Syntax-driven semantic frame composition in Lexicalized Tree Adjoining Grammars

Laura Kallmeyer and Rainer Osswald
Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany

ABSTRACT

The grammar framework presented in this paper combines Lexicalized Tree Adjoining Grammar (LTAG) with a (de)compositional frame semantics. We introduce elementary constructions as pairs of elementary LTAG trees and decompositional frames. The linking between syntax and semantics can largely be captured by such constructions since in LTAG, elementary trees represent full argument projections. Substitution and adjunction in the syntax then trigger the unification of the associated semantic frames, which are formally defined as base-labelled feature structures. Moreover, the system of elementary constructions is specified in a metagrammar by means of tree and frame descriptions. This metagrammatical factorization gives rise to a fine-grained decomposition of the semantic contributions of syntactic building blocks, and it allows us to separate lexical from constructional contributions and to carve out generalizations across constructions. In the second half of the paper, we apply the framework to the analysis of directed motion expressions and of the dative alternation in English, two well known examples of the interaction between lexical and constructional meaning.

Keywords: Lexicalized tree adjoining grammars, syntax-semantics interface, decompositional frame semantics, elementary constructions, metagrammar, feature structures, dative alternation, directed motion expressions

1 INTRODUCTION

The meaning of a verb-based construction depends not only 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 treat 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 and 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 Grammar* (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 and Schabes 1997):

- *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.
- *Factoring recursion from the domain of dependencies*: Constructions related to iteration and recursion are modelled 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 and Joshi (2010) subsume these two properties 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 of the architecture of grammar, which underlies LTAG, has direct consequences for 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 and 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 and Joshi 2003; Gardent and Kallmeyer 2003; Kallmeyer and Romero 2008).

The fact that elementary trees can be directly combined with semantic representations allows 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 and 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*. Crabbé (2005) proposes a purely constraint-based approach to metagrammatical specification (see also Crabbé and Duchier 2005), which does not presume a formal distinction between canonical and derived patterns but generates elementary trees uniformly as minimal models of metagrammatical descriptions. We will adopt this 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 to date. 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 also includes 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, an analysis of the syntax-semantics interface given by elementary constructions that goes beyond the mere enumeration of form-meaning pairs calls for a (meta)grammatical system of constraints consisting of both syntactic and semantic components. Note that such an approach 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 one hand and frames on the other hand, without requiring that there be any structural isomorphism between them. Frames can be understood as cognitive structures representing situations or states of affairs, and they can be formalized as typed feature structures (see Section 3). The metagrammar specifies the syntactic and semantic properties of constructional fragments and defines how they can combine to form 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 on directed motion expressions and on the dative alternation in English, which are well known to be sensitive to lexical and constructional meaning components. We will show how the constructional aspects that these phenomena have in common can be captured within the metagrammatical decomposition.

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 and Chang 2005) and Sign-Based Construction Grammar (Sag 2012).

The structure of the paper is as follows: Section 2 gives a short introduction to LTAG and the metagrammar approach; Section 3 introduces the idea of a frame-based semantics and provides the formal details of the kind of feature structures and feature logic we use for frame-semantic modelling. In Section 4, we present our model of the syntax-semantics interface, which crucially relies on elementary constructions defined as pairs of LTAG trees and semantic frames. We put the framework to work by modelling the syntax-semantics interface of directed motion and caused motion constructions (Section 5) and the dative alternation in English (Section 6). Section 7 briefly discusses the computational complexity of syntactic and semantic composition.

2 LEXICALIZED TREE ADJOINING GRAMMARS

2.1 *Brief introduction to TAG*

Tree Adjoining Grammar (*TAG*, Joshi and Schabes 1997; Abeillé and 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 Figure 1. In this derivation, the elementary trees for *John* and *spaghetti* substitute into the subject and the object slots of the elementary tree for *likes*, and the *obviously* modifier tree adjoins to the VP node.

In case of an adjunction, the tree being adjoined has exactly one leaf that is marked as the *foot node* (marked by 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 original 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.

In order to capture syntactic generalizations in a more satisfactory 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 and Joshi 1988). In an FTAG, each node has a top and a bottom feature structure (except substitution nodes that have only a top structure). Nodes

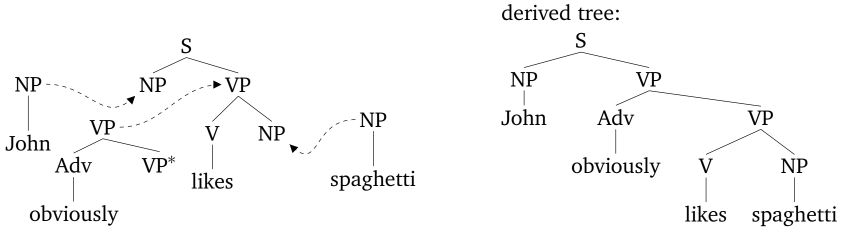


Figure 1: A sample derivation in TAG

in the same elementary tree can share features. Adjunction constraints are expressed via the feature structures. An example is given in Figure 2, where the top feature structure is notated as a superscript and the bottom feature structure as a subscript of the respective node. The features in this example are taken from the XTAG grammar (XTAG Research Group 2001). In the *singing* tree the label $\boxed{1}$ is used to express the fact that the agreement features of the VP have to be the same as those of the subject NP. Furthermore, the different MODE values in the top and bottom feature structures of the VP node express an *obligatory* adjunction constraint. Since the values are different, the two feature structures cannot unify. Consequently, one has to adjoin a tree that separates the top structure from the bottom structure.

During substitution and adjunction, the following unifications take place. In a substitution operation, the top of the root of the new

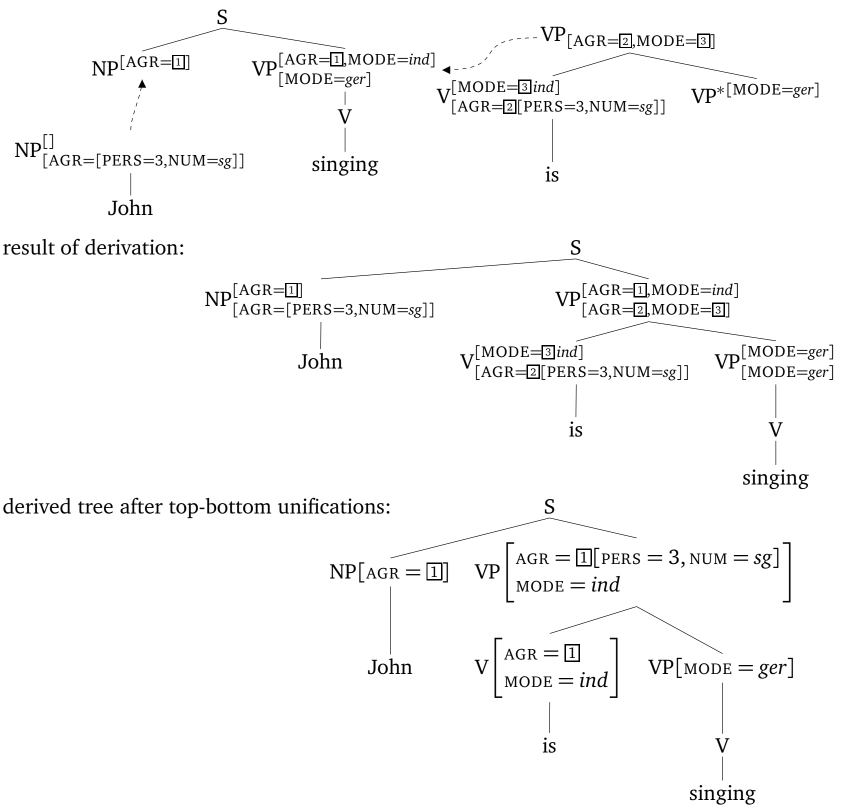


Figure 2: Feature sharing and adjunction constraints in FTAG

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. See again Figure 2 for an example. The middle tree shows the result of the derivation, including feature unifications arising from substitutions and adjunction. The lower tree shows the result one obtains after the final top-bottom unification. As illustrated by this example, constraints among dependent nodes can be more easily expressed in FTAG than in the original TAG formalism. Since the set of feature structures allowed in a given TAG is finite, the feature structures do not extend the generative capacity of the formalism and do not increase its parsing complexity.

2.2 *Elementary trees and tree families*

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 (*θ-criterion for TAG*). Most argument slots are substitution nodes, in particular the nodes for nominal arguments (see the elementary tree for *likes* in Figure 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).

LTAG allows for a high degree of factorization inside the lexicon, i.e., inside the set of lexicalized elementary trees. One factorization arises from separating the specification of *unanchored* elementary trees 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 Figure 3) containing the patterns for different realizations of the arguments (canon-

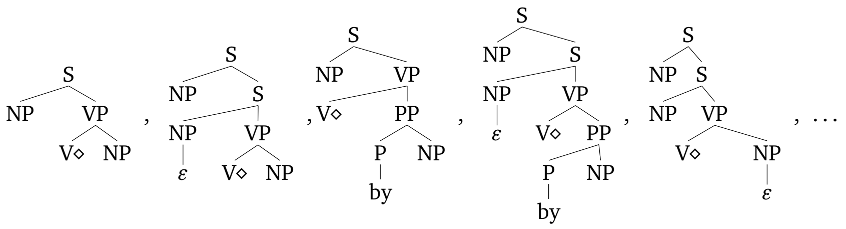


Figure 3: Unanchored tree family for transitive verbs

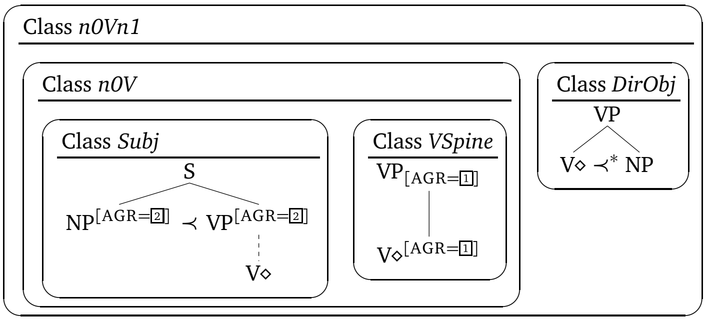
ical position, extraction, etc.) in combination with active and passive. The node marked with a diamond is the node that gets filled by the lexical anchor.

2.3 *Metagrammatical decomposition of elementary trees*

Unanchored elementary trees are usually specified by means of a *metagrammar* (Candito 1999; Crabbé and Duchier 2005; Crabbé *et al.* 2013) which consists of dominance and precedence constraints and category assignments. The elementary trees of the grammar are then 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 one 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 Figure 4 that can be implemented in XMG (*eXtensible MetaGrammar*; cf. Crabbé *et al.* 2013). Each class is represented in a box with the name of the class on top. The box contains a graphical representation of the tree description specified in the class and it contains other classes used in this class. The class *Subj* does not use any other class and it contains a tree description specifying that there are four nodes labelled S, NP, VP and V with the dominance (dashed edge), immediate dominance (solid edge) and immediate linear precedence ($\prec$)

Figure 4:
MG classes for transitive verbs (only canonical word order)



relations depicted in Figure 4. The diamond on the V node marks this node as the lexical anchor. Concerning features, the AGR feature values of the NP and VP nodes must be equal. The class *VSpine* specifies that there are nodes with categories VP and V with an immediate dominance such that the AGR features of the two nodes are equal. The class *n0V* uses the two classes for the subject and the verbal spine without adding any further constraints to the resulting tree description. Computing a minimal model for this class amounts to finding a tree that contains only nodes and edges described in the class. Since the lexical anchor is unique, the two anchor V nodes in *n0V* must be equal. This means that the only minimal model of *n0V* is the elementary tree for intransitive verbs with the subject in canonical position. The class *DirObj* in Figure 4 specifies that a direct object can be realized by an NP node that is immediately dominated by a VP node and that is a right sister of the V anchor node ($\prec^*$ stands for linear precedence). Combining this class with the *n0V* class yields the class *n0Vn1* for transitive verbs that leads, in this simple example, to the first tree of Figure 3 as a minimal model. The tree descriptions allow for conjunction and disjunction but not for quantification or negation. As we have seen, each class can use other classes and add new constraints on the minimal models to be computed.

3 DECOMPOSITIONAL FRAME SEMANTICS

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

$$\begin{bmatrix} \textit{eating} & \\ \text{ACTOR} & \boxed{1} \\ \text{THEME} & \boxed{2} \end{bmatrix} \qquad \begin{bmatrix} \textit{sending} & \\ \text{ACTOR} & \boxed{1} \\ \text{THEME} & \boxed{2} \\ \text{GOAL} & \boxed{3} \end{bmatrix}$$

Figure 5:
Simple role frames for eating and sending events

are to be understood as cognitive structures that represent the described situations or state of affairs. In their most basic form, frames represent the type of a situation and the semantic roles of the participants; they correspond to feature structures of the kind shown in Figure 5. Frame semantics as currently implemented in the Berkeley FrameNet project (Fillmore *et al.* 2003) builds basically on role frames of this form, 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. The ultimate goal of the FrameNet project is to devise a sufficiently rich collection of frames that allows one to describe all kinds of specific and general situation types. FrameNet frames can be related to each other in various ways. For instance, a frame can be characterized as being more specific than another frame (inheritance), or as putting a different focus on the involved participants (perspectivalization), or as being the cause or the effect component of a complex causation event. It is this interrelatedness of frames which, according to Fillmore (2007), will eventually give rise to generalizations about the morpho-syntactic realization of semantic arguments.

As shown by Osswald and Van Valin (2014), Fillmore's goal of deriving generalizations about the syntax-semantics interface would profit considerably from an analysis that takes into account the internal structure of events and state of affairs in a systemic way. This observation is in line with the fact that, 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 and Rappaport Hovav 2005, for an overview). These theories assume that the morphosyntactic realization of an argument depends crucially on the structural position of the argument within the decomposition. A simple example of such a decomposition for the transitive verb *break* is shown in (1), using the notation of Rappaport Hovav and Levin (1998).[1]

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

However, the precise status of such terms with respect to formal interpretation and inferencing is mostly neglected in the literature on argument realization.

3.2 *Semantic frames as models of meaning*

It is a central goal of the approach presented in this paper to integrate the template-based event structure decompositions with a fully formalized frame semantics. Such a decompositional semantic representation allows us to associate specific components of the semantic representation with specific syntactic fragments in the metagrammar. Crucially, we take the semantic structures associated with the syntactic structures as *genuine semantic representations*, not as some kind of yet to be interpreted logical expressions. The grammar generated by the metagrammar then consists of pairs of elementary morphosyntactic trees and elementary meaning structures. In Section 4, we will refer to such pairs as *elementary constructions*. The minimal model view is thus adopted for the semantic dimension as well: the semantic structures of elementary constructions are defined as minimal models of the constraints specified in the metagrammar – in much the same way as the syntactic structures of elementary constructions are minimal models of the specifications in the metagrammar.[2]

Let us return to the decomposition given in (1). It says, basically, that an event denoted by transitive *break* consists of an activity of someone or something *x* which causes a certain change of state of

[1] The idea of using representations of this kind for predicting argument realization patterns can be traced at least back to Foley and Van Valin (1984); see also Van Valin and LaPolla (1997).

[2] The use of minimal models in computational semantics has also been proposed by Blackburn and Bos (2003), among others; see also Konrad (2004). A view closely related to ours is advocated by Hamm *et al.* (2006), who propose that the logical expressions used in semantics are best considered as constraints on possible models, understood as conceptual representations. The main purpose of the logical expressions is then to characterize the minimal models, which play a crucial role in semantic processing.

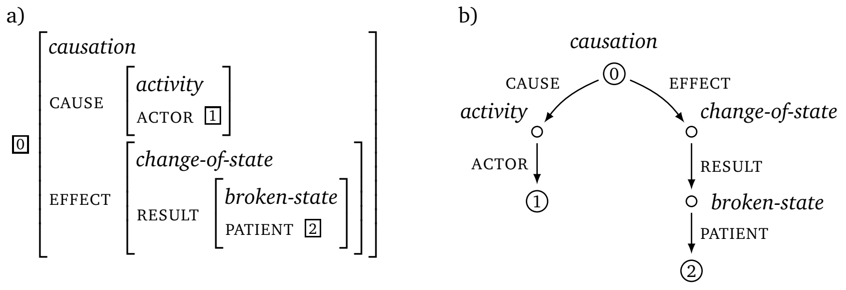


Figure 6: Possible frame representation corresponding to template (1)

something y, namely y becoming broken. Put differently, the events in question are of type *causation* and have as their CAUSE component an (unspecified) activity of x and as their EFFECT component a change of state that results into y's state of being broken. There are various ways of explicating (1) in logical terms. For instance, if we take the paraphrase just given as a blueprint, a possible logical formulation could look like (2).

