A Fibring Semantics for the Semantic-Morphological Interface in Natural Language

RALF NAUMANN and ANJA LATROUITE, *Seminar für Allgemeine Sprachwissenschaft, University of Düsseldorf, Germany.*
E-mail: naumann@phil-fak.uni-duesseldorf.de

Abstract

In Lexical Decompositional Grammar (LDG), Wunderlich (1997), a verb v in the lexicon is analyzed with respect to different levels like the *Semantic Form* (SF), the *Theta-Structure* (TS) and the *syntactic-morphological* (S/M) level. What distinguishes LDG from other grammatical theories like LFG or HPSG is the way the different parts are related to each other. E.g., at the level of linking the assignment of case to an argument NP at the morphological level is constraint by the position of the corresponding argument at the level of SF.

In this paper a formalization of LDG is presented. Structures for the analysis of verbs in LDG are tuples of the form $< \mathbf{S}_1, ..., \mathbf{S}_n, \gamma_1, ..., \gamma_m >$ such that each $\mathbf{S}_i$ is the substructure corresponding to the i-th level of linguistic analysis and each γ_j is a mapping relating the domains of two substructures. If $\mathbf{S}$ and $\mathbf{S}'$ are related by some γ, the information that is associated with the objects that are related by γ is constraint. E.g., if some $d \in D_{\mathbf{S}}$ has property P, $\gamma(d) \in D_{\mathbf{S}'}$ is required to have property P'. The combination of different ontologies is reflected at the syntactic level by *fibring* or *layering* the languages that are used to talk about the structures over each other.

1 Data and Evidence

An argument can be linked to a verb v in several ways. The German examples in (1) illustrate two principle mechanisms: that by case and that by a preposition.

(1) a. Peter aß einen Apfel. (Peter ate an apple)
b. Hans glaubt an Gott. (Hans believes in God)

Both arguments of 'essen' (eat) in (1a) as well as the first gument of 'glauben' ('believe') in (1b) are linked by case whereas the second one in (1b) is linked by a preposition. Two questions that any theory of grammar has to answer are: (α) What are the regularities of how an argument is linked to a verb, e.g., is there a relationship between argument positions and cases? and (β) To what degree is argument linking semantically driven, i.e., does it depend on some properties of the verb? Relative to the data given in (1) the following generalizations at the level of word order can be made: (a) The first argument is realized by nominative case and (b) The last argument that is realized by case gets accusative case. Pustejowsky (1991) answers question (α) in the affirmative. His approach is based on an underlying event ontology in the sense that an n-place verb v is interpreted as an $n+1$-ary relation such that $< d_1, ..., d_n, d_{n+1} > \in [\![v]\!]$ only if d_{n+1} is an event. The relationship between an event e and its participants is defined in terms of thematic roles like *Agent* or *Undergoer*. These roles are likewise used to define the mapping between the semantic component

and the argument structure of a verb. Two examples are: (i) The argument that is semantically linked to the Agent-role is realized as the highest argument of the argument structure (which often corresponds to the grammatical relation of subject) and (ii) The argument that is semantically linked to the Undergoer-role is realized as the lowest argument that is realized by case (which often corresponds to the grammatical relation of direct object). According to these two linking rules, particular thematic roles correspond to particular positions in the argument structure of a verb. If these rules are combined with the generalizations in (a) and (b), they can be extended to rules which relate particular thematic roles to particular cases: (iii) The argument that is semantically linked to the Agent-role is realized by nominative case and (iv) The argument that is semantically linked to the Undergoer-role is realized by accusative case.
The main problem that an approach like that of Pustjowsky faces is that there is strong evidence against the possibility of generalizing the rules in (i)-(iv) to all thematic roles. E.g., in 'John reached the station' and in 'Bill put the book on the table' the station and the table are both assigned the Goal-role. But the corresponding arguments are realized differently.

2 Lexical Decomposition Grammar

In Lexical Decomposition Grammar, Wunderlich (1997), three different grammatical levels of representation are distinguished: Semantic Form (SF), Theta Structure (TS) and Morphology/Syntax (MS). Although each of the three levels is characterized by its own set of structural properties, they are linked to each other by a set of mappings which constrain the relationships between them. Below this three-level architecture is illustrated by the ditransitive verb 'give'.

(2) The three grammatical levels of LDG

TS				SF
λz	λy	λx	λe	{(ACT(x) & BEC(POSS(y,z)))(e)}
+hr	+hr	-hr		
-lr	+lr	+lr		

$\Updownarrow$ $\Updownarrow$ $\Updownarrow$

ACC	DAT	NOM
MS		

When taken together, TS and SF correspond to the translation of 'give' in the representation language L which is assumed to be a variant of a many-sorted type-theory, i.e., they are a well-formed expression α of L. SF, the matrix of α, which is of type t, is based on a decompositional analysis of the verb. It is of the form $(\alpha_1 \ \& \ ... \ \& \ \alpha_n)(e)$. Each conjunct α_i is of type $< \epsilon, t >$ for ϵ the type of events (or eventualities) such that it is interpreted as a unary relation on the domain E of events. The connective & expresses a kind of temporal succession in the following sense: $\alpha_1 \ \& \ ... \ \& \ \alpha_n$ expresses a complex property of events such that this property is true of an event e just in case there are initial stages (prefixes) e_1, ..., e_n of e with α_i being true of e_i, $1 \leq i \leq n$, and e_i being an initial stage of e_j for $1 \leq i < j \leq n$. Thus, if the SF is evaluated with respect to a given event e (and suitable values for the remaining variables), each conjunct α_i is evaluated with respect to subevents e' of e. As & is not commutative, it is possible to define a linear ordering on the (non-eventive) arguments of the α_i.

The theta-structure TS is defined in terms of an initial prefix of the sequence of variables that are bound by the λ-operators such that each variable represents an argument position. For $n > 1$ it corresponds to the sequence of the first $n - 1$ λ-abstractions and to the only λ-abstraction otherwise. Each position of the sequence can be characterized in terms of properties that only depend on its place in the sequence (*structural* properties) or in terms of further information that is lexically determined (*non-structural* properties). The difference between the two sorts of properties is related to the distinction between structural and non-structural (semantic) linking. Arguments that are structurally linked are always linked by case and the assignment of case to an argument NP depends only on structural properties of the corresponding argument position, Naumann/Latrouite (2000). The structural properties are defined in terms of two binary features $[\pm lr]$ and $[\pm hr]$ with intuitive meaning 'there is a/no lower role' and 'there is a/no higher role', respectively.

