# Incremental Specialization of an HPSG-Based Annotation Scheme

**Kiril Simov, Milen Kouylekov, Alexander Simov**

BulTreeBank Project
http://www.BulTreeBank.org
Linguistic Modelling Laboratory, Bulgarian Academy of Sciences
Acad. G. Bonchev St. 25A, 1113 Sofia, Bulgaria
kivs@bgcict.acad.bg, mkouylekov@dir.bg, adis_78@dir.bg

**Abstract**

The linguistic knowledge represented in contemporary language resource annotations becomes very complex. Its acquiring and management requires an enormous amount of human work. In order to minimize such a human effort we need rigorous methods for representation of such knowledge, methods for supporting the annotation process, methods for exploiting all results from the annotation process, even those that usually disappear after the annotation has been completed. In this paper we present a formal set-up for annotation within HPSG linguistic theory. We present also an algorithm for annotation scheme specialization based on the negative information from the annotation process. The negative information includes the analyses, rejected by the annotator.

## 1. Introduction

In our project (Simov et. al., 2001a), (Simov et al., 2002) we aim at the creation of syntactically annotated corpus (treebank) based on the HPSG linguistic theory (Head-driven Phrase Structure Grammar — (Pollard and Sag, 1987) and (Pollard and Sag, 1994)). Hence, the elements of the treebank are not trees, but feature graphs. The annotation scheme for the construction of the treebank is based on the appropriate language-specific version of the HPSG sort hierarchy. On one hand, such an annotation scheme is very detailed and flexible with respect to the linguistic knowledge, encoded in it. But, on the other hand, because of the massive overgeneration, it is not considered to be annotator-friendly. Thus, the main problem is: how to keep the consistency of the annotation scheme and at the same time to minimize the human work during the annotation. In our annotation architecture we envisage two sources of linguistic knowledge in order to reduce the possible analyses of the annotated sentences:

- Reliable partial grammars.
- HPSG-based grammar: universal principles, language specific principles and a lexicon.

The actual annotation process includes the following steps:

- Partial parsing step:

  This step comprises several additional steps: (1) Sentence extraction from the text archive; (2) Morphosyntactic tagging; (3) Part-of-speech disambiguation; (4) Partial parsing;

  The result is considered a 100 % accurate partial parsed sentence.

- HPSG step:

  The result from the previous step is encoded into an HPSG compatible representation with respect to the sort hierarchy. It is sent to an HPSG grammar tool, which takes the partial sentence analysis as input and evaluates all the attachment possibilities for it. The output is encoded as feature graphs.

- Annotation step:

  The feature graphs from the previous step are further processed as follows : (1) their intersection is calculated; (2) on the base of the differences, a set of constraints over the intersection is calculated as well; (3) during the actual annotation step, the annotator tries to extend the intersection to full analysis, adding new information to it. The constraints determine the possible extensions and also propagate the information, added by the annotator, in order to minimize the incoming choices.

This architecture is being currently implemented by establishing an interface between two systems: CLaRK system for XML based corpora development (Simov et. al., 2001b) and TRALE system for HPSG grammar development (TRALE is a descendant of (Götz and Meurers, 1997)). The project will result in an HPSG corpus based on feature graphs and reliable grammars. One of the intended applications of these language resources consists of their exploration for improving the accuracy of the implemented HPSG grammar.

The work, reported in this paper, is a step towards establishing an incremental mechanism, which uses already annotated sentences for further specializing of the HPSG grammar and for reducing the number of the possible HPSG analyses. In fact, we consider the rejected analyses as negative information about the language and therefore the grammar has to be appropriately tuned in order to rule out such analyses.

The structure of the paper is as follows: in the next section we define formally what a corpus is with respect to a grammar formalism and apply this definition to the definition of an HPSG corpus. In Sect. 3. we present a logical formalism for HPSG, define a normal form for grammars in the logical formalism and on the basis of this normal form we define feature graphs that constitute a good representation for both — HPSG grammars and HPSG corpora. Sect. 4. presents the algorithm for specialization of an

HPSG grammar on the basis of accepted and rejected by the annotator analyses produced by the grammar. Then Sect. 5. demonstrates an example of such specialization. The last section outlines the conclusions and outlook.

## 2. HPSG Corpus

In our work we accept that the corpus is complete with respect to the analyses of the sentences in it. This means that each sentence is presented with all its acceptable syntactic structures. Thus a good grammar will not overgenerate, i.e. it will not assign more analyses to the sentences than the analyses, which already exist in the corpus. Before we define what an HPSG corpus is like, let us start with a definition of a grammar-formalism-based corpus in general. Such an ideal corpus has to ensure the above assumption.

**Definition 1 (Grammar Formalism Corpus)** *A corpus $C$ in a given grammatical formalism $G$ is a sequence of analyzed sentences where each analyzed sentence is a member of the set of structures defined as a* **strong generative capacity** *(SGC) of a grammar $\Gamma$ in this grammatical formalism:*

$\forall S.S \in C \rightarrow S \in \mathtt{SGC}(\Gamma)$,

*where $\Gamma$ is a grammar in the formalism $G$, and if $\sigma(S)$ is the phonological string of $S$ and $\Gamma(\sigma(S))$ is the set of all analyses assigned by the grammar $\Gamma$ to the phonological string $\sigma(S)$, then*

$\forall S'.S' \in \Gamma(\sigma(S)) \rightarrow S' \in C$.

The grammar $\Gamma$ is unknown, but implicitly represented in the corpus $C$. We could state that if such a grammar does not exist, then we consider the corpus inconsistent or uncomplete.