(2) $$\textit{causation}(e) \wedge \text{CAUSE}(e,e') \wedge \text{EFFECT}(e,e'') \wedge \textit{activity}(e') \wedge \text{ACTOR}(e',x) \\ \wedge\ \textit{change-of-state}(e'') \wedge \text{RESULT}(e'',s) \wedge \textit{broken-state}(s) \wedge \text{PATIENT}(s,y)$$

Let us keep aside for the moment the question of how to treat the arguments of the predicates in (2), that is, whether to treat them as constants or as variables, and if as variables, whether and how they are bound by quantification or lambda abstraction. Representations of verb meaning like (2) are closely related to Neo-Davidsonian approaches to event semantics as proposed in Parsons (1990), among others (cf. Maienborn 2011).

It is important to notice that CAUSE is used differently in (1) and (2). In (1), CAUSE is to be interpreted as a relation between an activity and a change of state, that is, as the causation relation between events. In (2), by contrast, CAUSE denotes a functional relation that relates a causation event e to its cause component e'. In fact, it is an essential property of (2) that *all* binary relations involved are *functional*. This allows us to associate with (2) the frame shown in Figure 6, with frames understood as potentially nested typed feature structures, possibly extended with additional constraints. The graph on the right of the figure 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. Frame representations combine two central aspects of template-based decompositions and logical representations: like decompositional schemas, frames 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. Using frames for semantic representation is also in line with Löbner's (2014) hypothesis about the structure of representations in the human cognitive system. And, last but not least, due to their functional backbone, frames have good computational properties (see Section 3.3 and Section 7).

3.3 *Formal specification of frames*

In the following, we define frames of the type depicted in Figure 6b in a more formal way as graph-like structures, and we define the notions of subsumption and unification of frames. Moreover, we introduce a specification language for frames, which we employ for extending the description of elementary syntactic trees in the metagrammar by a frame-semantic dimension. In analogy to the syntactic dimension, semantic frames are considered as minimal models of metagrammatical specifications. A good part of the following formal framework builds on existing work on feature logics as summarized in Rounds (1997).

3.3.1 Base-labelled feature structures with types and relations

Ordinary feature structures as defined, e.g., in Carpenter (1992) come with a distinguished node, the *root*, from which each other node of the structure is reachable along the (directed) edges of the graph. In the example in Figure 6, the root node is given by the node labelled by 0. The standard unification of feature structures requires the respective roots to be identified. Semantic composition associated with TAG operations, however, typically calls for unifying a certain semantic structure with a *sub*structure of another structure; this is even the case for plain argument insertion. Moreover, in later sections we will see examples of semantic structures for which the assumption of more than one root node seems appropriate. We therefore employ typed feature structures with *multiple* base nodes.[3] Furthermore, we also allow *relations* between nodes (see Section 5.2 for an application where

[3] Our approach builds partly on Hegner (1994). The need for multi-rooted feature structures in the context of language modelling has also been noted by Sikkel (1997).

the relation in question is the mereological part-of relation between regions of space).

It is useful to first define the structures in question without explicitly mentioning the multiple base nodes involved. The following definition presumes a *signature* $\langle A, T, R\rangle$ consisting of a finite set A of *attributes*, a finite set T of *types*, and a finite set R of *relation symbols*. Each relation symbol $r \in R$ has an *arity* $\alpha(r) \in \{2, 3, 4, \ldots\}$ and we write R_n for the set of n-ary relation symbols.[4] A *typed feature structure with relations* over the signature $\langle A, T, R\rangle$ is a quadruple $\langle V, \delta, \tau, \rho\rangle$ in which V is a finite set of *nodes*; δ is a partial function from $V \times A$ to V, the *node transition function*; τ is a function from V to $\wp(T)$, the *typing function*; and ρ is a function defined on $\bigcup\{V^n \mid n \in \alpha(R)\}$ which takes elements from V^n to subsets of R_n. Note that our definition comes without a type hierarchy and that the typing function τ assigns *sets* of types to each node. The reason is that we prefer to handle type hierarchies as generated by type constraints (cf. Section 3.3.4 below). If $\tau(v) = \varnothing$, this means that v has the most general type.[5]

The standard definition of *subsumption*[6] can be adapted to our framework in a straightforward way. Given two feature structures $F_1 = \langle V_1, \delta_1, \tau_1, \rho_1\rangle$ and $F_2 = \langle V_2, \delta_2, \tau_2, \rho_2\rangle$ over $\langle A, T, R\rangle$, then F_1 *subsumes* F_2, in symbols, $F_1 \sqsubseteq F_2$, if there is a function h from V_1 to V_2, called a *morphism*, which has the following properties:

- If $\delta_1(v, f)$ is defined for $v \in V$ and $f \in A$,
  then $\delta_2(h(v), f)$ is defined and $\delta_2(h(v), f) = h(\delta_1(v, f))$.
- For every $v \in V$, $\tau_1(v) \subseteq \tau_2(h(v))$.
- For every $n \in \alpha(R)$ and $v_1, \ldots, v_n \in V$,
  $\rho_1(v_1, \ldots, v_n) \subseteq \rho_2(h(v_1), \ldots, h(v_n))$.

[4] Strictly speaking, the arity function α is part of the signature, but we keep this aside for ease of exposition.

[5] Note that types could have also been introduced as unary relation symbols; but for the task of semantic modelling it seems appropriate to concede a special status to sortal information. Conversely, we could get rid of the relations by reifying tuples of nodes by separate nodes which are related to the elements of the tuple by special "argument" attributes.

[6] Cf., e.g., Rounds (1997).

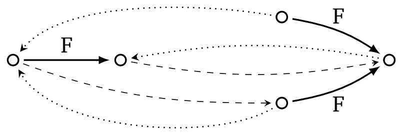


Figure 7: Non-isomorphic feature structures which subsume each other

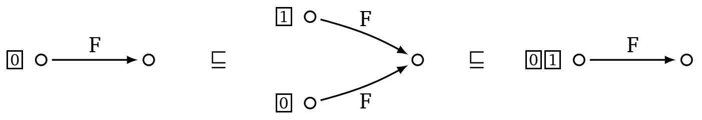


Figure 8: Strict subsumption between base-labelled feature structures

The subsumption relation defined this way is a preorder, but notice that mutual subsumption does not imply isomorphism, as illustrated by the example in Figure 7.

The above definition of feature structures does not yet capture one of the most crucial aspects of frame-based modelling, namely the property that every component is accessible via attributes from a distinguished set of base nodes. In order to formalize this requirement, let B be a countably infinite set of *base labels*. Without loss of generality, we may assume that $B = \{\boxed{0}, \boxed{1}, \boxed{2}, \dots\}$. A *base-labelled* feature structure over $\langle A, T, R, B\rangle$ is now defined as feature structure $\langle V, \delta, \tau, \rho\rangle$ over $\langle A, T, R\rangle$ together with a partial function β from B to V, the *base-labelling function*, such that every node is reachable from some *base node*, i.e., from some element of $\beta(B) \subseteq V$ via node transitions; that is, with δ extended to a partial function from $V \times A^*$ to V in the usual way, for every $v \in V$ there is a $v' \in \beta(B)$ and an *attribute path* $p \in A^*$ such that $v = \delta(v', p)$.

Morphisms between base-labelled feature structures are required to respect the base labelling. That is, a *morphism* between two base-labelled feature structures $G_1 = \langle F_1, \beta_1\rangle$ and $G_2 = \langle F_2, \beta_2\rangle$ is a morphism h from F_1 to F_2 such that if $\beta_1(l)$ is defined for $l \in B$, then $\beta_2(l) = h(\beta_1(l))$. In particular, this implies that if we add additional base labels to a given feature structure by extending the domain of the base-labelling function, we get a more specific feature structure with respect to subsumption; see Figure 8 for an example. It is not difficult to see that there exists at most one morphism between two base-labelled feature structures over a given signature; hence mutual subsumption now implies isomorphism. It follows that we can speak of "the" *least upper bound* $G_1 \sqcup G_2$ of two base-labelled feature structures G_1 and G_2 with respect to $\sqsubseteq$, which is uniquely determined up to isomorphism, if existent at all. Hegner (1994) shows that there are efficient *unification algorithms* for computing $G_1 \sqcup G_2$ (see also Section 7).

The usual scenario for the unification of base-labelled feature structures in the applications described in the following sections presumes that they come with disjoint sets of labels. This means that feature structures need to be *relabelled*, if required. Formally, $\langle F, \beta' \rangle$ is a *relabelling* of $\langle F, \beta \rangle$ if there is a function σ on B such that $\beta'(\sigma(B)) = \beta(B)$, i.e., if the same nodes of F are base-labelled as before.

Let $G_1 = \langle F_1, \beta_1 \rangle$ and $G_2 = \langle F_2, \beta_2 \rangle$ be two base-labelled feature structures with disjoint labellings, that is, β_1 and β_2 have disjoint domains. Suppose $\boxed{0}$ is a base label of G_1 and $\boxed{1}$ is a base label of G_2. Then, when we speak of the unification of G_1 and G_2 under "identification of $\boxed{0}$ and $\boxed{1}$", we mean the unification, as defined above, of G_1' and G_2, where G_1' is the relabelling of G_1 resulting from adding the label $\boxed{1}$ to the node labelled by $\boxed{0}$. Note that we can also define G_1' without resorting to relabelling by unifying G_1 with a single-node feature structure without attributes, type, and relations, where the single node carries the labels $\boxed{0}$ and $\boxed{1}$.

3.3.2 Attribute-value descriptions

In order to specify semantic frames in the metagrammar, we need a declarative language for describing the structures introduced in the last section. The crucial point about the base labelling is that a feature structure can be characterized completely by restricting explicit reference to base-labelled nodes only. The reason is that every node of a base-labelled feature structure is accessible from one of the base nodes via successive attribute transitions.

The following language of *attribute-value descriptions* builds on the versions summarized in Rounds (1997), extended by notations taken from Hegner (1994) and Osswald (1999). First, we introduce the language of *general* attribute-value descriptions, which allows us to talk about arbitrary nodes of a feature structure. The *primitive* general attribute-value descriptions over a signature $\langle A, T, R \rangle$ are expressions of the form t, r, $p : t$, $p \doteq q$, $p \triangleq q$, $(p_1, \ldots, p_n) : r$ and $\langle p_1, \ldots, p_n \rangle : r$, with $p, p_i, q \in A^*$, $t \in T$, and $r \in R$. Let F be a feature structure $\langle V, \delta, \tau, \rho \rangle$ of signature $\langle A, T, R \rangle$ with $v, w, v_i \in V$. The *satisfaction relation* $\vDash$ between nodes (and node tuples) of F and attribute-value descriptions is defined as follows:

(3) a. $v \vDash t$ iff $v \in \tau(t)$
b. $\langle v_1, \ldots, v_n \rangle \vDash r$ iff $\langle v_1, \ldots, v_n \rangle \in \rho(r)$
c. $v \vDash p : t$ iff $\delta(v, p) \vDash t$
d. $v \vDash p \doteq q$ iff $\delta(v, p) = \delta(v, q)$
e. $\langle v, w \rangle \vDash p \triangleq q$ iff $\delta(v, p) = \delta(w, q)$
f. $v \vDash (p_1, \ldots, p_n) : r$ iff $\langle \delta(v, p_1), \ldots, \delta(v, p_n) \rangle \vDash r$
g. $\langle v_1, \ldots, v_n \rangle \vDash \langle p_1, \ldots, p_n \rangle : r$ iff $\langle \delta(v_1, p_1), \ldots, \delta(v_n, p_n) \rangle \vDash r$

We allow general attribute-value descriptions (of the same arity) to be combined by all Boolean connectives plus $\top$ (true) and $\bot$ (false), with the usual Boolean semantics. Moreover, it is convenient to allow attribute prefixing for arbitrary (one-place) attribute-value descriptions. For instance, if ϕ is a general one-place description and $p \in A^*$, then $p : \phi$ is also a general attribute-value descriptions, with $v \vDash p : \phi$ iff $\delta(v, p) \vDash \phi$.

Now let us add base labels to the description language. *Labelled* attribute-value descriptions are of the form $l \cdot \phi$, $l \cdot p \triangleq k \cdot q$, and $\langle l_1 \cdot p_1, \ldots, l_n \cdot p_n \rangle : r$, with $k, l, l_i \in B$. In contrast to general descriptions, which are satisfied by nodes of a feature structure, labelled descriptions are satisfied by base-labelled feature structures.

(4) a. $\langle F, \beta \rangle \vDash l \cdot \phi$ iff $\beta(l) \vDash \phi$
b. $\langle F, \beta \rangle \vDash l \cdot p \triangleq k \cdot q$ iff $\langle \beta(l), \beta(k) \rangle \vDash p \triangleq q$
c. $\langle F, \beta \rangle \vDash \langle l_1 \cdot p_1, \ldots, l_n \cdot p_n \rangle : r$ iff
$\langle \delta(\beta(l_1), p_1) \ldots, \delta(\beta(l_n), p_n) \rangle \vDash r$

In the case of the empty attribute path ε, we write l instead of $l \cdot \varepsilon$. Again, we allow Boolean combinations of labelled descriptions.

Every base-labelled feature structure can be characterized by a finite conjunction of primitive labelled attribute-value descriptions. For instance, the frame structure of Figure 6 is specified by the following conjunction:

(5) $\boxed{0}$: *causation* $\wedge$ $\boxed{0}$ · CAUSE : *activity* $\wedge$ $\boxed{0}$ · CAUSE ACTOR $\triangleq$ $\boxed{1}$ $\wedge$
$\boxed{0}$ · EFFECT : *change-of-state* $\wedge$ $\boxed{0}$ · EFFECT RESULT : *broken-state* $\wedge$
$\boxed{0}$ · EFFECT RESULT PATIENT $\triangleq$ $\boxed{2}$

Note that the attribute-value matrix shown in Figure 6a can be regarded as a normal form of the attribute-value description in (5), with conjunction symbols left implicit.

3.3.3 Reformulation in first-order predicate logic

It is not difficult to reformulate the attribute-value descriptions introduced above as expressions in first-order predicate logic, thereby regarding feature structures as standard set-theoretic models. This viewpoint is useful because predicate logic is the most conceptually basic logical language at hand and, moreover, it gives us a better connection to standard approaches in linguistic semantics, such as Neo-Davidsonian approaches (cf. Section 3.2).

First we need to become clear about what to make of the signature in the context of a first-order interpretation. For the elements of A, T, and R this is fairly obvious: attributes denote functional relations and are hence to be seen as two-place predicates; types are one-place predicates; n-ary relation symbols are n-place predicates. The treatment of the elements of B is slightly more intricate. Since base labels serve as *names*, it seems appropriate to regard them as constants. However, the standard way of interpreting constants in first-order logic requires each of them to correspond to an element of the underlying domain, which is not the case for the base labels since only some of them are used in a given structure. The solution is to treat them as one-place predicates, with the additional requirement that they are satisfied by at most one element of the domain.[7]

An *interpretation* of A, T, R, and B in the usual sense of first-order logic is a pair $\langle D, M \rangle$, consisting of a set D, the *domain* (or *universe*) and an *interpretation function* M which takes the elements of A to binary relations on D, the elements of T and B to subsets of D, and elements of R of arity n to n-ary relations on D. Since we require attribute relations to be functional and base labels to denote at most one element, we are only interested in interpretations that satisfy the following axioms for all $f \in A$, $l \in B$:

[7] Our use of base labels is similar to using *nominals* in modal logic reformulations of attribute-value logic as proposed by Blackburn (1993). While Blackburn introduces nominals to replace path-value identities, we keep the latter expressions and reserve base labels for node identification "visible from the outside". In particular, base labels matter for subsumption and unification. Another approach worth mentioning is that of Reape (1994), who introduces a polymodal language with nominals and relations.