The syntactic component S of MS is an ordered finite binary tree the nodes of which are decorated with the categorial and subcategorization information determined by the verb such that it contains information about the (syntactic) categories and the relative order of the arguments. The morphological component M of MS consists of a feature structure for each node of the tree that contains information appropriate for the corresponding categorial information like case, gender and number insofar as this information is already determined by the verb.

The different levels are related by various mappings that constrain the distribution of information: (i) SF - TS: relationship between the order of arguments in SF and the order of λ-abstractions. This relationship is constrained by the *Hierarchy Principle*: The order of arguments in TS is the inverse of the order of arguments in SF. Thus, if $< x_1, ..., x_n >$ is the linearly ordered sequence of arguments in SF, $< x_n, ..., x_1 >$ is the ordering in TS; (ii) TS - syntactic component of MS: The default word order is the inverse of the order of arguments in TS; (iii) syntactic component of MS - morphological component of MS: the assignment of case to argument NPs depends on the position of the corresponding argument in TS. The task consists in formalizing the ideas that have informally been presented in this section. In particular, the following questions have to be answered: (a) what is an appropriate ontology (together with a structure) for each level?; (b) what is an appropriate language (logic) to talk about this ontology?; (c) how can the constraints between the different levels be defined at the level of structures and how can they be expressed at the corresponding level of logics? In the next section suitable mathematical structures for each level will be defined, i.e. an ontology, that is a domain of entities of a particular sort, will be determined together with a set of relations.

3 The Formalization

The outline of the basic ideas on which LDG is built in the previous section has shown that the linguistic analysis of a verb in the lexicon is cast in terms of a composite ontology. There are events and 'ordinary' objects (SF), (tree) nodes (TS, S) and points (in feature structures) (M). Each sort of entity comes equipped with a set of relations, giving rise to a structure in the model-theoretic sense, with the set of entities as the domain that underlies the structure. Suitable structures for the analysis of verbs will therefore have a complex ontology consisting of a *dynamic eventuality* (sub-)structure, *ordered finite binary trees*, *feature structures* and, most importantly from a linguistic point of view, *mappings between the different substructures* that mutually constrain the relations given in the structures.

3.1 The Structures

An LDG structure for a verb v in the lexicon is a tuple
$< \mathbf{E}, \mathbf{T}_{TS}, \mathbf{T}_S, \{\mathbf{FS}_n \mid n \in N_S\}, R_{z_f}, R_{z_{TS}}, R_{z_{sf}}, \gamma >$ such that

- $\mathbf{E} = < E, S, P_v, \leq_E, C, \alpha, \omega >$ is a dynamic eventuality structure such that (i) E is a (non-empty) set of events; (ii) S is a (non-empty) set of states (or time points); (iii) P_v is the set of all events of type v; (iv) $\leq_E$ is the material part of relation on E, that is a partial order; (v) C is the composition relation that is required to be associative and that is defined by $C = \{(e, e_1, e_2) \mid e_1 \leq_E e \wedge e_2 \leq_E e \wedge \omega(e_1) = \alpha(e_2) \wedge \alpha(e) = \alpha(e_1) \wedge \omega(e) = \omega(e_2)\}$; (vi) $R_B = \{(e, e') \mid \exists e'' : Ce', ee''\}$; (vii) $\alpha : E \rightarrow S$ and $\omega : E \rightarrow S$ assign to each event $e \in E$ its beginning point and end point, respectively
- $\mathbf{T}_{TS} =< N_{TS}, R_{\succ_{1_{TS}}}, R_{\succ_{2_{TS}}}, root_{TS}, \Theta_{TS} >$ and $\mathbf{T}_S = < N_S, R_{\succ_{1_S}}, R_{\succ_{2_S}}, root_S, \Theta_S >$ are finite ordered binary trees, respectively
- each $\mathbf{FS}_n = < W_n, \{R_f\}_{f \in F}, \{Q_a\}_{a \in A}, w_{0_n} >$ is a point-generated feature structure; For $n \neq n'$, $\mathbf{FS}_n$ and $\mathbf{FS}_{n'}$ are disjoint in the sense that $W_n \cap W_{n'} = \emptyset$
- R_{z_f}, $R_{z_{ts}}$ and $R_{z_{sf}}$ are mappings between $\mathbf{T}_S$ and $\{\mathbf{FS}_n \mid \mathbf{n} \in \mathbf{N}_S\}$, $\mathbf{T}_S$ and $\mathbf{T}_{TS}$, as well as $\mathbf{E}$ and $\mathbf{T}_{TS}$
- γ is a function that maps each $\mathbf{FS}_n$ to its root w_{0_n}

In the sequel of this section the relationship between the syntactic and the morphological component of MS will be analyzed. We will begin by defining the structures. In a next step, suitable languages, L^{T_S} and L^F for talking about these structures will be defined, followed by a definition of the relation R_{z_f}, which links the two structures. Finally, the two languages are combined so that it is possible to talk about the combined ontology.

