Proceedings of EACL 2003, 123–130.

Goldberg, Adele E. 1995. Constructions: A construction grammar approach to argument structure. University of Chicago Press.

Joshi, Aravind K., and Yves Schabes. 1997. Tree-Adjoning Grammars. Handbook of formal languages vol. 3. beyond words, ed. by Grzegorz Rozenberg and Arto Saloma, 69–123. Springer.

Kallmeyer, Laura, and Aravind K. Joshi. 2003. Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG. Research on Language and Computation 1.3–58.

Kallmeyer, Laura, and Rainer Osswald. 2012. An Analysis of Directed Motion Expressions with Lexicalized Tree Adjoining Grammars and Frame Semantics. Proceedings of WoLLIC 2012, ed. by Luke Ong and Ruy de Queiroz, LNCS 7456, 34–55. Springer.

Kallmeyer, Laura, and Maribel Romero. 2008. Scope and situation binding in LTAG using semantic unification. Research on Language and Computation 6.3–52.

Koenig, Jean-Pierre, and Anthony R. Davis. 2001. Sublexical modality and the structure of lexical semantic representations. Linguistics and Philosophy 24.71–124.

Koenig, Jean-Pierre, and Anthony R. Davis. 2006. The KEY to lexical semantic representation. Journal of Linguistics 42.71–108.

Krifka, Manfred. 2004. Semantic and pragmatic conditions for the dative alternation. Korean Journal of English Language and Linguistics 4.1–32.

Kroch, Anthony. 1989. Asymmetries in long-distance extraction in a Tree Adjoining Grammar. Alternative conceptions of phrase structure, ed. by Mark R. Baltin und Anthony S. Kroch. University of Chicago Press.

Levin, Beth, and Malka Rappaport Hovav. 2005. Argument realization. Cambridge University Press.

Müller, Stefan. 2006. Phrasal or lexical constructions. Language 82.850–883.

Rappaport Hovav, Malka, and Beth Levin. 1998. Building verb meanings. The projection of arguments: Lexical and compositional factors, ed. by Miriam Butt and Wilhelm Geuder, 97–134. CSLI Publications.

Rappaport Hovav, Malka, and Beth Levin. 2008. The English dative alternation: A case for verb sensitivity. Journal of Linguistics 44.129–167.

Sag, Ivan. 2012. Sign-Based Construction Grammar: An informal synopsis. Sign-based construction grammar, ed. by H. Boas and I. Sag. CSLI Publications.

Van Valin, Robert D., and Randy J. LaPolla. 1997. Syntax. Cambridge University Press.

Van Valin, Robert D., Jr. 2005. Exploring the syntax-semantics interface. Cambridge University Press.

Vijay-Shanker, K., and Aravind K. Joshi. 1988. Feature structures based tree adjoining grammar. Proceedings of COLING, 714–719.

Xia, Fei. 2001. Automatic grammar generation from two different perspectives. University of Pennsylvania dissertation.

Xia, Fei, Martha Palmer, and K. Vijay-Shanker. 2010. Developing tree-adjoining grammars with lexical descriptions. Supertagging: Using complex lexical descriptions in natural language processing, ed. by Srinivas Bangalore and Aravind K. Joshi, 73–110. MIT Press.

XTAG Research Group. 2001. A Lexicalized Tree Adjoining Grammar for English. Tech. rep., Institute for Research in Cognitive Science, University of Pennsylvania.