(6) a. $\forall x \forall y \forall z (f(x,y) \wedge f(x,z) \rightarrow y = z)$
b. $\forall x \forall y (l(x) \wedge l(y) \rightarrow x = y)$

In other words, we are interested in *models* of the *theory* given by the axioms in (6).

Next, we need to rephrase the attribute-value descriptions in (3) and (4) as first-order expressions. One way to do this is to explicate the intended meaning of these descriptions in terms of predicate logic. Consider, for instance, attribute-value descriptions of the form $p : t$ (3-c). Descriptions of this sort can be phrased as 'x such that the p of x is a t'. For example, EFFECT : *change-of-state* is short for 'e such that the effect of e is a change-of-state'. Formally, this formulation can be rendered as $\lambda x(t(\iota y(p(x,y))))$. Elimination of the definite description gives $\lambda x(\exists y(p(x,y) \wedge t(y)))$ plus a uniqueness constraint that is already captured by (6-a). Hence, $p : t$ can be explicated as in (7-a), given (6-a).

(7)	a.	$p : t$	$\lambda x \exists y (p(x,y) \wedge t(y))$
	b.	$p \doteq q$	$\lambda x \exists y (p(x,y) \wedge q(x,y))$
	c.	$f p$	$\lambda x \lambda z \exists y (f(x,y) \wedge p(y,z))$

By a similar argument, $p \doteq q$ can be translated as in (7-b). (7-c) simply says that attribute concatenation means relational composition. (The empty attribute path ε is interpreted by the identity relation on D.) A possible explication of (4-a) and (4-b) is shown in (8-a) and (8-b), respectively.

(8)	a.	$l \cdot p : \phi$	$\exists x(l(x) \wedge \exists y(p(x,y) \wedge \phi(y)))$
	b.	$l \cdot p \triangleq k \cdot q$	$\exists x \exists y(l(x) \wedge k(y) \wedge \exists z(p(x,z) \wedge q(y,z)))$

The labelled description $l \cdot p : \phi$ says that the element labelled by l satisfies $p : \phi$, in symbols, $\exists y(p(\iota x(l(x)), y) \wedge \phi(y))$. Another elimination of the definite description, using (6-b), then gives rise to $\exists x(l(x) \wedge \exists y(p(x,y) \wedge t(y)))$, as desired. (8-b) can be derived in a similar vein, and the same is true of the translations of the remaining descriptions of (3) and (4).

If we apply the described reformulation technique to the description in (5), then the resulting first-order expression is equivalent, under the axioms in (6), to the following expression:

(9) $\exists e \exists e' \exists e'' \exists s \exists x \exists y\, (\boxed{0}(e) \wedge \textit{causation}(e) \wedge \textsc{cause}(e,e') \wedge \textsc{effect}(e,e'')$
$\wedge\, \textit{activity}(e') \wedge \textsc{actor}(e',x) \wedge \boxed{1}(x) \wedge \textit{change-of-state}(e'')$
$\wedge\, \textsc{result}(e'',s) \wedge \textit{broken-state}(s) \wedge \textsc{patient}(s,y) \wedge \boxed{2}(y))$

We can rewrite (9) more succinctly by the following slight abuse of notation. Since base labels are unique identifiers (if they refer at all), we can introduce them via the back door as constants by replacing $\boxed{0}$ by $\lambda x(x = \boxed{0})$, etc. Then (9) simplifies to (10).

(10) $\exists e' \exists e'' \exists s\, (\textit{causation}(\boxed{0}) \wedge \textsc{cause}(\boxed{0},e') \wedge \textsc{effect}(\boxed{0},e'') \wedge \textit{activity}(e')$
$\wedge\, \textsc{actor}(e',\boxed{1}) \wedge \textit{change-of-state}(e'') \wedge \textsc{result}(e'',s)$
$\wedge\, \textit{broken-state}(s) \wedge \textsc{patient}(s,\boxed{2}))$

Let us now look at the *(minimal) generic model* $\langle D, M\rangle$ of formula (9) under the "background" theory (6). "Generic" means that in this model no attribute-value description holds which is not derivable from (9) and (6) by logical inference. The domain D consists of six elements, one for each variable in (9), and the interpretation of the attributes, types, and base labels can be directly read off from (9). The resulting model is basically the structure depicted by Figure 6b, now viewed as a first-order model. This observation can be generalized as follows: There is a one-to-one correspondence between the most general base-labelled feature structures that satisfy conjunctive labelled attribute-value descriptions and the minimal generic first-order models of these descriptions rephrased as first-order expressions, presuming the axioms in (6).

3.3.4 Attribute-value constraints

We define *attribute-value constraints* as universally quantified (one-place) general attribute-value descriptions. That is, if ϕ is a one-place attribute-value description then $\forall\phi$ (in first-order notation, $\forall x(\phi(x))$) is a constraint, which is satisfied by a feature structure if and only if ϕ is satisfied by every node of the structure. We write $\phi \preceq \psi$ for $\forall(\phi \rightarrow \psi)$.

Since boolean expressions have an equivalent *conjunctive normal form*, every constraint $\forall\phi$ can be transformed into a conjunction of constraints of the form listed in (11), in which the ϕ_i's and ψ_j's are primitive attribute-value descriptions.

(11) a. $\phi_1 \wedge \ldots \wedge \phi_n \preceq \psi_1 \vee \ldots \vee \psi_m$
b. $\top \preceq \psi_1 \vee \ldots \vee \psi_m$
c. $\phi_1 \wedge \ldots \wedge \phi_n \preceq \bot$

If $m = 1$, then $\forall\phi$ is called a *Horn constraint.* In the following we are concerned with Horn constraints, if not otherwise indicated. Here are some examples of Horn constraints:

(12) a. *activity* $\preceq$ *event*
b. *causation* $\preceq \neg$*activity*
(equivalently, in normal form: *causation* $\wedge$ *activity* $\preceq \bot$)
c. $\textsc{agent} : \top \preceq \textsc{agent} \doteq \textsc{actor}$
d. *activity* $\preceq \textsc{actor} : \top$
e. *activity* $\wedge$ *motion* $\preceq \textsc{actor} \doteq \textsc{mover}$

Note that the first-order translation of the constraint in (12-c) gives rise to $\forall x(\exists y(\textsc{agent}(x,y)) \rightarrow \exists z(\textsc{agent}(x,z) \wedge \textsc{actor}(x,z)))$, which is logically equivalent under (6-a) to the formula $\forall x \forall y(\textsc{agent}(x,y) \rightarrow \textsc{actor}(x,y))$. Constraints of the form (12-c) thus express *attribute inclusions*.

In order to apply constraints to labelled descriptions for inferencing, the former need to be turned into labelled descriptions themselves. Recall that constraints hold at each node of a frame, and each node can be accessed from a base node along some attribute path. A constraint $\forall\phi$ thus gives rise to infinitely many labelled descriptions $l \cdot p : \phi$, with $l \in B$ and $p \in A^*$. In fact, $l \cdot p : \phi$ is a logical consequence of $\forall\phi$ in terms of first-order logic, under the axioms in (6). The constraint (12-a), for instance, implies the labelled description $\boxed{0} \cdot \textsc{cause} : \textit{activity} \rightarrow \boxed{0} \cdot \textsc{cause} : \textit{event}$, which can be applied to the description in (5) for type inference.

An important application scenario of the constraints is unification. The crucial question while unifying is: under which conditions do we need to consider only a finite number of descriptions? Feature structure unification under a finite set of labelled Horn descriptions is well-defined and computationally tractable (Hegner 1994); cf. Section 7. A simple sufficient condition is that none of the constraints enforces the introduction of additional nodes. If this is the case then the number of nodes of the unified structure is finite, and hence also the number of relevant paths.[8] It follows that the number of labelled descriptions to be considered for inferencing is finite.

Let us have a look at the constraints in (12) from this perspective. (12-a) and (12-b) are unproblematic since they have no attribute expressions in their consequents. (12-c) also poses no problem because the constraint just adds an attribute leading to a node already given in the antecedent. (12-d) and (12-e), by contrast, both do imply the existence of additional nodes in their consequent. Note that the issue with (12-e) can be remedied by conjoining ACTOR : $\top$ (or MOVER : $\top$) to the antecedent. Note also that a constraint like (12-d) does not necessarily imply that an infinite number of base constraints has to be taken into account. In fact, in this case it wouldn't. But a more careful analysis is required in such cases in general.[9]

The constraints (12-a) and (12-b) express *type inclusion* and *exclusion*, respectively. Recall that our definition of feature structures in Section 3.3.1 does not make use of a type hierarchy. Instead, we explicitly specify the possible combinations of atomic types by type inclusion and exclusion constraints. The elements of the type hierarchy in the usual sense are then defined as the sets of atomic types that are closed and consistent with respect to type inclusion and exclusion constraints. It is well known that in the finite case, Horn constraints give rise to bounded-complete ordered sets (ordered by set inclusion) by this construction, and that every bounded-complete ordered set can be constructed this way.[10] Whether to precompile the type hierarchy or to do type inference on the fly is an issue of implementation.

Relational descriptions can also be used in constraints. For instance, the transitivity of a binary relation is expressed by the following Horn constraint:

(13) $(p_1, p_2) : r \wedge (p_2, p_3) : r \rightarrow (p_1, p_3) : r$

[8] Infinite paths that might arise through cyclic structures can be avoided by limiting the maximal length of a path to the number of nodes.

[9] See also Carpenter (1992, pp. 95ff).

[10] An ordered set is *bounded-complete* if every subset that has an upper bound has a least upper bound. Note that bounded-complete ordered sets come with a least element, namely the least upper bound of the empty set.

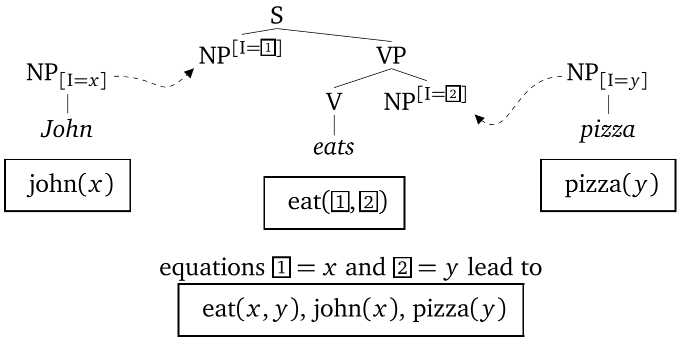


Figure 9: Syntactic and semantic composition as in Gardent and Kallmeyer (2003) and Kallmeyer and Romero (2008)

Since the consequent of (13) does not instantiate new nodes, this constraint is unproblematic when being processed during unification.[11]

4 LTAG WITH FRAME SEMANTICS

4.1 *Elementary constructions and the syntax-semantics interface*

As to the syntax-semantics interface, we basically build on approaches which link a semantic representation to an entire elementary tree and which model composition by unifications triggered by substitution and adjunction. For example, in Gardent and Kallmeyer (2003), every elementary tree is paired with a set of typed predicate logical formulas containing meta-variables linked to features in FTAG structures (see also Kallmeyer and Joshi 2003, Kallmeyer and Romero 2008). The syntactic composition then triggers unifications that lead to equations between semantic components. A (simplified) example is given in Figure 9. The feature I on the nodes is a syntax-semantics interface feature which stands for "individual". Linking, i.e., the assignment of semantic roles to syntactic arguments, is done via these interface features. In Figure 9, the syntactic unifications lead to equations $\boxed{1} = x$ and $\boxed{2} = y$. Therefore, in the semantic formulas, we have replacements of the variables $\boxed{1}$ and $\boxed{2}$ with x and y respectively. The formulas we obtain after having applied these assignments are collected in a set that is then interpreted conjunctively.

[11] Inference closure under (13) corresponds to calculating the transitive closure of the denoted relation on the given domain; its time complexity is known to be better than $\mathcal{O}(n \cdot e)$, where e is the number of pairs initially falling under r.

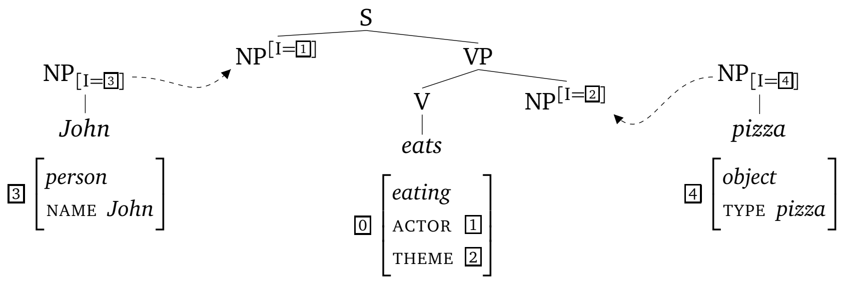


Figure 10: Syntactic and semantic composition for *John eats pizza*

unification under the descriptions $\boxed{1} \triangleq \boxed{3}$ and $\boxed{2} \triangleq \boxed{4}$ leads to

$$\boxed{0}\begin{bmatrix} \textit{eating} & \\ \text{ACTOR} & \boxed{1}\boxed{3}\begin{bmatrix}\textit{person}\\ \text{NAME}\ \textit{John}\end{bmatrix} \\ \text{THEME} & \boxed{2}\boxed{4}\begin{bmatrix}\textit{object}\\ \text{TYPE}\ \textit{pizza}\end{bmatrix}\end{bmatrix}$$

The focus of this paper is on a decompositional semantics for elementary LTAG trees using frames. Figure 10 shows how the example from Figure 9 can be translated into a frame-based semantic representation in a fairly straightforward way. Each elementary tree is paired with a frame, that is, with a base-labelled feature structure as defined in Section 3.3, and base labels are used as values of interface features on the tree. Syntactic unification then triggers label equations. Here, the substitutions give rise to $\boxed{1} \triangleq \boxed{3}$ and $\boxed{2} \triangleq \boxed{4}$. Unification of the semantic frames is then performed under the additional constraints triggered by syntactic composition. This leads to an insertion of the corresponding argument frames into the frame of *eats*. Note that when using an elementary tree with its frame in a derivation, in order to avoid unintended identifications of feature structure, we always use a relabelling with fresh base labels.

A key advantage of syntax-driven approaches to semantic composition like those of Gardent and Kallmeyer (2003) and Kallmeyer and Romero (2008) is that they overcome the limitations of approaches which adhere solely to logical mechanisms such as functional application. In particular, the order of semantic argument filling is not specified by successive lambda abstraction or the like. Instead, semantic argument slots can be filled in any order (in particular, independently of surface word order) via unifications triggered by syntactic composition.

The frame-semantic representations introduced in this paper retain this crucial property and they show a number of additional advantages. A first point is that even plain Fillmorean role frames of the kind employed in Figure 10 provide a natural way for representing semantic arguments that are not necessarily realized in the syntax (cf. Fillmore 1986). For instance, we can assume that an *eating* frame always contains a role THEME even if the theme is not overtly expressed. More importantly, using decompositional frames for semantic modelling comes with the assumption that all subcomponents of a semantic structure (i.e., participants, subevents, paths, etc.) can be accessed via *functional* relations (attributes, features) from a controlled set of base elements (cf. Section 3). As a consequence, the semantic unifications triggered by substitution and adjunction come down, to a large extent, to feature structure unifications. In particular, if a semantic structure combines values of a feature coming from different constituents then the feature values are necessarily unified as well. (We will see examples in Sections 5 and 6 below.) Moreover, feature structure unification under constraints is computationally tractable, given that certain general conditions are respected (cf. Section 3.3 and Section 7).

Notice that the use of frames does not preclude an approach as in Gardent and Kallmeyer (2003) and Kallmeyer and Romero (2008) for modelling 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 the integration of quantifiers into frames remain to be worked out and are beyond the scope of this article.

Another approach to the syntax-semantics interface worth mentioning in this context is the synchronous TAG model of Nesson and Shieber (2006). That model employs the TAG formalism not only for the syntax of the object language but also for representing the structure of type-logical formulas on the semantic side. In our approach, by comparison, the semantic structures associated with the syntactic trees are not regarded as expressions of a formal language but as semantic models. Since feature structures are not necessarily trees, the use of synchronous TAG is not an option in our case.