Finite Ordered Binary Trees The definition of a finite ordered tree given below applies both to $\mathbf{T}_S$ and $\mathbf{T}_{TS}$. They differ with respect to the (basic part of) languages that are used to talk about them. The definition is taken from Blackburn/Meyer-Viol (1997). A finite ordered binary tree $\mathbf{T}$ is a tuple $< N, R_{\succ_1}, R_{\succ_2}, root, \Theta >$ such that (i) N is a finite, non-empty set of (tree) nodes; (ii) $R_{\succ_i}$ for $1 \leq i \leq 2$ is a binary relation on N that is a partial function; $R_{\succ_1}(n, n')$ just in case n' is the first daughter of n and $R_{\succ_2}(n, n')$ just in case n' is the second daughter of n; (iii) $root$ is the (unique) root of the tree; (iv) $\Theta \subseteq N$ is the set of terminal nodes of the tree, i.e., $\forall n[n \in N \wedge n \in \Theta \rightarrow \forall n'[n' \in N \rightarrow \neg R_{\succ_i}(n, n')]]$, i.e., terminal nodes do not immediately dominate any other node of the tree
In terms of $R_{\succ_1}$ and $R_{\succ_2}$ the following binary relations are defined:

(3) a. $R_{\succ} = R_{\succ_1} \cup R_{\succ_2}$ (the 'daughter of' relation)

b. $R_{\prec} = (R_{\succ})^c = \{(n, n') \mid (n', n) \in R_{\succ}\}$ (the 'mother of' relation, which is a partial function that is defined on $N - \{root\}$)

c. $R_{\succ^*}$ and $R_{\prec^*}$ are the transitive, reflexive closure of $R_{\succ}$ and $R_{\prec}$, respectively. $R_{\succ^*}$ is the 'dominates' relation whereas $R_{\prec^*}$ is the 'is dominated by' relation; note that due to the reflexive character one has $\forall n[R_{\succ^*}(n,n) \wedge R_{\prec^*}(n,n)]$.

d. $R_{>} = \{(n, n') \mid \exists n_1 \exists n_2 \exists n_3 [R_{\succ^*}(n_1, n) \wedge R_{\succ_1}(n_2, n_1) \wedge R_{\succ_2}(n_2, n_3) \wedge R_{\succ^*}(n_3, n')]\}$ (the precedence relation on N)

e. $R_{<}$, the succession relation on N, is defined as the inverse of $R_{>}$: $R_{<} = (R_{>})^c$

Finite rooted Feature Structures Feature Structures **FS** are defined as labeled decorated directed graphs that are multi-modal Kripke models in which every relation is a partial function (Blackburn/Meyer-Viol (1997), Blackburn et al. (1993)). A finite rooted feature structure **FS** of signature $< F, A >$, F the set of *features* and A the set of *attributes* (or *decorations*) is a triple $< W, \{R_f\}_{f \in F}, \{Q_a\}_{a \in A}, w_0 >$ such that (i) W is a non-empty set, the set of (feature structure) points; (ii) each R_f, $f \in F$, is a binary relation on W that is required to be a partial function; (iii) each Q_a, $a \in A$, is a unary relation on W; (iv) w_0 is the (unique) root of **FS**, i.e., **FS** is *generated* by w_0 in the sense that each point $w \in W$ can be reached from w_0 in a finite number of steps.

The Languages $L^{T_S}(VAR)$ and L^F The well-formed formulas of the tree language $L^{T_S}(VAR)$ are defined as follows: (i) $p \in L^{T_S}(VAR)_{wff}$, for $p \in VAR$, (ii) if ϕ, $\psi \in L^{T_S}(VAR)_{wff}$, then $\phi \wedge \psi \in L^{T_S}(VAR)_{wff}$ and $\neg\phi \in L^{T_S}(VAR)_{wff}$, (iii) if $\phi \in L^{T_S}(VAR)$, then $\alpha\phi \in L^{T_S}(VAR)$ for $\alpha \in \{\downarrow^1, \downarrow^2, \uparrow, \downarrow^*, \uparrow^*, <, >\}$ and nothing else is an element of $L^{T_S}(VAR)_{wff}$. In terms of the modalities used in clause (iii) the following modalities and constants are defined.

(4) a. $\downarrow \phi =_{def.} \downarrow^1 \phi \vee \downarrow^2 \phi$

b. $\Downarrow^i \phi =_{def.} \neg \downarrow \neg\phi$ for $1 \leq i \leq 2$; $\Downarrow \phi =_{def.} \neg \downarrow \neg\phi$ and $\Uparrow \phi =_{def.} \neg \uparrow \neg\phi$

c. $\Uparrow^* \phi =_{def.} \neg \uparrow^* \neg\phi$ and $\Downarrow^* \phi =_{def.} \neg \downarrow^* \neg\phi$

d. $s_0 =_{def.} \Uparrow \bot$ and $t =_{def.} \Downarrow \bot$

The wffs of $L^{T_S}(VAR)$ are interpreted on finite ordered binary trees. The satisfaction relation is defined as follows (for the base clause see below (7) and (11)).

(5) a. $M, n \models \neg\phi$ iff $M, n \not\models \phi$

b. $M, n \models \phi \wedge \psi$ iff $M, n \models \phi$ and $M, n \models \psi$

c. $M, n \models \downarrow^1 \phi$ iff $\exists n'(R_{\succ_1}(n, n') \wedge M, n' \models \phi)$

d. $M, n \models \downarrow^2 \phi$ iff $\exists n'(R_{\succ_2}(n, n') \wedge M, n' \models \phi)$

e. $M, n \models \uparrow \phi$ iff $\exists n'(R_{\prec}(n, n') \wedge M, n' \models \phi)$

f. $M, n \models \downarrow^* \phi$ iff $\exists n'(R_{\succ^*}(n, n') \wedge M, n' \models \phi)$

g. $M, n \models \uparrow^* \phi$ iff $\exists n'(R_{\prec^*}(n, n') \wedge M, n' \models \phi)$

h. $M, n \models < \phi$ iff $\exists n'(R_{<}(n, n') \wedge M, n' \models \phi)$

i. $M, n \models > \phi$ iff $\exists n'(R_{>}(n, n') \wedge M, n' \models \phi)$

The clauses for the defined constants s and t are $M, n \models s_0$ iff $n = root_S$ and $M, n \models t$ iff $n \in \Theta_S$ from which it follows that they are true only at the root node and at terminal nodes, respectively.