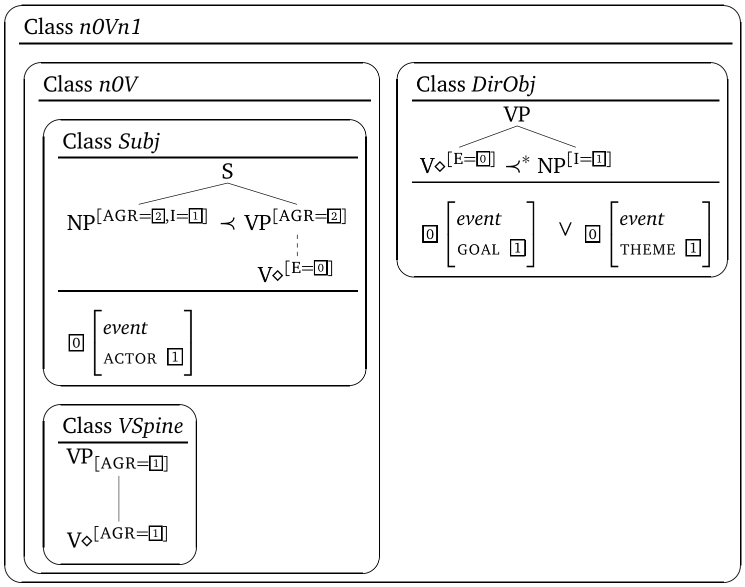


Figure 11:
MG classes with semantic frames

4.2 *Metagrammatical decomposition of elementary constructions*

Similar to the metagrammar factorization in the syntax, a decomposition of the semantic frames paired with unanchored elementary trees is possible as well. Firstly, the semantic contribution of unanchored elementary trees, i.e., constructions, can be separated from their lexicalization, and, secondly, 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.

As an example consider Figure 11 that repeats the MG classes from Figure 4, equipped with frame-semantic descriptions. The *Subj* class now tells us that the subject can contribute the actor of an event. (This is of course not the only possible contribution of a subject; the example is highly incomplete.) According to the *DirObj* class, the object NP can contribute the goal or the theme of an event. When compiling the description of *n0Vn1*, i.e., when computing its minimal models, both the actor-theme and the actor-goal combination are generated.

Several remarks are in order concerning this example. The interface feature E (“event”) is the label of the event frame of the verb. By

Figure 12:
Tree and frame descriptions in *DirObj*

Class *DirObj*

$n_1[cat : \mathrm{VP}] \wedge$
$n_2(mark : anchor)[cat : \mathrm{V}[top : [e : \boxed{0}]]] \wedge$
$n_3[cat : \mathrm{NP}[top : [i : \boxed{1}]]] \wedge$
$n_1 \rightarrow n_2 \wedge n_1 \rightarrow n_3 \wedge n_2 \prec^* n_3$

$\boxed{0} : event \wedge (\boxed{0} \cdot \textsc{goal} \triangleq \boxed{1} \vee \boxed{0} \cdot \textsc{theme} \triangleq \boxed{1})$

equating different E values on the V nodes, the corresponding event frames are unified. Concerning the status of the semantic elements in the metagrammar classes, we take them to be feature structure descriptions. This is in parallel to the syntactic parts that are tree descriptions. Incorporating a class C into a higher class C' (e.g., *Subj* into *n0V*) amounts to adding the descriptions of C as conjuncts to the syntactic and semantic descriptions of C', using fresh base labels in the descriptions if necessary.

The syntax of the tree descriptions is the one from XMG (Crabbé *et al.* 2013), a quantifier- and negation-free first order logic while the syntax of the feature structure descriptions leans on the attribute-value language introduced in Section 3.[12] To see how the tree descriptions and feature structure descriptions in the metagrammar could look, consider Figure 12 that gives the tree and frame descriptions for the class *DirObj*. In the syntax, we have free variables $n_1, n_2, \ldots$ for nodes. The conjuncts can describe the categories of nodes, special markings (for instance, anchor or foot node), and the feature values defined in the top and bottom feature structures of the nodes. The binary relations on nodes can specify dominance ($\rightarrow^*$), immediate dominance ($\rightarrow$), linear precedence ($\prec^*$) and immediate linear precedence ($\prec$). The node variables are taken to be existentially bound. In the semantics, we have base labels $\boxed{0}, \boxed{1}, \ldots$ and we allow for conjunctions and disjunctions of labelled attribute-value descriptions (cf. Section 3.3.2). In addition there are constraints on frames (cf. Section 3.3.4) that are relevant both for metagrammar compilation and for frame unification during parsing. In any minimal model computed for a metagrammar class, the frame has to satisfy these constraints.

The pairs of elementary trees and frames resulting from the compilation of the grammar are called unanchored *elementary construc-*

[12] Concerning the integration of frame descriptions into XMG, first proposals can be found in Lichte *et al.* (2013).

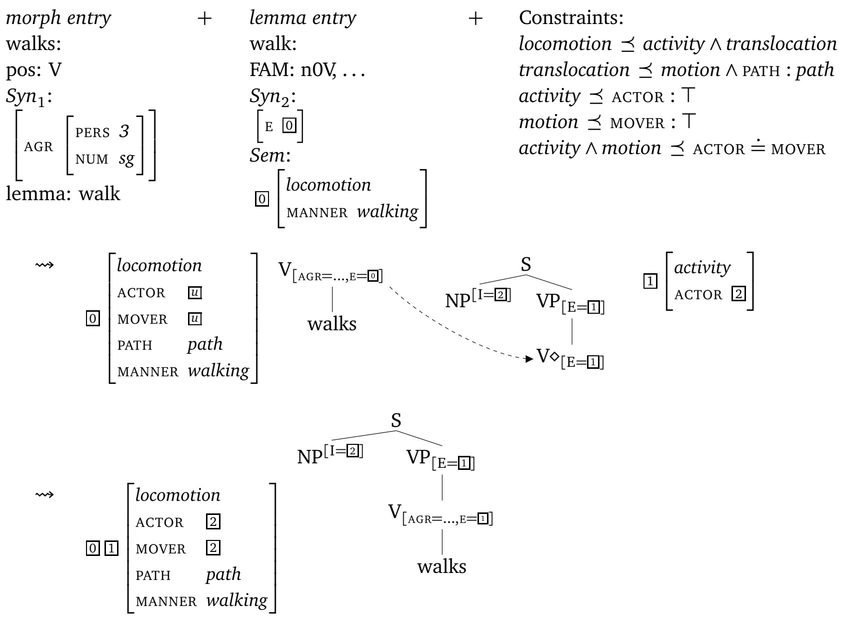


Figure 13: *Morph* and *Lemma* lexicons and lexical anchoring

tions. The step from the description in the metagrammar to the object in the grammar amounts to computing a minimal model. In the syntax, this model is such that all of its nodes and edges have to be present in the description. In the semantics, this minimal model is the smallest feature structure (with respect to subsumption) that satisfies the descriptions given by the metagrammar class and the constraints. Only those metagrammar classes that are marked as characterizing a tree family are compiled in this way (in our example only *n0V* and *n0Vn1*). Each of these classes then yields a set of unanchored elementary constructions which is an unanchored construction family of the LTAG in question.[13]

4.3 *Lexical anchoring*

In order to obtain lexicalized elementary trees, we have to fill the anchor nodes with lexical items. An example is shown in Figure 13. Lexical information is stored in a lemma lexicon and a morphological lexicon containing inflected forms. The latter gives for each form

[13] Note that in the simplified examples in this section, the class *n0V* yields only a single minimal model while *n0Vn1* leads to two minimal models since only the canonical realizations of arguments are taken into account.

the category (part-of-speech), a syntactic feature structure Syn_1 contributed by this form, and its lemma. The lemma lexicon specifies for each lemma the selected tree family (or families), again a syntactic feature structure Syn_2 and a semantic frame description *Sem*. This frame description is combined with the general constraints on frames and a minimal model is computed that is the semantics of the lexical element. In our example, the minimal model has one base label ($\boxed{0}$) and it also satisfies a path identity ($\boxed{0} \cdot \textsc{actor} \triangleq \boxed{0} \cdot \textsc{mover}$).[14] In the corresponding attribute-value matrix, we express the latter using boxed letters, here $\boxed{u}$, instead of numbers, in order to make clear that this is not a base label but just the common shorthand notation for path identity in the matrix notation of attribute-value descriptions. In parallel, the syntactic feature structures are unified and a node of the category specified in the morphological lexicon is created, decorated with the resulting syntactic feature structure. The lexical item is a daughter of this node. Lexical anchoring can then be considered as a substitution step.

Note that according to the distribution of semantic and syntactic information among the different components in Figure 13, valency information is provided by the unanchored tree (e.g., the information that the actor is contributed by the subject NP). The lexical anchor specifies its semantics, in particular its semantic arguments, but does not determine the syntactic realizations of these arguments. In other words, linking is specified in the metagrammar.

In the following sections, we apply our syntax-semantics architecture to directed motion expressions and to the dative alternation. In the metagrammar decomposition, we will be able to share several metagrammar classes in the specifications of the elementary constructions of the two phenomena.

[14] Note that the description $\boxed{0} \cdot \textsc{actor} \triangleq \boxed{0} \cdot \textsc{mover}$ is equivalent to $\boxed{0} \cdot (\textsc{actor} \doteq \textsc{mover})$.

5 APPLICATION I: DIRECTED MOTION EXPRESSIONS

5.1 *The expression of directed motion in English*

Modelling the syntax-semantics interface of directed motion expressions requires us to be explicit about a number of issues concerning the syntactic and semantic structure of such expressions, many of which have been discussed extensively in the literature. In the following, we are concerned with directional expressions in English that are constructed from verbs of motion and directional prepositional phrases (PPs). The relevant constructions include intransitive verbs of motion (14) as well as transitive verbs of caused motion and transport (15).

(14) a. Mary walked to the house.
b. The ball rolled into the goal.

(15) a. John threw/kicked the ball into the goal.
b. John pushed/pulled the cart to the station.
c. John rolled the ball into the hole.

Directional specifications are not restricted to goal expressions as in (14) and (15) but can also describe the source or the course of the path in more detail. Moreover, path descriptions can be iterated to some extent (16).

(16) a. John walked through the gate along the fence to the house.
b. John threw the ball over the fence into the yard.

Below we will use this property as an indicator for distinguishing between arguments and adjuncts.

5.1.1 Verbs of motion

It is common to distinguish between manner-encoding and path-encoding verbs of motion. The first kind of verbs (*run*, *roll*) lexically encode the manner of the motion but no path-related information, while the second kind of verbs (*enter*, *leave*) do not encode the manner but specify the direction of motion. Manner-encoding motion verbs lexically characterize activities or processes. Directional information about the goal or path can be added by appropriate adverbials, i.e., by "satellite framing" constructions in the sense of Talmy (2000b). In the following, we focus on manner-encoding verbs since our goal is to model the syntactic and semantic processes of combining directional specifications with motion expressions.

There are also motion verbs for which the actor differs from the entity that undergoes the motion. This class includes verbs of transport and caused motion (*carry*, *drag*, *push*, *throw*). As with manner-of-motion verbs, transport and caused motion verbs do not lexically specify a direction or goal. Again, directional information can be added by adverbials. The verbs of transport and caused motion are basically transitive verbs whose direct object refers to the moving entity. They can be sub-divided into different classes depending on (i) how the motion of the object is enforced by the actor and (ii) the extent to which the activity of the actor and the manner of motion are lexically specified (cf. Ehrich 1996). Concerning (i), we can distinguish between *onset causation* (*throw*, *kick*) and *extended causation* (*pull*, *drag*), following the terminology of Talmy (2000a). Verbs of the first type describe the punctual initiation of a motion event; verbs of the second type describe the continuous enforcement of the motion. As to (ii), some of the verbs in question specify the manner of motion of the moved object but say nothing about the activity of the actor (*roll*, *slide*), while for other verbs the converse is true (*pull*, *drag*).[15]

5.1.2 Syntactic issues

In the context of the LTAG analysis presented in the following sections, a crucial issue is whether to treat directional expressions such as those in (14)–(16) as complements or as adjuncts. Moreover, an argument can be determined by the base lexeme or it can be introduced by a construction or a lexical rule. For instance, sentences of type (15-c) are often characterized as *caused motion constructions* or *causative path resultatives* (Goldberg and Jackendoff 2004). That is, the directional argument is constructionally introduced. Within the LTAG approach both the basic argument structure construction and the extended construction are represented by elementary trees. The relation between these trees, and the fact that one of them builds on the other, is captured in the class structure of the metagrammar (cf. Section 5.3).

[15] Cf. Ehrich (1996) for further distinctions.

Dowty (2003) counts directional PPs as adjuncts of motion verbs since their presence is not obligatory and they do not "complete" but "modify" the meaning of the head verb. Dowty distinguishes adjuncts from elliptical complements by characterizing the latter as cases where a semantically required element must be inferred contextually. Van Valin and LaPolla (1997) classify directional PPs as "argument-adjuncts". Like adjuncts, argument-adjuncts are predicative, but they introduce an argument into the syntactic core of the head verb and they typically share an argument with the predicate encoded by the verb. A well known distinction observed by Jackendoff, Verkuyl and Zwarts, among others, is the distinction between *bounded* and *unbounded* directional PPs, which give rise respectively to telic (17-a) and atelic (17-b) event descriptions (Jackendoff 1991; Verkuyl and Zwarts 1992; Zwarts 2005).

(17) a. She walked to the brook (in half an hour/*for half an hour).
b. She walked along the brook (*in half an hour/for half an hour).

With reference to this distinction, and based on data from Dutch and other languages, Gehrke (2008) argues that bounded directional PPs are complements of the verb while unbounded PPs are adjuncts. For verbs of motion and transport, which are lexically atelic, this means that a directional expression is regarded as a complement just in case it changes the aspectual type of the expression. This assumption is compatible with the formal criterion that expressions that can be added iteratively (as, e.g., prenominal adjectives) need to be analyzed as adjuncts. In the following, we take this criterion as a preliminary working definition of adjuncthood.

5.1.3 Translocation, paths, and directions

Since the general notion of motion covers also motion in place (e.g., shaking), we use the more technical term *translocation* when we refer to the continuous change of an object's position in space (cf. Zlatev *et al.* 2010). A translocation event is by definition associated with some trajectory, trace or path of the moving entity. The approaches found in the literature differ with respect to the explicit representation of the path in the lexical semantics of the respective verbs. While in Dowty (1979) and Kaufmann (1995), paths are not part of the semantic representations of translocation verbs, Zwarts (2005) proposes a thematic function TRACE that maps motion events to the path traversed by the moving entity and in Mani and Pustejovsky (2012), it is assumed that "manner-of-motion predicates leave a trail of the motion along an implicit path, as measured over time." Similarly, Eschenbach *et al.* (2000) take paths as part of the semantics of verbs of motion. The paths referenced by verbs are here again understood as trajectories, that is, as the collection of "all points the object occupies during its course."

Paths, traces or trajectories provide a straightforward semantic link between motion verbs and directional specifications. Directionals (in English) often occur morphologically combined with locatives. For example, the directional preposition *into* specifies a path whose end point is in the interior of the goal expressed by the nominal complement of the preposition. The interior region associated with an object, as well as other regions specified by locatives, can be regarded as functional attributes of that object. We will employ this view below for the frame representations of directional prepositions.

5.2 *Analysis of directional expressions*

5.2.1 Frame representation of directed motion

The semantics of directional expressions has often been analyzed in terms of logical expressions of one kind or another (cf. Eschenbach *et al.* 2000). Our approach employs frames for semantic representation, with frames understood to be typed feature structures with relations. As explained in Section 3, this does not preclude a logical perspective but puts emphasis on the role of minimal models for semantic representation. Moreover, our frame-semantic approach takes into account semantic composition and thus goes beyond flat role frame approaches à la FrameNet. For example, the verb *throw* expresses a caused motion, that is, the described event can be analyzed as a complex causation event whose cause component consists of the activity of the thrower and whose effect is the ballistic motion of the thrown object. A possible frame-semantic representation of this decompositional analysis is shown on the right side of Figure 14, which also shows the frame for *walk*.

walk

$$\boxed{0}\begin{bmatrix} \textit{locomotion} & \\ \text{ACTOR} & \boxed{u} \\ \text{MOVER} & \boxed{u} \\ \text{MANNER} & \textit{walking} \\ \text{PATH} & \begin{bmatrix} \textit{path} & \\ \text{STARTP} & \top \\ \text{ENDP} & \top \end{bmatrix} \end{bmatrix}$$

throw

$$\boxed{0}\begin{bmatrix} \textit{onset-causation} & \\ \text{CAUSE} & \begin{bmatrix} \textit{activity} & \\ \text{ACTOR} & \top \\ \text{THEME} & \boxed{u} \\ \text{MANNER} & \textit{throwing} \end{bmatrix} \\ \text{EFFECT} & \begin{bmatrix} \textit{translocation} & \\ \text{MOVER} & \boxed{u} \\ \text{PATH} & \begin{bmatrix} \textit{path} & \\ \text{STARTP} & \top \\ \text{ENDP} & \top \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 14:
Possible frame-semantic representations of some verbs of (caused) motion

In the given representations, a good part of the lexical meaning is condensed in the types or left implicit. For instance, the precise way of how the actor induces the (ballistic) motion of the object in throwing events is simply encoded by an atomic value of the attribute MANNER. Similarly, the causation type of throwing events is encoded by the type *onset-causation* of the main event. A more explicit representation would include the temporal characteristics of an onset causation, i.e., punctuality and temporal precedence of the causing event. Notice that the path or trace of the moving entity is made explicit by the frames in Figure 14. As argued above, the path of the moving object is an inherent semantic component of translocation events; the path provides the anchor for directional specifications. It is important to keep in mind that the presence of the PATH attribute in the frame representation of, say, *walk* does not imply by any means that *walk* lexically encodes information about the path of the movement.[16] Concerning semantic roles, we allow multiple descriptions of an event participant in a single frame. Each *motion* event has a participant that moves, the

[16] It is instructive to compare our use of the attribute PATH with the corresponding semantic role (frame element) of the frames 'Motion', 'Motion_directional', and 'Self_motion' in the Berkeley FrameNet database. In our decompositional approach, the path (or trace, or trajectory) is an inherent component of translocation events. In FrameNet, the 'Path' element is directly related to path descriptions such as *down the stairs*, *along the brook*, etc.; see Section 5.2.3 for our analysis of such expressions. Moreover, the relevant FrameNet frames come with core elements 'Goal', 'Direction', 'Source', etc., which is not the case for the representations shown in Figure 14. The underlying intuition is that while the path of a translocation has an end point, it is not part of the concept *per se* to have a goal.

Figure 15: Frame examples for directional prepositions

to

$$\boxed{0}\begin{bmatrix}\textit{event} & \\ \text{PATH} & \begin{bmatrix}\textit{path} & \\ \text{ENDP} & \boxed{v}\end{bmatrix}\end{bmatrix}$$
$$\boxed{1}\begin{bmatrix}\text{AT-REGION} & \boxed{w}\end{bmatrix}$$
$$\textit{part-of}(\boxed{v},\boxed{w})$$

into

$$\boxed{0}\begin{bmatrix}\textit{event} & \\ \text{PATH} & \begin{bmatrix}\textit{path} & \\ \text{ENDP} & \boxed{v}\end{bmatrix}\end{bmatrix}$$
$$\boxed{1}\begin{bmatrix}\text{IN-REGION} & \boxed{w}\end{bmatrix}$$
$$\textit{part-of}(\boxed{v},\boxed{w})$$

along

$$\boxed{0}\begin{bmatrix}\textit{event} & \\ \text{PATH} & \begin{bmatrix}\textit{path} & \\ \text{REGION} & \boxed{v}\end{bmatrix}\end{bmatrix}$$
$$\boxed{1}\begin{bmatrix}\text{AT-REGION} & \boxed{w}\end{bmatrix}$$
$$\textit{part-of}(\boxed{v},\boxed{w})$$

MOVER. If the event is an activity, this participant becomes at the same time the ACTOR.

The frame representation of directional prepositions follows the outline described in the previous section. The basic idea is that frames associated with directional prepositions can unify with frames of translocation, which gives rise to frames that express directed motion. For example, the frame for the preposition *into* shown in the middle of Figure 15 represents (directed) motion to the interior region of an object $\boxed{1}$ which is denoted by the nominal complement of the preposition. The frame description in the last line of the figure encodes the condition that the end point of the path or trajectory generated by the motion is in fact a mereological part of the region in question. In the matrix notation, now extended by relational descriptions, boxed letters serve again as a shorthand notation for (labelled) attribute paths. That is, *part-of*($\boxed{v}$, $\boxed{w}$) is short for the labelled description in (18).

(18) $\langle \boxed{0} \cdot \text{PATH REGION}, \boxed{1} \cdot \text{AT-REGION} \rangle : \textit{part-of}$

Note that the intended meaning of *part-of* has to be spelled out by appropriate constraints. In the case at hand, this includes transitivity (cf. (13)), reflexivity, and anti-symmetry, as well as type constraints on the domain and range of the relation. So far, our impression is that all (non-functional) relations we need are of this sort. In Section 7, we will say a few words about applying such constraints during unification.

The semantic representations described so far allow us to introduce the basic ideas of a syntax-driven semantic frame composition in the following sections. In a fully developed theory of frame-semantic representations for events, the types and features used in the frames are systematically related to each other by constraints. For instance, the inheritance hierarchy of the event types introduced so far would look like the one depicted in Figure 16. Each type comes with feature declarations that formulate constraints on the frames of this type.

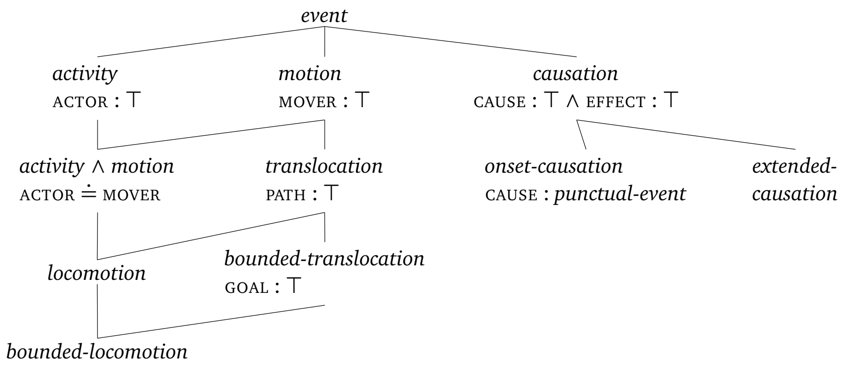


Figure 16: Partial sketch of constraints on event types and attributes

The constraints in Figure 16 specify for instance that frames of type *causation* have a CAUSE and an EFFECT attribute, and that the value of the CAUSE attribute of *onset-causation* events is of type *punctual-event.*

5.2.2 Intransitive directed motion constructions

This section deals with the combination of motion verbs and directional PPs as shown in (19).

(19) a. Mary walked/ran to/into the house.
b. Mary walked/ran along the river.
c. Mary walked/ran over the bridge along the fence through the meadows.

Recall that our criterion for treating a constituent as an argument or an adjunct is iterability. Constituents that cannot be iterated and that add a semantic role (no matter whether the role is already present in the frame contributed by the verb) are taken to be complements in the sense that their integration into the unanchored tree for the verb is part of the metagrammatical specification of elementary trees. For this reason, the examples in (19-a) are treated as PP complements while the PP in (19-b) is considered an adjunct. PPs of the type in (19-b) can be iterated as can be seen in (19-c).

In the PP complement case, the preposition is not part of the elementary tree of the verb since it is not determined by the verb. This is in contrast to constructions where a specific preposition is treated as a coanchor of the elementary tree. An example is the elementary tree for phrasal verbs such as *subscribe to*, as in *Mary subscribes to a*

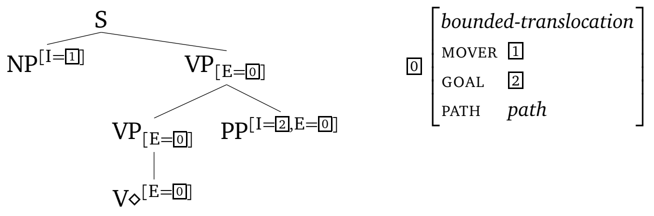


Figure 17: Unanchored tree and semantics of *n0Vpp(dir)* construction

linguistics journal, where the preposition *to* is taken to be a coanchor of the elementary tree.

As explained in Section 5.2.1, we assume that the motion verbs in (19) define a *locomotion* that has a certain path (trace, trajectory) associated with it. This path has a start and an end point. In the directed motion construction, the additional PP adds a further argument with the semantic role GOAL. The way this goal combines with the path, i.e., whether it is its end point, whether it adds a direction to the path, etc., depends on the preposition.

The unanchored elementary tree for an intransitive verb with an additional directional PP is given in Figure 17. Note that we assume a binary left-branching structure for the VP, i.e., every argument inside the VP is the right sister of a VP node and the lowest VP node immediately dominates the verbal anchor. This allows for the adjunction of modifiers between the verb and the directional PP as in (20).

(20) He ran quickly to the river.

The decoration of the elementary tree with the interface features I and E ensures that the substitutions of the subject NP and the object PP fill the corresponding argument roles and, furthermore, that adjunctions of modifiers to the VP node extend the event frame $\boxed{0}$.

The preposition determines the relation between the path of the motion and the goal. Figure 18 shows the elementary trees of different directional prepositions (cf. Figure 15). We assume that objects such as *the house* have a certain topological structure. They come with different types of regions, an at-region that contains all points that can be said to be *at* the object, an in-region that determines the space that constitutes the inner part of the object, etc. The preposition *to* makes reference to the at-region of an object; it expresses that the endpoint of the path must be contained in the at-region of the entity denoted by the NP complement of the preposition. Similarly, *into* expresses that

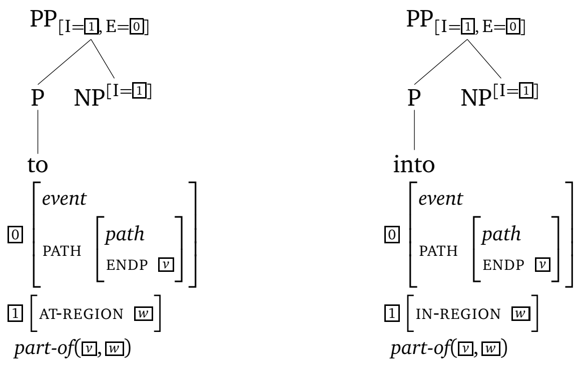


Figure 18: Elementary trees for prepositions

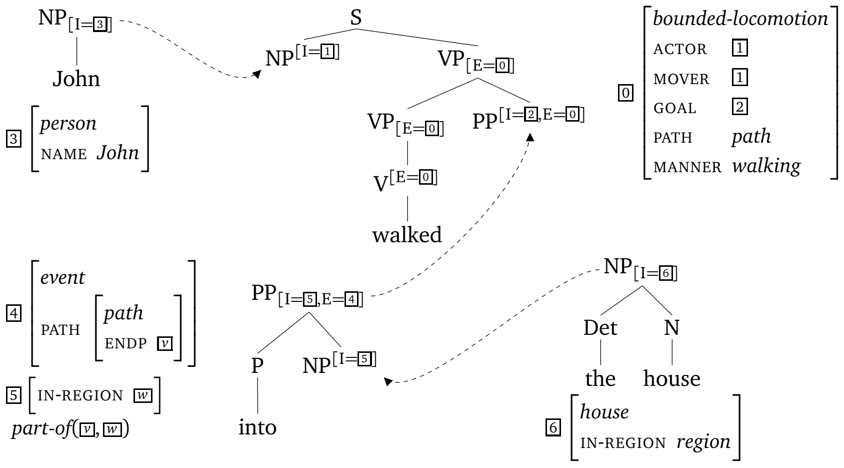


Figure 19: Derivation of (21)

the endpoint must be contained in the in-region of the entity.

As an example, let us consider the derivation of (21). Figure 19 shows the elementary constructions involved and how they are combined.

(21) John walked into the house.

The representation for *the house* comes with an in-region (among others). (The composition of the determiner and the noun into the NP *the house* is left aside in this example.) The preposition *into* links the in-region to the end point of the path traversed throughout the walking activity. The various substitutions give rise to the following identities: $\boxed{1} \triangleq \boxed{3}$, $\boxed{2} \triangleq \boxed{5} \triangleq \boxed{6}$ and $\boxed{0} \triangleq \boxed{4}$. With the corresponding unifications, the resulting frame is the one given in Figure 20.

Figure 20:
Resulting frame for (21)

$$\boxed{0}\begin{bmatrix} \textit{bounded-locomotion} & \\ \textsc{actor} & \boxed{1}\begin{bmatrix} \textit{person} & \\ \textsc{name} & \textit{John} \end{bmatrix} \\ \textsc{mover} & \boxed{1} \\ \textsc{goal} & \boxed{2}\begin{bmatrix} \textit{house} & \\ \textsc{in-region} & \boxed{w} \end{bmatrix} \\ \textsc{path} & \begin{bmatrix} \textit{path} & \\ \textsc{endp} & \boxed{v} \end{bmatrix} \\ \textsc{manner} & \textit{walking} \end{bmatrix}$$
$$\textit{part-of}(\boxed{v}, \boxed{w})$$

Motion verbs that are turned into a directed motion by adding a goal and a path (such as in (22)) differ from verbs of locomotion (as in (21)) with respect to their lexical semantics.

(22) Mary danced into the room.

Walk comes with a path while *dance* does not. The lexical frame for *dance* is shown in Figure 21. When combining it with the unanchored construction tree, the path attribute is added and the goal argument is linked to the PP.

Figure 21:
Frame for *dance*

$$\boxed{0}\begin{bmatrix} \textit{activity} \wedge \textit{motion} & \\ \textsc{actor} & \boxed{u} \\ \textsc{mover} & \boxed{u} \\ \textsc{manner} & \textit{dancing} \end{bmatrix}$$

5.2.3 Path modification

Now let us turn to the case where the directional PP is an adjunct that gives an additional specification of the path of the event as in (19-b). In these cases, the verb of locomotion anchors an intransitive activity tree. As an example, consider the derivation of (23). Figure 22 shows the adjunction of the *along* elementary tree into the elementary intransitive construction of *walked* (see Figure 13 for the anchoring step for this construction). The frame linked to *along* expresses that the entity denoted by the NP within the PP has an at-region that must contain the entire region of the path. Note that the frame contributed by the preposition does not have a unique root. The reason

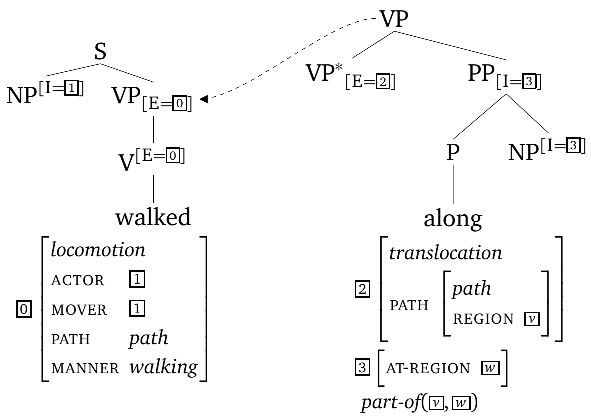


Figure 22: Derivation of (23)

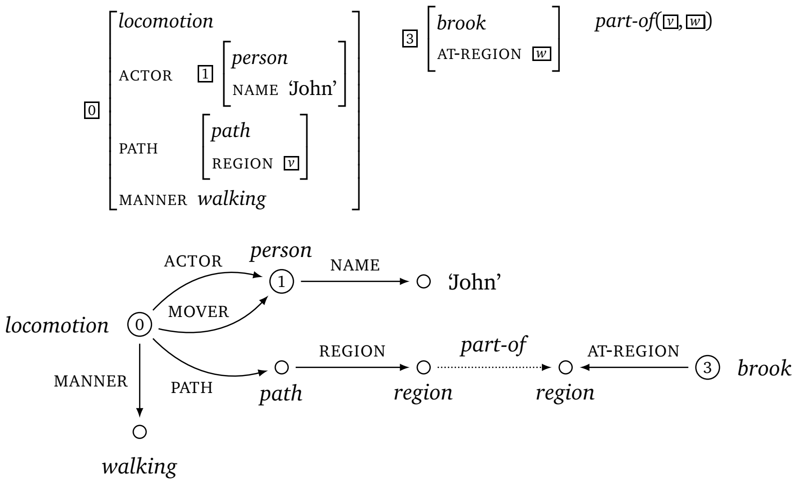


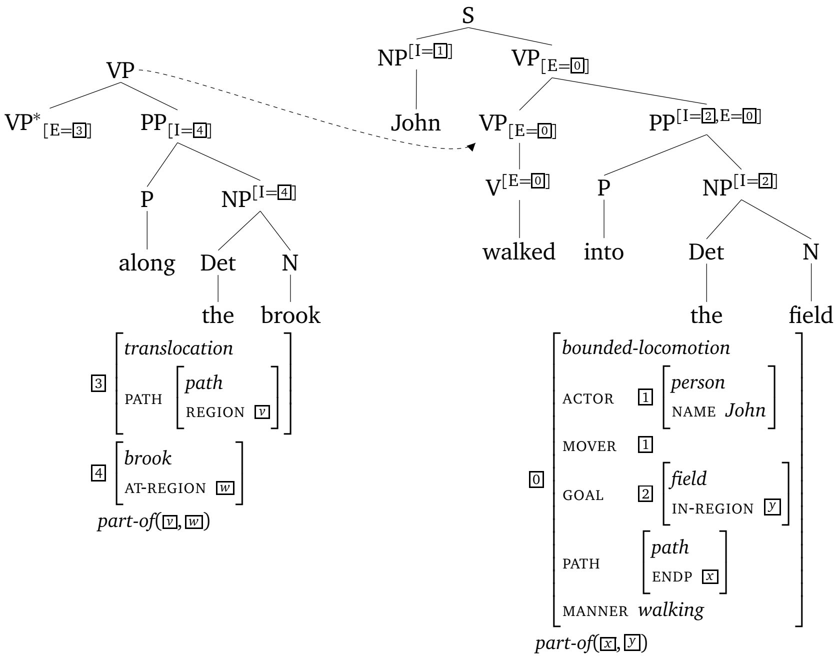
Figure 23: Resulting frame for (23)

is that the NP does not contribute an argument and therefore it does not fill a semantic role slot. The link between the object denoted by the NP and the walking activity concerns only the at-region of the former.