The wffs of L^F are defined as follows. The set L^F_{wfff} is the smallest set such that (i) $a \in L^F_{wff}$ for each $a \in A$ (i.e., the set of attributes functions as the set of propositional variables for L^F, (ii) if ϕ, $\psi \in L^F_{wff}$, then $\neg\phi, \phi \wedge \psi \in L^F_{wff}$ and (iii) if $\phi \in L^F_{wff}$, then $< f > \phi \in L^F_{wff}$ for each $f \in F$, i.e., the elements of F function as modal operators. The choice of both A and F depends on the application as well as on the linguistic theory that is used. For the present purpose, A will be assumed to contain at least s, np, vp and vp' which are used to express purely categorial information about nodes, nom, dat and acc, which are the values of the feature $CASE$ as well as pl and sg, that are admissible values for the NUM feature. Thus, A can be taken as the union of the label sets $Cat = \{s, np, vp, vp'\}$, $Case = \{nom, dat, acc\}$ and $Num = \{sg, pl\}$. The set F contains at least the features $CASE$ and $NUM(BER)$. An example of a wff of L^F is $np \wedge < CASE > acc \wedge < NUM > sg$.
Elements of L^F_{wff} are interpreted on feature structures **FS**. The satisfaction relation is given in (6).

(6) a. $M, w \models a$ iff $w \in Q_a$ for $a \in A$
b. $M, w \models \neg\phi$ iff $M, w \not\models \phi$
c. $M, w \models \phi \wedge \psi$ iff $M, w \models \phi$ and $M, w \models \psi$
d. $M, w \models < f > \phi$ iff $\exists w'(R_f(w, w') \wedge M, w' \models \phi)$

The injective functional relation R_{z_f} with domain N_S assigns to each node $n \in N_S$ a feature structure $R_{z_f}(n) = \mathbf{FS}_n$. The image of R_{z_f} is the set $\{\mathbf{FS}_n \mid n \in N_S\}$ which is an element of the LDG structure defined above. The composition of R_{z_f} with the function γ assigns to each node $n \in N_S$ the root w_{0_n} of the feature structure $\mathbf{FS}_n$ assigned to n by R_{z_f}: $[\gamma \circ R_{z_f}](n) = w_{0_n}$.
In order to be able to talk about LDG structures for verbs, i.e. about the combined ontology, it is necessary to combine the languages with each other. This means that it must be possible to express in at least one language L constraints that are evaluated with respect to elements of structures with respect to which formulas from L are not evaluated. The languages will be combined at the level of atomic wffs of L^{T_S}. Instead of building L^{T_S} on top of a set VAR of propositional variables, it is built in terms of a set of structured atomic formulas, namely the wffs of $L^{T_{TS}}$ and L^F. In contrast to Blackburn et al. (1993), the wffs ϕ of $L^{T_{TS}}$ and L^F are not directly taken as wffs of L^{T_S}. Rather, the formulas will be of the form $< z_{ts} > \phi$ and $< z_f > \psi$ for ϕ a wff of $L^{T_{TS}}$ and ψ a wff of L^F. Thus, VAR is taken to be the union of the sets $\{< z_{ts} > \psi \mid \psi \in L^{T_{TS}}_{wff}\}$ and $\{< z_f > \phi \mid \phi \in L^F_{wff}\}$ such that syntactically the propositional variables $p \in VAR$ are replaced by formulas of the form $< \alpha > \phi$ as defined above. The modalities z_{ts} and z_f are interpreted in the LDG structure by one of the binary relations that link the tree structure $\mathbf{T}_S$ to the tree structure $\mathbf{T}_{TS}$ and one of the feature structures $\mathbf{FS}_n$, respectively. The base clause in the definition of satisfaction for z_f is given in (7).

(7) $M, n \models < z_f > \phi$ iff $\exists \mathbf{FS}[R_{z_f}(n, \mathbf{FS})\ (= \mathbf{FS}_n) \wedge M, \gamma(\mathbf{FS}) \models \phi]$

Following Blackburn et al. (1993) and Blackburn/Meyer-Viol (1997), the resulting language, $L^{T_S}(L^{T_{TS}}, L^F)$, will be called the language L^{T_S} *layered over* the languages $L^{T_{TS}}$ and L^F. Thus, a language with three 'layers' is used. The top layer are the formulas from L^{T_S} as defined above. They move us around the syntactic tree and are used to talk about structural (or configurational) properties of the tree. The base

layer are formulas of the form $< \alpha > \phi$ with $\alpha = z_{ts}$ or $\alpha = z_f$. Formulas $< z_f > \phi$ with $\phi \in L^F_{wff}$ express both syntactic and morphological information about a node n: what is the syntactic category of n?; what are the values of features like $CASE$ and $NUM(BER)$? They are used to talk about the 'internal' structure of nodes.
In order to be linguistically admissible, a feature decorated tree must satisfy a number of constraints of how atomic information is distributed. One prominent example of a constraint are phrase structure rules which constrain the distribution of purely categorial information. A typical example is $S \rightarrow NP\ VP$ which says that a local tree with root node of sort s is admissible if it immediately dominates a left node of sort vp and if it immediately dominates a right node of sort np. Formally, they impose restrictions on $R_{\succ_1}$ and $R_{\succ_2}$, i.e. the 'immediately left (right) dominates' relation. In $L^{T_S}(L^{T_{TS}}, L^F)$ this rule is expressed by the $L^{T_S}(L^F)$ formula $< z_F > (s \rightarrow \downarrow^1 np \wedge \downarrow^2 vp)$ $(= < z_F > s \rightarrow < z_F > \downarrow^1 np \wedge < z_F > \downarrow^2 vp$. This constraint concerns the distribution of purely categorial information and can be given the form $< z_f > (\phi \rightarrow \psi)$ for $\phi, \psi \in L^F_{wff}$. The structural information, expressed by pure L^{T_S} formulas, plays no role. The constraints on the relation between the syntactic and the morphological component imposed by LDG are of the form $< z_f > p \wedge\ t \wedge \alpha(< z_f > p \wedge t)$ $(p \in Cat,\ \alpha \in \{<, >\})$, i.e., the antecedent ϕ is not a pure L^F formula but expresses both purely categorial information and structural information about the position of the node in the tree in relation to its categorial classification. Antecedents of this form require that terminal nodes n $(n \in \Theta_S)$ of sort p satisfy the information expressed by the succedent ψ $(= < z_f > \phi)$, depending on their position among those elements of Θ_S that are also of sort p. An antecedent of the form $< z_f > p \wedge\ t \wedge \alpha$ for $\alpha = \ < (< z_f > p \wedge t)$ or $\alpha = \ > (< z_f > p \wedge t)$ expresses information about the order of arguments of the verb that are of sort p. As the constraint on the categorial sort of the terminal node expresses purely syntactic information, it follows that particular syntactic information at one place of the tree $\mathbf{T}_S$ requires some information to be present at that node too. In LDG this information concerns the value of the $CASE$ feature and therefore morphological information. For a three place verb like 'give' the constraints are given in (8).