(23) John walked along the brook.

As a result, when combining further with the elementary trees for *John* and *the brook*, we obtain the frame in Figure 23. In addition to the attribute-value matrix, the figure also shows the corresponding feature structure depicted as a graph. The graph shows more clearly that we have more than one root node in this frame and, furthermore,

Figure 24: Derivation of (25)



if we disregard the non-functional relation *part-of*, the frame is not even a connected graph.

Obviously, examples with motion verbs such as *dance* which do not lexically specify translocation work as well, cf. (24). In these cases, the preposition introduces the path.

(24) Mary danced along the fence.

As a last example, let us consider a combination of argument directional PPs and adjoining directional PPs.

(25) John walked along the brook into the field.

Figure 24 illustrates the derivation step that combines the PP *along the brook* with the rest of the sentence. The unification of $\boxed{3}$ and $\boxed{0}$ triggered by the adjunction gives rise to the frame shown in Figure 25, which combines the two constraints on the path contributed by the two PPs: The entire path (i.e., its REGION) must be contained in the AT-REGION of the brook and the ENDP (endpoint) of the path must be contained in the IN-REGION of the field.

$$\boxed{0}\begin{bmatrix} \textit{bounded-locomotion} & \\ \text{ACTOR} & \boxed{1}\begin{bmatrix}\textit{person} & \\ \text{NAME} & \textit{John}\end{bmatrix} \\ \text{MOVER} & \boxed{1} \\ \text{GOAL} & \boxed{2}\begin{bmatrix}\textit{field} & \\ \text{IN-REGION} & \boxed{y}\end{bmatrix} \\ \text{PATH} & \begin{bmatrix}\textit{path} & \\ \text{ENDP} & \boxed{x} \\ \text{REGION} & \boxed{v}\end{bmatrix} \\ \text{MANNER} & \textit{walking}\end{bmatrix} \qquad \boxed{4}\begin{bmatrix}\textit{brook} & \\ \text{AT-REGION} & \boxed{w}\end{bmatrix}$$

$$\textit{part-of}(\boxed{x},\boxed{y}) \quad \textit{part-of}(\boxed{v},\boxed{w})$$

Figure 25:
Resulting frame for (25)

5.2.4 Caused motion constructions

We now turn to verbs of transport and caused motion as exemplified in (26).

(26) a. Mary threw the ball into the hole.
b. Mary pulled the cart along the river.
c. Mary kicked the ball along the line into the goal.

Our proposal for the unanchored construction and its semantics is shown in Figure 26. The difference relative to the intransitive directed motion construction *n0Vpp(dir)* discussed above is that now the theme, i.e., the entity denoted by the direct object is moving. This movement is the effect of an action performed by the actor that affects the theme. Therefore the directed motion of the moving entity is represented as the effect of a causation whose cause is an action performed by the subject.

A difficulty with this construction is that the PP argument and directional PP modifiers need to access the embedded translocation event while other modifiers might want to access the main event. As a solution that makes both accessible and that distinguishes them, we propose to use the feature E on the PP argument slot for the embedded translocation event (here $\boxed{4}$) and the E feature on the VP node for the highest event (here $\boxed{0}$). This allows for the insertion of modifiers between the verb and the PP that modify the higher event, as in (27). Such modifiers adjoin to the lower VP node.

(27) Paul threw the ball immediately into his opponent's goal.

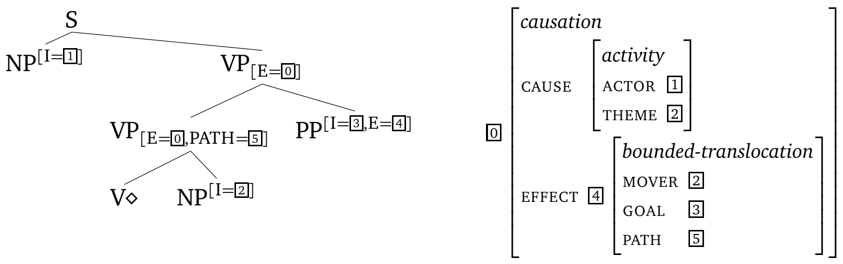


Figure 26: Unanchored tree and semantics of *n0Vn1pp(dir)* construction

However, we also want to allow path modifiers between the verb and the PP that modify the embedded event as in (26-c). A modifier such as *along the line* does not contribute a participant to the motion event, in contrast to the case of the directional argument PP. It only adds some further specification about the path of this event. Therefore, it is actually enough for it to have access to the path and not to the event this path belongs to. For this reason, we propose to add a new interface feature PATH on the syntactic trees. This feature is accessible at nodes where path modifiers could adjoin, in particular on the VP node preceding the PP argument in Figure 26. The feature PATH has to appear as well on the VP nodes in *n0Vpp(dir)* trees, except that here the path is part of the main event.

With the additional interface feature PATH, we have to revise the directional PP modifier trees; they now access the path they refer to via this feature. The associated frame relates PATH to the AT-REGION of the NP; see Figure 27.

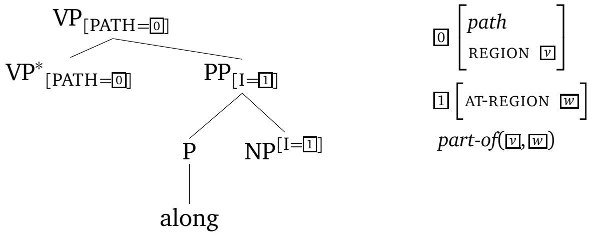


Figure 27: Revised elementary tree for *along*

5.3 *MG decomposition of directed motion and caused motion constructions*

We have already introduced metagrammar classes for elementary intransitive and transitive constructions in Section 4.2. We will now ex-

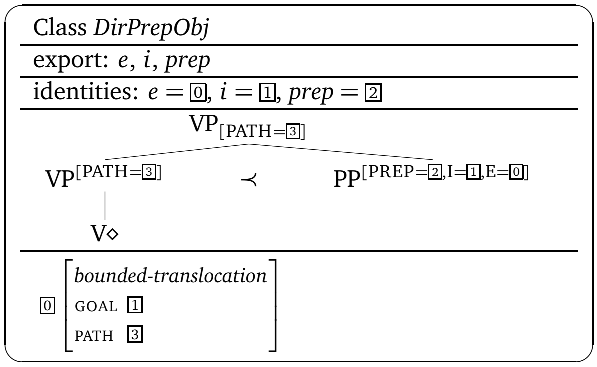

Class *DirPrepObj*
export: *e*, *i*, *prep*
identities: $e = \boxed{0}$, $i = \boxed{1}$, $prep = \boxed{2}$
$VP_{[PATH=\boxed{3}]}$
$VP^{[PATH=\boxed{3}]}$
$\prec$
$PP^{[PREP=\boxed{2},I=\boxed{1},E=\boxed{0}]}$
V◇
$\boxed{0}$ *bounded-translocation*
GOAL $\boxed{1}$
PATH $\boxed{3}$

Figure 28:
MG class for a directional PP object

tend these classes in order to cover the directed motion constructions presented in the previous section.

The MG classes for these constructions are given in Figures 28 and 29. In addition to what we have seen in Section 4.2, we now allow the definition of export variables within a class. These export variables are visible to other classes using this class and can then be used to identify nodes between classes. The class for the directional prepositional object, *DirPrepObj* contributes the goal of some directed motion event. The export variable *prep* is not relevant for the directed motion case; it will serve to constrain the preposition in the prepositional object case treated in Section 6 below. Crucially, in the *DirPrepObj* class, the event described here need not be the event denoted by the verb. Therefore, the event identifier $\boxed{0}$ is not linked via an E feature to the verb. Depending on the context in which we use this class, this event is either the main event of the verb (28-a) or an embedded event (28-b).

(28) a. Mary walked into the house
b. Mary threw the ball into the hole.

The solution is to make the event in question accessible via the declaration of export variables. For the combination of *DirPrepObj* with the intransitive or with the transitive class, we assume that the two classes *n0V* and *n0Vn1* in turn have an export variable *e* that is the event frame variable linked to the V node. The class *n0Vpp(dir)* (Figure 29, left side) for constructions without a direct object (as in (28-a)) is rather simple since the directional PP adds a participant to the event denoted by the verb. Obviously, this yields the unanchored tree for

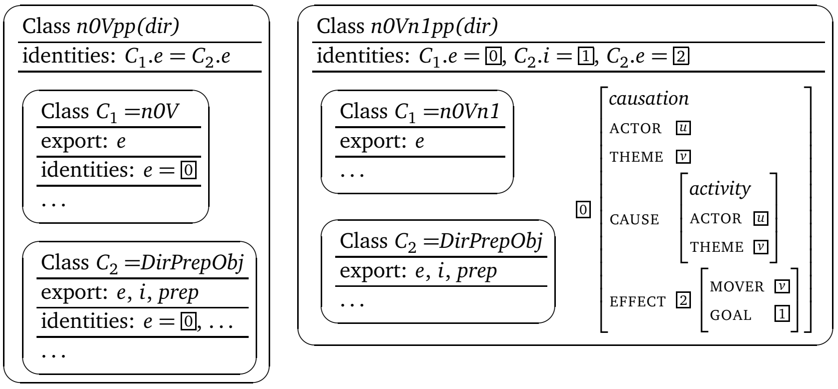

Class *n0Vpp(dir)*
identities: $C_1.e = C_2.e$
Class C_1 =*n0V*
export: *e*
identities: $e = \boxed{0}$
...
Class C_2 =*DirPrepObj*
export: *e*, *i*, *prep*
identities: $e = \boxed{0}$, ...
...
Class *n0Vn1pp(dir)*
identities: $C_1.e = \boxed{0}$, $C_2.i = \boxed{1}$, $C_2.e = \boxed{2}$
Class C_1 =*n0Vn1*
export: *e*
...
Class C_2 =*DirPrepObj*
export: *e*, *i*, *prep*
...
$\boxed{0}$ *causation* ACTOR $\boxed{u}$ THEME $\boxed{v}$ CAUSE *activity* ACTOR $\boxed{u}$ THEME $\boxed{v}$ EFFECT $\boxed{2}$ MOVER $\boxed{v}$ GOAL $\boxed{1}$

Figure 29: MG classes for directed motion constructions

walked as used in (28-a). Cases such as (28-b) are more complex since they involve an embedding of the event in which the directional PP participates. The class *n0Vn1pp(dir)* (Figure 29, right side) is for constructions as in (28-b) which have a direct object and a directional PP. It identifies the directed motion event of the PP with the event embedded under EFFECT, via the *e* export variable of the *DirPrepObj* class (identity $C_2.e = \boxed{2}$).

5.4 *Summary*

In this section, we have presented an analysis of verbs and constructions of directed motion and caused motion using LTAG and frame semantics. We have shown how to decompose elementary constructions into, first, the unanchored tree and its semantics and the lexical entry and, second, into smaller syntactic and semantic fragments of which the unanchored elementary construction is built.

We have seen that the metagrammar architecture of LTAG allows us to capture components which several elementary constructions have in common, for instance the class *DirPrepObj*, which contributes the syntactic slot of the goal of a *bounded-translocation*. This small piece of syntactic structure and related meaning can be used in different ways in larger classes, depending on the embedding of the *bounded-translocation* event.

The decoration of the syntactic trees with interface features allows us to access different nodes in a semantic frame, making them accessible for semantic composition. Summarizing, this first case study has demonstrated the flexibility with respect to semantic composition and the capability of factorization and generalization offered by our LTAG syntax-semantics interface architecture.

6 APPLICATION II: THE DATIVE ALTERNATION

6.1 *Caused possession vs. caused motion construction*

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 (29-a) and (29-b), respectively. The PO construction is closely related to the caused motion construction discussed in the previous section, except that the preposition in the PO construction is always *to*.

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

The two constructions are traditionally associated with a 'caused possession' (29-a) and 'caused motion' (29-b) interpretation, respectively (see, e.g., Goldberg (1995)). These two interpretations have often been analyzed by decompositional schemas of the type shown in (30-a) and (30-b).

(30) a. $[\,[x \text{ ACT}] \text{ CAUSE } [y \text{ HAVE } z]\,]$
b. $[\,[x \text{ ACT}] \text{ CAUSE } [z \text{ GO TO } y]\,]$

In a similar vein, Krifka (2004) uses event logical expressions of the sort shown in (31) for distinguishing the two interpretations. Note that (31-b) is very close to the semantic frame used in the preceding sections for caused motion.[17]

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

The differences between the DO and the PO constructions and their respective interpretations span a wider range of options than those

[17] Recall the difference between the relational uses of CAUSE in (30) and (31) and our use of CAUSE as an event attribute that singles out the cause component of a causation event; cf. Section 3.2.

described so far. Rappaport Hovav and 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.).[18] 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 in which they occur. The *send* and *throw* verbs differ in the meaning components they lexicalize: *send* lexicalizes caused motion towards a goal, whereas *throw* encodes the caused initiation of motion and the manner in which this is done. A goal is not lexicalized by *throw* verbs, which accounts for the larger range of directional PPs allowed for these verbs (cf. Section 5.2.4).

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", while 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 goal 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 this article, 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

[18] 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* (cf. Koenig and Davis 2001).

	lexical meaning					PO pattern	DO pattern
	#args	result	punctual	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

possession is concerned. All other alternating ditransitive verbs show such a difference since only the DO pattern implies prospective possession.[19] Beavers (2011) draws a distinction between different types of caused possession verbs. Verbs such as *give* encode pure caused possession without motion necessarily being involved. Verbs 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 goal (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, i.e., they do not lexicalize a goal, and they are usually regarded as being incompatible with the DO pattern.[20]

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

[20] 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 indicat-

In sum, the DO and PO constructions strongly interact with the lexical semantics of the verb.[21] Table 1, which builds on Beavers' analysis, gives an overview of the contribution of the lexicon and the constructions. Prospectivity is indicated by '$\Diamond$'.

6.2 *Analysis of DO and PO*

6.2.1 Frame representations

For some of the verbs listed in Table 1, possible frame semantic representations are given in Figure 30. We have added a further event type *undergoing* which comes with a participant role THEME (32-a) and which is incompatible with *activity* (32-b). The main purpose of this extension in the current context is to characterize the MOVER of a non-active motion event as a THEME (32-c), in much the same way as the MOVER of active motion has been co-classified as ACTOR (cf., e.g., Figure 16).

(32) a. $\textit{undergoing} \preceq \textit{event} \wedge \text{THEME} : \top$
b. $\textit{undergoing} \wedge \textit{activity} \preceq \bot$
c. $\textit{undergoing} \wedge \textit{motion} \preceq \text{THEME} \doteq \text{MOVER}$

Consider the frame for *send*. The bounded translocation subframe encodes motion towards the goal without necessarily implying arrival. The motion is non-active, i.e., of type *undergoing*, which means that the mover is the theme of the event. The representation for *throw* differs from that for *send* in that *throw* lexicalizes a certain manner of activity.

ing accompanied motion like *carry* has been called into question by Bresnan and Nikitina (2010) on the basis of corpus evidence. Building on Krifka's approach, Beavers (2011, pp. 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 of the theme at the final point of the event. Conditions of this type would naturally go into a more detailed frame-semantic analysis elaborating on the ones given in this paper.

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

send

$$\boxed{0}\left[\begin{array}{ll} \textit{causation} & \\ \textsc{cause} & \left[\begin{array}{ll}\textit{activity} & \\ \textsc{actor} & \top\end{array}\right] \\ \textsc{effect} & \left[\begin{array}{ll}\textit{bounded-translocation} \wedge \textit{undergoing} & \\ \textsc{theme} & \boxed{u} \\ \textsc{mover} & \boxed{u} \\ \textsc{goal} & \top\end{array}\right]\end{array}\right]$$

throw

$$\boxed{0}\left[\begin{array}{ll} \textit{onset-causation} & \\ \textsc{cause} & \left[\begin{array}{ll}\textit{activity} & \\ \textsc{actor} & \top \\ \textsc{theme} & \boxed{u} \\ \textsc{manner} & \textit{throwing}\end{array}\right] \\ \textsc{effect} & \left[\begin{array}{ll}\textit{translocation} \wedge \textit{undergoing} & \\ \textsc{theme} & \boxed{u} \\ \textsc{mover} & \boxed{u}\end{array}\right]\end{array}\right]$$

give

$$\boxed{0}\left[\begin{array}{ll} \textit{causation} & \\ \textsc{cause} & \left[\begin{array}{ll}\textit{activity} & \\ \textsc{actor} & \top\end{array}\right] \\ \textsc{effect} & \left[\begin{array}{ll}\textit{change-of-possession} & \\ \textsc{theme} & \boxed{u} \\ \textsc{recipient} & \boxed{v} \\ \textsc{result} & \left[\begin{array}{ll}\textit{possession} & \\ \textsc{possessor} & \boxed{v} \\ \textsc{possessed} & \boxed{u}\end{array}\right]\end{array}\right]\end{array}\right]$$

Figure 30: Possible frame representations for some of the lexical items in Table 1.

Moreover, it is inherent in the given representation that the destination of the entity thrown is not part of the lexical meaning of *throw*. Concerning the semantics of *give*, we have a caused change of possession that results in an actual *possession* state. The embedded event type *change-of-possession* introduces a participant role RECIPIENT (33-a).

(33) a. $\textit{change-of-possession} \preceq \textit{undergoing} \wedge \textsc{recipient} : \top$

b. $\textsc{recipient} : \top \preceq \textsc{recipient} \doteq \textsc{goal}$

We furthermore assume that a RECIPIENT can be described as a kind of GOAL (33-b).

6.2.2 Constructions

The PO construction is analyzed as a caused motion construction with a *to*-PP. Some verbs allowing the DO-PO alternation can also be used in a general caused motion construction (tree family *n0Vn1PP(dir)*); see (34).

(34) a. He sends the boy into the house.

b. He throws the ball into the basket/at the boy.

The base trees of the DO and PO families involved in the alternation are depicted on the left side of Figure 31. The fact that the preposition is required to be *to* is encoded in the PREP feature on the PP. The DO tree is flatter since in this construction, modifiers between the verb and the first NP object or between the first NP and the second are not possible.

The semantics of the DO construction is a (prospective) caused possession meaning which gets further constrained when being linked to a specific lexical anchor. More concretely, a RESULT feature is possible but not obligatory for events of type *change-of-possession*. Figure 31a shows how the unanchored tree is linked to its semantic frame. Again, the identities between the interface features I in the syntactic tree 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 Figure 31b.

6.2.3 Lexical anchoring

Anchoring the trees from Figure 31 means that the lexical anchor is substituted into the anchor node and thereby contributes parts of a semantic frame. The example in Figure 32 shows the lexical anchoring of the PO construction with the anchor *throws*. The resulting anchored elementary tree has a semantic frame that is the unification of the frames $\boxed{4}$ and $\boxed{0}$. In a similar way, caused possession verbs like *give* can anchor the DO construction.

Now, what happens if *throw* or *send* try to anchor the DO construction? That is, how can, e.g., the frame of *send* (cf. Figure 30) that represents a caused directed motion be unified with the frame of the DO construction which represents a caused change of possession (Figure 31a). The meaning of the combined frame (i.e., of the DO construction anchored with *sends*) is, roughly, a causation with effects along two dimensions: there is a directed motion of the theme and at the same time the theme undergoes a change of possession. In the model presented here, this double perspective can be captured by assuming that the event types *bounded-translocation* and *change-of-possession* do not exclude each other.[22] Hence, the effected event can be charac-

[22] An alternative solution that keeps these types disjoint would be to use set-

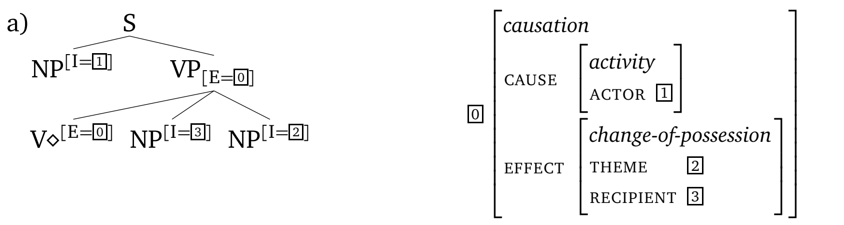
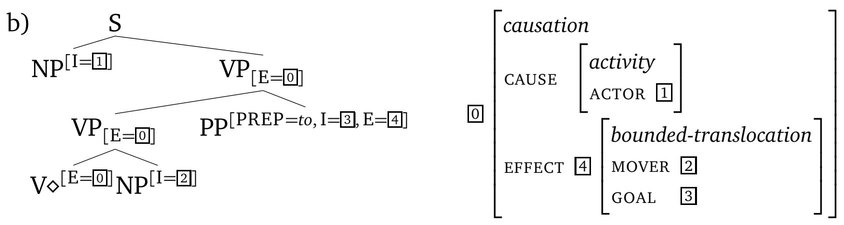



Figure 31: Unanchored elementary trees and semantics of the DO and PO constructions

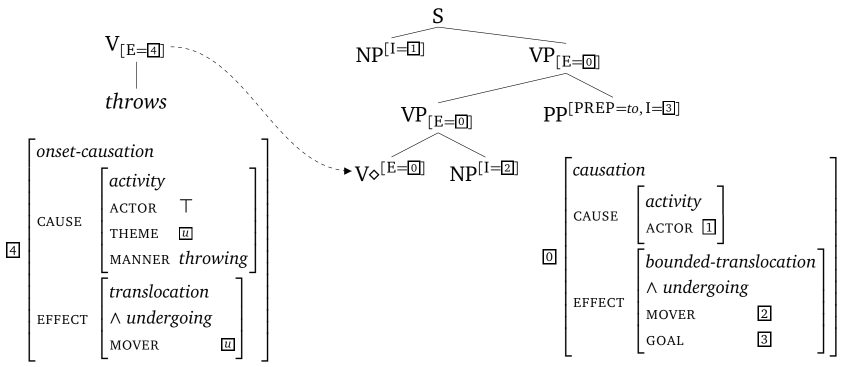


Figure 32: Lexical selection of the elementary tree for *throws* in the PO construction

terized by a conjunction of these types. The appropriate matching of the semantic roles is enforced by the constraints (32-c) and (33-b). The result of the unification is given in Figure 33. A participant can thus have different semantic roles that reflect the ways in which it is involved in the different characterizations of the event.

6.3 *MG decomposition*

We will now consider the metagrammar classes needed for the dative alternation, i.e., for the DO and PO constructions. The factorization

valued attributes, which requires however a specific definition of subsumption for these sets. Another option could be to use different attributes for the different dimensions along which an event is described, for instance LOC-ASPECT and POSS-ASPECT in our case.

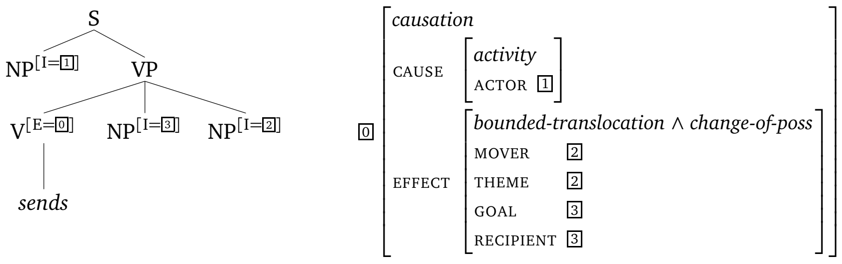


Figure 33: Anchored tree for *sends* with the DO construction

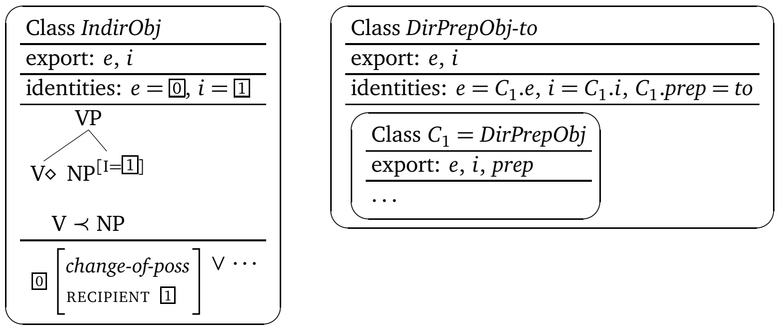


Figure 34: MG classes for indirect object and directional prepositional object

of grammatical information in the metagrammar enables us to generalize from the two phenomena that we deal with in this paper and to use the class for directional PP arguments given in Section 5.3 in both the prepositional object construction of the dative alternation and constructions with verbs of directed motion.

The classes for the indirect object and the prepositional object are given in Figure 34. A dative object (class *IndirObj*) can contribute the recipient of a change of possession event. This is not the only way an indirect object can contribute a participant to an event. Note that, according to the syntactic tree description, the NP node must immediately follow the verb node, in contrast to the NP node of a direct object that stands only in a (not necessarily immediate) linear precedence relation to the verb. The class *DirPrepObj-to* simply uses the *DirPrepObj* given above and specifies in addition that the preposition has to be *to*.

Crucially, in these classes, as in the directional PP class, the event described in the class need not be the event denoted by the verb. Therefore, again, the event identifier $\boxed{0}$ is not linked via an E feature to the verb. Depending on the context in which we use the classes, this event is either the event of the verb or an embedded event.

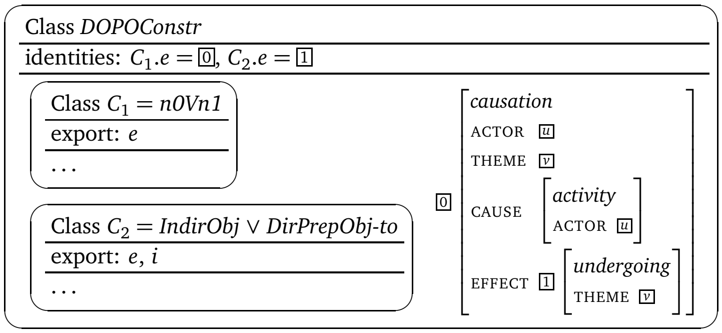


Figure 35:
MG class for
the alternation
between DO and PO
constructions

Now let us inspect the way these classes combine with the transitive verb class *n0Vn1* in order to build our unanchored DO and PO trees. The class for the DO-PO alternation is given in Figure 35. We can capture both constructions in a single class that tells us that we combine the transitive class with either *IndirObj* or *DirPrepObj-to*. The result is a causation involving an action performed by the actor of the transitive class. This causation has an effect on the theme of the transitive class. The nature of this effect depends on the class used for the third argument. In the DO case (*IndirObj*) it is a change-of-poss while in the PO case (*DirPrepObj-to*) it is a directed motion.

Note that the PO construction is actually slightly more restricted than the caused motion construction with a directional PP, not only with respect to the prepositions allowed. The NP of the directional PP, even if it is a location, receives a kind of institutional reading. Therefore, purely locational specifications such as *the house* are odd here:

(35) a. She sent the package to London.
b. ?She sent the letter to the house.

Such constraints could be modelled via restrictions on the possible goals. Either the type could be restricted or certain features could be required for the GOAL value in the *DOPOConstr* class. For this paper, we leave the detailed modelling of these constraints aside.

6.4 *Further issues*

It goes without saying that a full account of the dative alternation has to cope with many 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 and 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 our grammar framework seems to be well-suited for implementing requirements of this sort, they are beyond the scope of our study here, which is primarily concerned with modelling the influence of narrow verb classes on constructional form and meaning.

7 COMPLEXITY CONSIDERATIONS

Concerning computational complexity, we have to consider the two main processing components of our architecture: metagrammar compilation and parsing. During metagrammar compilation, we compute a finite set of minimal models. For the syntactic tree, the search for minimal models means that all nodes and edges in the models have to be present in the descriptions given in the metagrammar. The current XMG implementation (Crabbé *et al.* 2013) already provides this model-building step. For the frame descriptions, a frame must first be built containing all nodes and edges described in the MG classes. In a second step, the set of constraints on frames has to be checked on this particular frame, which might lead to additional type assignments and even additional edges and nodes. In other words, we have to compute the closure with respect to the constraints. The tractability of this step depends heavily on the nature of these constraints. For instance, in order to ensure the existence of finite minimal models, we need to avoid constraint loops (see also Carpenter 1992, pp. 95ff). Tying down the exact conditions on the constraint system that make it well-behaved with respect to model construction is part of current research. Note that the metagrammar compilation can be preprocessed and is independent from the size of the input string. Therefore its complexity matters less than the complexity of unification during parsing.

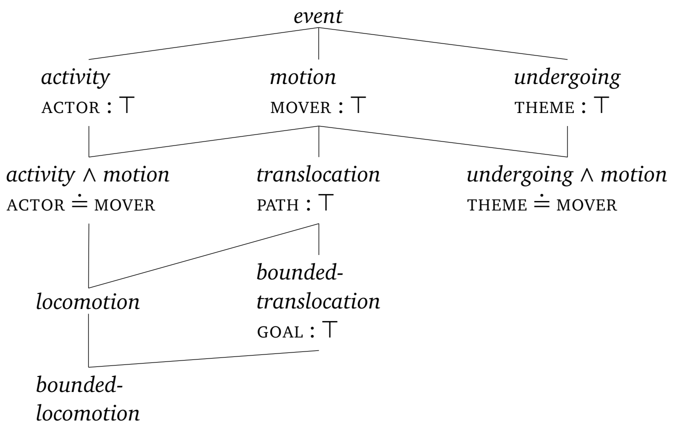


Figure 36:
Partial sketch of constraints on event types

During parsing, we have to build larger trees via substitution and adjunction. For this, algorithms of complexity $\mathcal{O}(n^6)$ with n the length of the input string are known (Vijay-Shanker and Joshi 1985; Joshi and Schabes 1997). On the semantic side, substitution and adjunction go along with the unification of frames (base-labelled feature structures) as defined in Section 3.3.1. As pointed out by Hegner (1994, pp. 136ff), the complexity of the unification of base-labelled feature structures is close to linear in the number of nodes. In fact, Hegner (1994, ibid.) shows that the complexity increases only slightly if the resulting feature structure is moreover required to satisfy a finite number of Horn descriptions. Recall that this is what we need in our approach since the unification of two frames may activate additional Horn constraints. But there is a caveat here. Hegner's result presumes a *finite* number of Horn descriptions while (universal) Horn constraints can give rise to an infinite number of them (cf. Section 3.3.4).