(8) a. $< z_f > np\ \wedge t \wedge \neg < (< z_f > np_t \wedge t) \rightarrow < z_f >< CASE > nom$

b. $< z_f > np \wedge t \wedge\ < (< z_f > np \wedge t) \wedge\ > (< z_f > np \wedge t) \rightarrow < z_f >< CASE > dat$

c. $< z_f > np \wedge t\ \wedge \neg > (< z_f > np \wedge t) \wedge\ < (< z_f > np \wedge t) \rightarrow < z_f >< CASE > acc$

The constraints in (8) only hold if the argument positions are structural, otherwise there are exceptions to the constraints in (8b) and (8c). An example for (8b) are verbs with double accusative like 'lehren' ('teach'), 'Ich lehrte einen Studenten Semantik' ('I taught a student semantics'), where the intermediate argument is assigned accusative and not dative case, as predicted by (8b). The constraints in (8) must then be made conditional on the fact that additional information that is introduced by the verb does not hold. If this information is present, other constraints are effective and linking is no longer structural (see Naumann/Latrouite (2000) for details).

4 The SF-and the TS-Level

As was said in section (2), in LDG the SF of a verb v is cast in terms of an underlying event ontology. Yet, no formal definitions of the atomic SF-predicates and the connective & are given. The formalization of the SF-level that will be presented is based on *Dynamic Event Semantics* (DES). DES is based on the intuition that non-stative verbs express changes, Naumann (1999). A change is either an object (event) or a transformation of states (TS). Changes as objects bring about results by transforming a state s at which a result does not hold into a state s' where the result holds. This double perspective on the intuitive notion of a change is modeled in DES by having both an eventuality-structure $\mathbf{E}$ with an underlying domain E of events and a transition-structure $\mathbf{S}$ with an underlying domain S of states that is linearly ordered by $<_S$. Each event $e \in E$ is assigned its execution sequence, that is a convex subset of S. The domain E is sorted by the label set $VERB$: for $v \in VERB$, $P_v \subseteq E$ is the set of all events of type v, e.g. the set of all eating events if $v = eat$.
An event e can bring about more than one result (or TS) such that the relationship between an event and a TS is not one-one but one-many. P_v determines for each of its elements a set $Res(P_v, e)$ of results that e can possibly bring about. These results can be distinguished by the way they are evaluated on the execution sequence of e: (i) a result Q is *minimal* if it holds at all states of the execution sequence of e that are end points of initial stages e' of e that are of type P_v and (ii) a result Q is *maximal* if it only holds at the end point of e (if it holds at all on e's execution sequence). Results that are both minimal and maximal are called *w-maximal*, whereas results that are minimal without being maximal are called *w-minimal* and those that are maximal but not minimal *s-maximal.*
For an event of type 'John eat a fish' examples are: w-minimal: part of the fish is in John's stomach; s-maximal: (the mass of) the fish is in John's stomach; the fish does not exist. These results are (temporally) ordered: first the w-minimal results are brought about and then the s-maximal ones. It is therefore possible to define an ordering $\leq_v$ on $Res(P_v, e)$: $Q \leq_v Q'$ if Q is brought about not later than Q' on the execution sequence of e. Two results Q and Q' that are indistinguishable according to $\leq_v$ can be distinguished by implication relations. This gives rise to the more fine-grained ordering $\leq_v^*$: $Q_1 \leq_v^* Q_2$ iff $Q_1 <_v Q_2 \vee [Q_1 =_v Q_2 \wedge Q_1 \subseteq Q_2]$. Thus, for results that are temporally equivalent, implied results are 'greater' than those which imply them. Implied results are indicated by the subscript 2 whereas implying ones by the subscript 1. For the examples above using $\leq_{eat}^*$ yields the ordering w-minimal $\leq_{eat}^*$ s-maximal$_1$ $\leq_{eat}^*$ s-maximal$_2$.
The set $Res(P_v, e)$ can be partitoned by the equivalence relation 'is $\approx_v^*$-equivalent to': $[Q_T]_{v,e} = \{Q \mid Q \in Res(P_v, e) \wedge T(P_v, Q, e)\}$ for T a type of result. The ordering $\leq_v^*$ induces a corresponding ordering on these equivalence classes. At the level of v, the equivalence classes are defined as follows: $[Q_T]_v = \{Q \mid \exists e[e \in P_v \wedge Q \in [Q_T]_{v,e}]\}$.
Verb-classes are defined in terms of types of results. E.g., in English Accomplishment-verbs like 'eat' determine all three types of results whereas activity-verbs like 'push' only determine a w-minimal result. Transfer-verbs like 'give' determine only w-maximal results. The results assigned to an event $e \in P_v$ are always brought about with respect to objects that participate in e. E.g., if e is of type 'John eat a fish', the results are brought about with respect to John and the fish. John is assigned both the w-minimal result (part of the fish is in his stomach) and the s-maximal$_1$ one ((the

whole mass of) the fish is in his stomach). These results are also assigned to the fish which is, in addition, assigned the s-maximal$_2$ result (the fish no longer exists). For an event e of type 'John give the book to Mary' there are two w-maximal results: 'Mary does have the book' (w-max$_1$) and 'John does not have the book' (w-max$_2$) which yields the assignment: w-max$_1$: Mary, the book; w-max$_2$: John, the book. The relationship between an event $e \in P_v$, an object d and a result Q that e can bring about with respect to d is captured by a function Δ_v: $\Delta_v(e)(d)(Q)$ holds just in case Q is a result that e can possibly bring about with respect to d.

The Level of TS To each result Q from $Res(P_v, e)$ there corresponds a subset of e's participants. These sets can be defined using Δ_v and the equivalence classes defined by $O_{i_{v,e}} = \{d \mid \exists Q[\Delta_v(e)(d)(Q) \wedge Q \in [Q_i]_{v,e}]\}$ ($[Q_i]_{v,e}$) is the i-th equivalence class). The $O_{i_{v,e}}$ need not be singletons, as the examples above have shown. Yet, as shown in Naumann (2000), each participant of e can be uniquely singled out in terms of a particular result Q from $Res(P_v, e)$ plus a maximality or minimality condition with respect to Q. It is therefore possible to define a correspondence between participants d of e and the set $Res(P_v, e)$ ordered by $\leq_v^*$. From the temporal ordering of the results it follows that in addition it is possible to decompose e in such a way that each element of $Res(P_v, e)$ is assigned to exactly one element of the decomposition. The decomposition is defined in terms of the composition relation C. For a given verb v, let j be the number of results $Q \in Res(P_v, e)$ that are not $\approx_v^*$-equivalent. Then, for each $e \in P_v$ a decomposition T_j is defined. For $j = 3$ the definition is $T_3(e) = \{\{e_1, e_2, e_3\} \mid \exists e_{21} \exists e_{321}[C(e_{21}, e_1, e_2) \wedge C(e_{321}, e_{21}, e_3) \wedge e = e_{321}]\}$. In terms of T_3 the following equivalence classes are defined.

(9) a. $[e_1]_e = \{e_1 \mid \exists T_j[T_j(e) \wedge e_1 \in T_j(e) \wedge \forall e'[e' \in T_j(e) \rightarrow e_1 \leq_E e']]\}$

b. $[e_2]_e = \{e_2 \mid \exists T_j[T_j(e) \wedge e_2 \in T_j(e) \wedge \exists e'[e' \in T_j(e) \wedge e' \leq_E e_2] \wedge \exists e'[e' \in T_j(e) \wedge e_2 \leq_E e']]\}$

c. $[e_3]_e = \{e_3 \mid \exists T_j[T_j(e) \wedge e_3 \in T_j(e) \wedge \forall e'[e' \in T_j(e) \rightarrow e' \leq_E e_3]]\}$

The relation $\leq_E$ is defined as: $e \leq_E e'$ iff $\omega(e) \leq_S \alpha(e')$. As the elements of an T_j-decomposition of an $e \in P_v$ are linearly ordered, $T_j(e)$ can be taken as an n-tuple $< e_1 ... e_j >$ such that $\omega(e_i) = \alpha(e_{i+1})$ for $1 \leq i < j$; $\alpha(e) = \alpha(e_1)$ and $\omega(e) = \omega(e_j)$. Furthermore, to each such n-tuple there exists a second n-tuple $< e_1^* ... e_j^* >$ such that each e_i^* is an initial stage of e with e_i as final stage (suffix). The e_i^* are called the *initial extensions* of the e_i. Together, they are a T_j^*-decomposition of e. For T_j^*-triples a precedence ordering is defined by $e_i^* \leq e_{i'}^*$ iff $R_B(e_i, e_{i'})$. The definitions of a T_j decomposition and of the corresponding equivalence classes are lifted to the level of P_v as follows: $T_{j,v} = \{E' \mid \exists e[e \in P_v \wedge E' = T_j(e)]\}$ and $[e_i]_v = \{e \mid \exists e' \exists T[e' \in P_v \wedge T(e') \wedge e \in [e_i]_{e'}]\}$.

Each argument position α of the TS of a verb v uniquely corresponds to a pair $< [e_i]_v, [Q_i]_v >$ in the sense that if an $n+1$ tuple $< d_1, ..., d_n, e >$ belongs to the interpretation of the verb v, then there is a T_j decomposition $< e_1, ..., e_j >$ of e with $j = n$ such that (i) $e_i \in [e_i]_v$, (ii) the result $Q \in Res(P_v, e)$ corresponding to e_i is an element of $[Q_i]_v$ and (iii) Q is the result characterizing d_{n-i+1} relative to e. The argument positions determined by a verb v can therefore be identified with the corresponding pairs $< [e_i]_v, [Q_i]_v >$. κ is a function that assigns to P_v the n-tuple of

pairs $< [e_i]_v, [Q_i]_v >$. The TS of v can then be defined as a tree the leafs of which are the pairs $< [e_i]_v, [Q_i]_v >$.
Note that the relationship between the $[e_i]_v$ and the $[Q_i]_v$ need not be a one-one correspondence because equivalence classes $[Q_i]_v$ and $[Q_j]_v$ the elements of which are $\approx_v$- but not $\approx_v^*$-equivalent are assigned the same equivalence class $[e_k]_v$. The nodes N_{TS} of $\mathbf{T}_{TS}$ are pairs of equivalence classes of events and results. The non-terminal nodes are generalizations of the terminal nodes. One constraint that the set of terminal nodes Θ_{TS} must satisfy is that their order is the inverse of the order determined by κ. This constraint is defined in terms of the leaf relation $R_{leaf_{TS}}$.

(10) a. $(\alpha, \sigma) \in R_{leaf_{TS}}$ iff (i) $\sigma = < p_1, ..., p_n >$, (ii) $R_{\succ^*_{TS}}(\alpha, p_i)$, (iii) $p_i \in \Theta_{TS}$ for $1 \leq i \leq n$ and (iv) $R_>(p_i, p_j)$ for $1 \leq i < j \leq n$
b. if $\kappa(P_v) = < p_1, ..., p_j >$, then $R_{leaf_{ts}}(root_{TS}) = < p_j, ..., p_1 >$

As language for talking about $\mathbf{T}_{TS}$ L^{T_S} can be chosen, i.e. $L^{T_S}(VAR)$ without the base clauses. That means that there is no atomic, i.e. non-structural, information associated with the nodes of $\mathbf{T}_{TS}$. This is a consequence of the fact that in the present context only structural linking is considered for which only the purely structural properties of TS are relevant (see section (2) for this distinction and Naumann/Latrouite (2000) for an alternative formalization of the notion of TS in LDG in which both structural and non-structural linking are analyzed). The strategy of how L^{T_S} and $L^{T_{TS}}$ are combined was already described above. The base clause is (11).