One way to keep this tractable is to make sure that the constraints under consideration do not introduce new nodes to the structure. Then the number of generated descriptions to be taken into account is finite. A new edge and a new node (and possibly further new edges and nodes) would for instance be added if we had a constraint in our system saying that if a frame is of type t_1 and of type t_2, then some additional attribute f has to be added, i.e., $t_1 \wedge t_2 \preceq f : \top$. So far, we do not have such constraints in our system. (The typical situation is illustrated by the partial sketch in Figure 36.) None of the conjunctive types introduces a new feature. Therefore, we make the assumption that constraints of this type are not allowed, and that, consequently, the frames obtained during parsing do not contain more nodes and edges than the union of the frames involved in the derivation. Constraints on relations, such as the transitivity of *part-of*, must of course be taken with care, too. But again, as long as no new nodes are added by the constraints involved, the complexity remains polynomial.

8 CONCLUSION

In this paper, we introduced an LTAG-based syntax-semantics interface with a fine-grained frame-based semantics. We have shown that this architecture provides the means to associate a detailed decomposition and composition of syntactic building blocks with a parallel decomposition and composition of meaning components. Due to its various possibilities for decomposing elementary trees and because of its extended domain of locality, LTAG allows one to pair not only lexical items with lexical meaning but also constructions with their meaning contributions. Furthermore, due to the metagrammatical specification of TAG elementary trees, the meaning contributions of single argument realizations and of their combinations can be described in a principled way, in parallel to a similar decomposition of the syntactic elementary trees.

We applied the framework to the case of directed motion expressions, and we have shown how to capture the various ways a directional PP adds information about the path of the motion event. Furthermore, we have demonstrated how to model syntax and semantics of the dative alternation, separating constructional aspects of meaning from lexical ones. Finally, we have presented a metagrammatical decomposition of our constructions that allows for an elegant meaning factorization which brings the two phenomena together by characterizing the parts that they have in common.

Besides giving a detailed frame-based analysis of lexical and constructional meaning aspects, our approach integrates this into a syntax-semantics interface. Via substitution and adjunction, the frame-based characterization of the events described by entire sentences can be compositionally derived.

The frames we use for semantics are typed feature structures that do not necessarily have a unique root, that allow to access any node in the feature structure via designated base nodes, and that allow for relations between nodes, besides the usual functional attributes. Such structures can be formalized as base-labelled feature structures. We have presented a feature logic that allows us to specify these frames and to express general constraints on them. We also described criteria for these constraints such that semantic composition (i.e., unification) under constraints is tractable.

ACKNOWLEDGEMENTS

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 Timm Lichte, Maribel Romero, and three anonymous reviewers for their valuable comments on earlier drafts of this article.

REFERENCES

Anne ABEILLÉ (2002), *Une Grammaire Électronique du Français*, CNRS Editions, Paris.

Anne ABEILLÉ and Owen RAMBOW (2000), Tree Adjoining Grammar: An Overview, in Anne ABEILLÉ and Owen RAMBOW, editors, *Tree Adjoining Grammars: Formalisms, Linguistic Analyses and Processing*, pp. 1–68, CSLI Publications, Stanford, CA.

Srinivas BANGALORE and Aravind K. JOSHI (2010), Introduction, in S. BANGALORE and A. K. JOSHI, editors, *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing*, pp. 1–31, MIT Press, Cambridge, MA.

John BEAVERS (2011), An Aspectual Analysis of Ditransitive Verbs of Caused Possession in English, *Journal of Semantics*, 28:1–54.

Benjamin K. BERGEN and Nancy CHANG (2005), Embodied Construction Grammar in simulation-based language understanding, in Jan-Ola ÖSTMAN and Mirjam FRIED, editors, *Construction Grammars. Cognitive Grounding and Theoretical Extensions*, pp. 147–190, John Benjamins, Amsterdam.

Patrick BLACKBURN (1993), Modal Logic and Attribute Value Structures, in Maarten DE RIJKE, editor, *Diamonds and Defaults*, pp. 19–65, Kluwer, Dordrecht.

Patrick BLACKBURN and Johan BOS (2003), Computational Semantics, *Theoria*, 18(1):27–45.

Joan BRESNAN and Marilyn FORD (2010), Predicting Syntax: Processing Dative Constructions in American and Australian Varieties of English., *Language*, 86(1):186–213.

Joan BRESNAN and Tatiana NIKITINA (2010), The Gradience of the Dative Alternation, in Linda UYECHI and Lian Hee WEE, editors, *Reality Exploration and Discovery: Pattern Interaction in Language and Life*, pp. 161–184, CSLI Publications, Stanford, CA.

Marie-Hélène CANDITO (1999), *Organisation modulaire et paramétrable de grammaires électroniques lexicalisées. Application au français et à l'italien*, Ph.D. thesis, Université Paris 7, Paris.

Bob CARPENTER (1992), *The Logic of Typed Feature Structures*, Cambridge University Press, Cambridge.

Ann COPESTAKE, Dan FLICKINGER, Carl POLLARD, and Ivan A. SAG (2005), Minimal Recursion Semantics: An Introduction, *Research on Language and Computation*, 3:281–332.

Benoit CRABBÉ (2005), Grammatical Development with XMG, in Philippe BLACHE, Edward STABLER, Joan BUSQUETS, and Richard MOOT, editors, *Proceedings of LACL 2005*, Lecture Notes in Artificial Intelligence 3492, pp. 84–100, Springer, Berlin.

Benoit CRABBÉ and Denys DUCHIER (2005), Metagrammar Redux, in Henning CHRISTIANSEN, Peter Rossen SKADHAUGE, and Jørgen VILLADSEN, editors, *Constraint Solving and Language Processing*, Lecture Notes in Computer Science 3438, pp. 32–47, Springer, Berlin.

Benoit CRABBÉ, Denys DUCHIER, Claire GARDENT, Joseph LE ROUX, and Yannick PARMENTIER (2013), XMG: eXtensible MetaGrammar, *Computational Linguistics*, 39(3):591–629.

David DOWTY (1979), *Word Meaning and Montague Grammar*, D. Reidel, Dordrecht.

David DOWTY (2003), The dual analysis of adjuncts/complements in Categorial Grammar, in Ewald LANG, Claudia MAIENBORN, and Cathrine FABRICIUS-HANSEN, editors, *Modifying Adjuncts*, Interface Explorations 4, pp. 33–66, Mouton de Gruyter, Berlin.

Veronika EHRICH (1996), Verbbedeutung und Verbgrammatik: Transportverben im Deutschen, in Ewald LANG and Gisela ZIFONUN, editors, *Deutsch - typologisch*, pp. 229–260, de Gruyter, Berlin.

Carola ESCHENBACH, Ladina TSCHANDER, Christopher HABEL, and Lars KULIK (2000), Lexical Specifications of Paths, in Christian FREKSA, Wilfried BRAUER, Christopher HABEL, and Karl Friedrich WENDER, editors, *Spatial Cognition II*, Lecture Notes in Computer Science 1849, pp. 127–144, Springer, Berlin.

Charles J. FILLMORE (1982), Frame Semantics, in *Linguistics in the Morning Calm*, pp. 111–137, Hanshin Publishing Co., Seoul.

Charles J. FILLMORE (1986), Pragmatically controlled zero anaphora, in *Berkeley Linguistics Society 12*, pp. 95–107.

Charles J. FILLMORE (2007), Valency Issues in FrameNet, in Thomas HERBST and Katrin GÖTZ-VOTTELER, editors, *Valency: Theoretical, Descriptive and Cognitive Issues*, pp. 129–160, Mouton de Gruyter, Berlin.

Charles J. FILLMORE, Christopher R. JOHNSON, and Miriam R. L. PETRUCK (2003), Background to FrameNet, *International Journal of Lexicography*, 16(3):235–250.

William A. FOLEY and Robert D. VAN VALIN (1984), *Functional Syntax and Universal Grammar*, Cambridge University Press, Cambridge.

Robert FRANK (2002), *Phrase Structure Composition and Syntactic Dependencies*, MIT Press, Cambridge, MA.

Claire GARDENT and Laura KALLMEYER (2003), Semantic Construction in FTAG, in *Proceedings of EACL 2003*, pp. 123–130.

Berit GEHRKE (2008), *Ps in Motion. On the semantics and syntax of P elements and motion events*, LOT, Utrecht.

Adele E. GOLDBERG (1995), *Constructions: A Construction Grammar Approach to Argument Structure*, University of Chicago Press, Chicago, IL.

Adele E. GOLDBERG and Ray JACKENDOFF (2004), The English resultative as a family of constructions, *Language*, 80:532–568.

Fritz HAMM, Hans KAMP, and Michiel VAN LAMBALGEN (2006), There is no opposition between Formal and Cognitive Semantics, *Theoretical Linguistics*, 32(1):1–40.

Stephen J. HEGNER (1994), Properties of Horn Clauses in Feature-Structure Logic, in C. J. RUPP, Michael A. ROSNER, and Rod L. JOHNSON, editors, *Constraints, Language and Computation*, pp. 111–147, Academic Press, San Diego, CA.

Ray JACKENDOFF (1991), Parts and Boundaries, *Cognition*, 41:9–45.

Aravind K. JOSHI and Yves SCHABES (1997), Tree-Adjoining Grammars, in Grzegorz ROZENBERG and Arto SALOMAA, editors, *Handbook of Formal Languages. Vol. 3: Beyond Words*, pp. 69–123, Springer, Berlin.

Laura KALLMEYER and Aravind K. JOSHI (2003), Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG, *Research on Language and Computation*, 1(1/2):3–58.

Laura KALLMEYER and Maribel ROMERO (2008), Scope and Situation Binding in LTAG using Semantic Unification, *Research on Language and Computation*, 6(1):3–52.

Ingrid KAUFMANN (1995), *Konzeptuelle Grundlagen semantischer Dekompositionsstrukturen. Die Kombinatorik lokaler Verben und prädikativer Argumente*, Niemeyer, Tübingen.

Jean-Pierre KOENIG and Anthony R. DAVIS (2001), Sublexical Modality and the Structure of Lexical Semantic Representations, *Linguistics and Philosophy*, 24:71–124.

Karsten KONRAD (2004), *Model Generation for Natural Language Interpretation and Analysis*, Springer, Berlin.

Manfred KRIFKA (2004), Semantic and pragmatic conditions for the Dative Alternation, *Korean Journal of English Language and Linguistics*, 4:1–32.

Anthony S. KROCH (1989), Asymmetries in long-distance extraction in a Tree Adjoining Grammar, in Mark R. BALTIN and Anthony S. KROCH, editors, *Alternative Conceptions of Phrase Structure*, pp. 66–98, University of Chicago Press, Chicago, IL.

Beth LEVIN and Malka RAPPAPORT HOVAV (2005), *Argument Realization*, Cambridge University Press, Cambridge.

Timm LICHTE, Alexander DIEZ, and Simon PETITJEAN (2013), Coupling Trees and Frames through XMG, in Denys DUCHIER and Yannick PARMENTIER, editors, *ESSLLI 2013 Workshop on High-level Methodologies for Grammar Engineering*, pp. 37–48.

Sebastian LÖBNER (2014), Evidence for Frames from Human Language, in Thomas GAMERSCHLAG, Doris GERLAND, Rainer OSSWALD, and Wiebke PETERSEN, editors, *Frames and Concept Types*, pp. 23–67, Springer, Dordrecht.

Claudia MAIENBORN (2011), Event Semantics, in Claudia MAIENBORN, Klaus VON HEUSINGER, and Paul PORTNER, editors, *Semantics. An International Handbook of Natural Language Meaning. Volume 1*, pp. 802–829, Mouton de Gruyter, Berlin.

Inderjeet MANI and James PUSTEJOVSKY (2012), *Interpreting Motion. Grounded Representations for Spatial Language*, Oxford University Press, Oxford.

Stefan MÜLLER (2006), Phrasal or Lexical Constructions, *Language*, 82(4):850–883.

Rebecca NESSON and Stuart M. SHIEBER (2006), Simpler TAG Semantics Through Synchronization, in Shuly WINTNER, editor, *Proceedings of the 11th Conference on Formal Grammar*, pp. 129–142.

Rainer OSSWALD (1999), Semantics for Attribute-Value Theories, in Paul DEKKER, editor, *Proceedings of the Twelfth Amsterdam Colloquium*, pp. 199–204, University of Amsterdam, ILLC, Amsterdam.

Rainer OSSWALD and Robert D. VAN VALIN (2014), FrameNet, Frame Structure, and the Syntax-Semantics Interface, in Thomas GAMERSCHLAG, Doris GERLAND, Rainer OSSWALD, and Wiebke PETERSEN, editors, *Frames and Concept Types*, pp. 125–156, Springer, Dordrecht.

Terence PARSONS (1990), *Events in the Semantics of English*, MIT Press, Cambridge, MA.

Malka RAPPAPORT HOVAV and Beth LEVIN (1998), Building Verb Meanings, in Miriam BUTT and Wilhelm GEUDER, editors, *The Projection of Arguments: Lexical and Compositional Factors*, pp. 97–134, CSLI Publications, Stanford, CA.

Malka RAPPAPORT HOVAV and Beth LEVIN (2008), The English dative alternation: A case for verb sensitivity, *Journal of Linguistics*, 44:129–167.

Mike REAPE (1994), A Feature Value Logic with Intensionality, Nonwellfoundedness and Functional and Relational Dependencies, in C. J. RUPP, Michael A. ROSNER, and Rod L. JOHNSON, editors, *Constraints, Language and Computation*, pp. 77–110, Academic Press, San Diego, CA.

William C. ROUNDS (1997), Feature Logics, in Johan VAN BENTHEM and Alice TER MEULEN, editors, *Handbook of Logic and Language*, pp. 475–533, North-Holland, Amsterdam.

Ivan SAG (2012), Sign-Based Construction Grammar: An informal synopsis, in Hans BOAS and Ivan SAG, editors, *Sign-Based Construction Grammar*, pp. 61–188, CSLI Publications, Stanford.

Klaas SIKKEL (1997), *Parsing Schemata*, Springer, Berlin.

Leonard TALMY (2000a), *Toward a Cognitive Semantics. Volume I: Concept Structuring Systems*, MIT Press, Cambridge, MA.

Leonard TALMY (2000b), *Toward a Cognitive Semantics. Volume II: Typology and Process in Concept Structuring*, MIT Press, Cambridge, MA.

Robert D. VAN VALIN and Randy J. LAPOLLA (1997), *Syntax*, Cambridge University Press, Cambridge.

Robert D. VAN VALIN, Jr. (2005), *Exploring the Syntax-Semantics Interface*, Cambridge University Press, Cambridge.

Henk VERKUYL and Joost ZWARTS (1992), Time and Space in Conceptual and Logical Semantics: The Notion of Path, *Linguistics*, 30:483–511.

K. VIJAY-SHANKER and Aravind K. JOSHI (1985), Some computational properties of Tree Adjoining Grammars, in *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pp. 82–93.

K. VIJAY-SHANKER and Aravind K. JOSHI (1988), Feature Structures Based Tree Adjoining Grammar, in *Proceedings of the 12th International Conference on Computational Linguistics (COLING-88)*, pp. 714–719.

Fei XIA (2001), *Automatic grammar generation from two different perspectives*, Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Fei XIA, Martha PALMER, and VIJAY-SHANKER (2010), Developing Tree-Adjoining Grammars with Lexical Descriptions, in S. BANGALORE and A. K. JOSHI, editors, *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing*, pp. 73–110, MIT Press, Cambridge, MA.

XTAG RESEARCH GROUP (2001), A Lexicalized Tree Adjoining Grammar for English, Technical report, Institute for Research in Cognitive Science, University of Pennsylvania, Philadelphia, PA.

Jordan ZLATEV, Johan BLOMBERG, and Caroline DAVID (2010), Translocation, language and the categorization of experience, in Vyvyan EVANS and Paul CHILTON, editors, *Language, Cognition and Space*, pp. 389–418, Equinox, London.

Joost ZWARTS (2005), Prepositional Aspect and the Algebra of Paths, *Linguistics and Philosophy*, 28(6):739–779.