(11) $M, n_s \models < z_{ts} > \phi$ iff $\exists n_{ts}[R_{z_{ts}}(n_s, n_{ts}) \wedge M, n_{ts} \models \phi]$

The relation between $\mathbf{T}_S$ and $\mathbf{T}_{TS}$ is defined in terms of R_{leaf_S}, defined in (12a), and $R_{leaf_{TS}}$. Recall from section (2) that in LDG the default word order (syntactic level) must be the inverse of the argument position in TS which are realized by the argument NPs. In the present context this constraint amounts to the requirement that given the k-tuple of terminal nodes $< n_1, ..., n_k >$ of sort np that is the value of R_{leaf_S}, each n_i belonging to this tuple is mapped to the $k - i + 1$-th element of the k-tuple $< p_1, ..., p_k >$ $(= R_{leaf_{TS}})$ of leafs of $\mathbf{T}_{TS}$.
The mapping is defined as the partial function $R_{z_{ts}}$ from N_S to Θ_{TS}, (12b). As the value of $R_{leaf_{TS}}$ is the inverse of $\kappa(P_v)$, the mapping can equally be defined in terms of κ, (12c). This way of defining the mapping will simplify expressing the constraint in L^{T_S}.

(12) a. $(\alpha, \sigma) \in R_{leaf_S}$ iff (i) $\sigma = < n_1, ..., n_m >$, (ii) $R_{\succ^*_S}(\alpha, n_i)$, (iii) $n_i \in \Theta_S$, (iv) $n_i \in \gamma(R_{z_F}(n_i))$ for $1 \leq i \leq m$ and (iv) $R_{<_S}(n_i, n_j)$ for $1 \leq i < j \leq m$
b. $R_{z_{ts}}(n, p)$ iff $n = \pi^i(R_{leaf_S}(root_S))$ and $p = \pi^{k-i+1}(R_{leaf_{TS}}(root_{TS}))$ for some i such that $1 \leq i \leq k$ and $k =$ the length of $R_{leaf_S}(root_S) = R_{leaf_{TS}}(root_{TS})$
c. $R_{z_{ts}}(n, p)$ iff $n = \pi^i(R_{leaf_S}(root_S))$ and $p = \pi^i(\kappa(P_v))$ for some i such that $1 \leq i \leq k$ and $k =$ the length of $R_{leaf_S}(root_S)$

$R_{z_{ts}}$ differs from R_{z_f} in the way conditions are imposed on the value of the function for a given argument. Whereas R_{z_f} constraints the value of a particular feature, namely the $CASE$ feature, no corresponding condition is imposed by $R_{z_{ts}}$. It constrains only the position of the value and does not impose any requirement on what has to hold at that value. This is reflected in the way the constraint is expressed in L^T: $< z_f > np \wedge t \wedge \alpha \rightarrow < z_{ts} > (\alpha \wedge t)$, where α is a formula expressing information about the relative order of the terminal node at which the whole formula is required to be true among the terminal nodes that are of sort np too, similarly to the constraints expressed in (8) ($\alpha =. \neg < (< z_f > np_t \wedge t)$ or $\alpha =< (< z_f > np \ \wedge t) \wedge \ > (< z_f > np \wedge t)$ or $\alpha = \neg > (< z_f > np \wedge t) \wedge \ < (< z_f > np \wedge t)$). This constraint imposes the constraint that a terminal node of sort np which is at a particular position among the terminal nodes of the same sort (expressed by a) is mapped to a terminal node in the tree spawned by $\kappa(P_v)$ that has the same position in relation to the terminal nodes of $\kappa(P_v)$. From the fact that the leafs of $\kappa(P_v)$ are inverse order to those of $\mathbf{T}_{TS}$ this amounts to the requirement that the default word order determined by v is the inverse of the order of arguments in TS. Note that the antecedents of the constraints in (8) and of the above scheme are identical. Thus, if a constraint in (8) and the corresponding instantiation of the above scheme are required to hold at a node n of $\mathbf{T}_S$, one gets the equivalence $M, [\gamma \circ R_{z_f}](n) \models \ < CASE > \beta$ iff $M, R_{z_{ts}}(n) \models \alpha$ (where β is a variable over atomic values for which the $CASE$ feature is defined). Linguistically, this equivalence requires a position in TS (expressed by α) to be related to a particular assignment of case to an argument-NP realizing this position.

The Level of SF If an event $e \in P_v$ occurs that is denoted by v in the lexicon, e can be decomposed into a $T_j = \{e_1, ..., e_j\}$, for $j =$ the number of results from $Res(P_v, e)$ that are not $\approx_v$-equivalent, such that e_1 brings about the results from $[Q_1]_e$ and no results from equivalence classes following $[Q_1]_e$, e_2 brings about the results from $[Q_2]_e$ and no result from a $[Q_k]_e$, $2 < k \leq j$, that follows $[Q_2]_2$ in the ordering $\leq_v$ (such that at the end of e_2 Q_1 and Q_2 have been brought about) and so on until finally the results from $[Q_j]_e$ are brought about by e_j. Furthermore, as shown above, to each $e_i \in T_j(e)$ there corresponds the largest prefix e' that is an initial stage of e and of which e_i is a final stage. Recall that the SF of a verb v is of the form $[\alpha_1 \ \& \ ... \ \& \ \alpha_n](e)$. Each SF predicate α_i can be interpreted with respect to the initial extension of the i-th event in $T_j(e)$. The α_i can be taken to express that the results from $[Q_i]_e$ corresponding to e_i is brought about.
A suitable formalism for expressing these relationships is (an extension of) *Dynamic Modal Arrow Logic* (DMAL), Naumann/Osswald (1999). DMAL combines Dynamic Modal Logic and Arrow Logic, van Benthem (1991). As shown in Naumann/Osswald (1999), the way a result from $[Q_i]_e$ is evaluated on the initial extension of e_i is captured by a dynamic mode, i.e. a function that maps pairs of sets of events and sets of states to sets of events. In DMAL a verb v is translated by formulas of the form $\pi_1 \ \& \ ... \ \& \ \pi_n$ where each π_i is of the form $DM(\sqsubseteq_v, \phi)$ or $DM(\sqsubseteq_v, \phi) \cap DM'(\sqsubseteq_v, \psi)$ for DM, DM' a dynamic mode. The satisfaction clause for & is (13) (σ is a finite sequence of events; σ_ϵ is the empty sequence).

(13) $M, e, \sigma \models \pi \ \& \ \pi'$ iff there is an e_1 such that (i) $R_B(e_1, e)$; (ii) $R_B(e_1, last(\sigma)) \vee \sigma = \sigma_\epsilon$; (iii) $M, e_1, \sigma \models \pi$ and (iv) $M, e, < \sigma \mid e_1 > \models \pi'$

(13) is an instance of a translation scheme because the second argument of the DM are dependent on a particular event $e \in P_v$. The translation of v in DMAL is therefore not unique. Yet, for a given $e \in P_v$ one has the following. Let $\pi_1 \ \& \ ... \ \& \ \pi_n$ be a DMAL translation of v. Then $M, e, \sigma \models \pi_1 \ \& \ ... \ \& \ \pi_n$ iff there is a T_n decomposition $< e_1, ..., e_n >$ of e such that (i) $R_B(e, e_i)$, $1 \leq i \leq n$, (ii) $R_B(e_i, e_j)$, $1 \leq i < j \leq n$, and (iii) $M, e_i, < \sigma, < e_1, ..., e_{i-1} >> \models \pi_i$. The tuple $< e_1, ..., e_n >$ need not be

unique. Rather, they are a particular T_n decomposition of e. What is common to all T_n decompositions that make π_1 & ... & π_n true is captured in terms of restrictions of the equivalence classes $[e_i]_{v,e}$ induced by T_n according to what has been said at the beginning of this section: $[e_i^*]_{v,e} = \{e' \mid e' \in [e_i]_{v,e} \wedge \forall Q[Q \in [Q_i]_{v,e} \rightarrow [\omega(e') \in Q \wedge . \forall Q'[Q' \in Res(P_v, e') \wedge Q <_v Q' \rightarrow \omega(e') \notin Q']]]\}$. The $[e_i^*]_{v,e}$ are the leafs of a tree which are labeled by formulas from L^{DMAL}. A language to talk about this tree is $L^{DMAL}(L^{T_{TS}})$. This language is L^{DMAL} augmented by formulas of the form $< z'_{ts} > \phi$ for ϕ an $L^{T_{TS}}$ well-formed formula. The atomic information associated with the nodes are subformulas of π_1 & ... & π_n. For the i-th leaf it is π_i. The trees T_{TS} and $\{T_{SF_e} \mid e \in R_v\}$ are related by a function R_{SF} with domain the nodes of the trees in $\{T_{SF_e} \mid e \in R_v\}$, i.e. $R_v \cup \{e \mid \exists e'[e' \in R_v \wedge R_B(e, e')]\}$ ($R_v = \{e \mid e \in [\![Tr(v)]\!]\}$ is the set of all events $e \in P_v$ denoted by v in the lexicon; $Tr(v)$ is the set of translations of v) and range N_{TS} that assigns to each e in the domain a node in T_{TS}.

5 Conclusion

In LDG a verb is analyzed at the grammatical levels of Semantic Form, Theta-Structure and the syntactic-morphological level. In the formalization presented in this paper each level is modeled by a mathematical structure. Constraints between the different levels that restrict the distribution of information are defined in terms of mappings that relate the domains of the structures with each other. The constraints are formulated by *fibring* or *layering* the different languages. The base clauses are modal formulas of the form $< \alpha > \phi$, where α is interpreted as the relation R_α that links elements of two domains. E.g., the constraint between the syntactic and the morphological level concerns the (default) constituent order and the assignment of case. For structural linking, particular positions in the order of constituents are related to particular cases. The (default) constituent order is represented by (the linear ordering of) the terminal nodes of the tree structure $\mathbf{T}_S$. The morphological (case) information is encoded in the feature structure corresponding to a terminal node (of category np). The constraint therefore requires that if a terminal node n of category np satisfies ϕ, expressing information about the position of n in the constituent order, then the corresponding feature structure must satisfy ψ, expressing information about the case of the constituent: $M, n \models \phi \Rightarrow M, \gamma[R_{z_f}(n)] \models \psi$, where R_{z_f} maps an $n \in N_S$ to its corresponding feature structure and $\gamma(\mathbf{FN})$ is the root of $\mathbf{FN}$.

References

[1] Blackburn, P. et al. (1993). Talking about Trees, in: *Proc. 6th Conference of the European Chapter of the ACL*, pp. 21-29

[2] Blackburn, P./Meyer-Viol, W. (1997). Modal Logic and Model-Theoretic Syntax, in: M. de Rijke (ed.): *Advances in Intensional Logic*, Kluwer Academic Publishers, pp. 29-60

[3] Blackburn, P./de Rijke, M. (1997). Zooming In, Zooming Out, *JoLLI*, vol.6, pp. 5-31

[4] Naumann, R. (1999). A Dynamic Approach to Aspect: Verbs as Programs, to appear in *Journal of Semantics*

[5] Naumann, R. (2000). Events and Objects, to appear *Proc. 7th Symposium on Logic and Language*, Budapest, Hungary

[6] Naumann, R./Latrouite A. (2000). Linking and the Dynamic Structure of Verbs: Underspecification at the morphological-semantic Interface, Obrst, L. (ed.): *Proceedings of the Workshop on Semantic Approximation, Vagueness and Granularity*, AAAI Publications

[7] Naumann, R./Osswald, R. (1999). A Dynamic Modal Arrow Logic for the Analysis of Aspectual Phenomena in Natural Language, to appear in *Proceedings 3rd International Tbilisi Symposium on Logic, Language and Computation*

[8] Pustejovsky, J. (1991). The generative Lexicon, *Computational Linguistics*, vol.17, pp.409-441

[9] van Benthem, J. (1991). *Language in Action*, Amsterdam, North-Holland

[10] Wunderlich, D. (1997). CAUSE and the Structure of Verbs, *Linguistic Inquiry*, vol. 28, pp.27-68