In order to define a corpus in HPSG with respect to this definition, we have to define a representation of HPSG analysis over the sentences. This analysis must correspond to a definition of strong generative capacity in HPSG. Fortunately, there exist such definitions - (King 1999) and (Pollard 1999). We adopt them for our purposes. Thus in our work we choose:

- A logical formalism for HPSG — King's Logic (SRL) (King 1989);
- A definition of strong generative capacity in HPSG as a set of feature structures closely related to the special interpretation in SRL (exhaustive models) along the lines of (King 1999) and (Pollard 1999).
- A definition of corpus in HPSG as a sequence of sentences that are members of $\mathtt{SGC}(\Gamma)$ for some grammar $\Gamma$ in SRL.

It is well-known that an HPSG grammar in SRL formally comprises two parts: a signature and a theory. The signature defines the ontology of the linguistic objects in the language and the theory constraints the shape of the linguistic objects. Usually the descriptions in the theory part are presented as implications. In order to demonstrate in a better way the connection between the HPSG grammar in SRL and the HPSG corpus, we offer a common representation of the grammar and the corpus.

We define a normal form for HPSG grammars which ideologically is very close to the feature structures defining the strong generative capacity in HPSG as it has proposed in the work of (King 1999) and (Pollard 1999). We define both the corpus and the grammar in terms of clauses (considered as graphs) in a special kind of matrices in SRL. The construction of new sentence analyses can be done using the inference mechanisms of SRL. Another possibility is such a procedure to be defined directly using the representations in the normal form. In order to distinguish the elements in our normal form from the numerous of kinds of feature structures we call the elements in the normal form *feature graphs*. One important characteristic about our feature graphs is that they are viewed as descriptions in SRL, i.e. as syntactic entities.

In other works (Simov, 2001) and (Simov, 2002) we showed how from a corpus, consisting of feature graphs, a corpus grammar could be extracted along the lines of Rens Bod's ideas on Data-Oriented Parsing Model (Bod, 1998). Also, in (Simov, 2002) we showed how one could use the positive information in the corpus in order to refine an existing HPSG grammar. In this paper we discuss and illustrate the usage of the negative information compiled as a by-product during the annotation of the corpus.

## 3. Logical Formalism for HPSG

In this section we present a logical formalism for HPSG. Then a normal form (exclusive matrices) for a finite theory in this formalism is defined and then we show how it can be represented as a set of feature graphs. These graphs are considered a representation of grammars and corpora in HPSG.

### 3.1. King's Logic — SRL

This section presents the basic notions of Speciate Reentrancy Logic (SRL) (King 1989).

$\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$ is a finite **SRL signature** iff $\mathcal{S}$ is a finite set of *species*, $\mathcal{F}$ is a set of *features*, and $\mathcal{A} : \mathcal{S} \times \mathcal{F} \rightarrow Pow(\mathcal{S})$ is an *appropriateness function*. $\mathcal{I} = \langle \mathcal{U}_\mathcal{I}, \mathcal{S}_\mathcal{I}, \mathcal{F}_\mathcal{I} \rangle$ is a **SRL interpretation** of the signature $\Sigma$ (or $\Sigma$-interpretation) iff

$\mathcal{U}_\mathcal{I}$ is a non-empty set of objects,
$\mathcal{S}_\mathcal{I}$ is a total function from $\mathcal{U}_\mathcal{I}$ to $\mathcal{S}$,
called **species assignment function**,
$\mathcal{F}_\mathcal{I}$ is a total function from $\mathcal{F}$ to the set of partial function from $\mathcal{U}_\mathcal{I}$ to $\mathcal{U}_\mathcal{I}$ such that
for each $\phi \in \mathcal{F}$ and each $v \in \mathcal{U}_\mathcal{I}$, if $\mathcal{F}_\mathcal{I}(\phi)(v)\downarrow$[1]
then $\mathcal{S}_\mathcal{I}(\mathcal{F}_\mathcal{I}(\phi)(v)) \in \mathcal{A}(\mathcal{S}_\mathcal{I}(v), \phi)$, and
for each $\phi \in \mathcal{F}$ and each $v \in \mathcal{U}_\mathcal{I}$, if $\mathcal{A}(\mathcal{S}_\mathcal{I}(v), \phi)$ is not empty then $\mathcal{F}_\mathcal{I}(\phi)(v)\downarrow$,
$\mathcal{F}_\mathcal{I}$ is called **feature interpretation function**.

$\tau$ is a term iff $\tau$ is a member of the smallest set $\mathcal{TM}$ such that (1) $: \in \mathcal{TM}$, and (2) for each $\phi \in \mathcal{F}$ and each $\tau \in \mathcal{TM}$, $\tau\phi \in \mathcal{TM}$. For each $\Sigma$-interpretation $\mathcal{I}$, $\mathcal{P}_\mathcal{I}$ is a **term interpretation function** over $\mathcal{I}$ iff (1) $\mathcal{P}_\mathcal{I}(:)$ is the identity function from $\mathcal{U}_\mathcal{I}$ to $\mathcal{U}_\mathcal{I}$, and (2) for each $\phi \in \mathcal{F}$ and each $\tau \in \mathcal{TM}$, $\mathcal{P}_\mathcal{I}(\tau\phi)$ is the composition of the partial functions $\mathcal{P}_\mathcal{I}(\tau)$ and $\mathcal{F}_\mathcal{I}(\phi)$ if they are defined.

[1] $f(o)\downarrow$ means the function $f$ is defined for the argument $o$.

$\delta$ is a **description** iff $\delta$ is a member of the smallest set $\mathcal{D}$ such that (1) for each $\sigma \in \mathcal{S}$ and for each $\tau \in \mathcal{TM}$, $\tau \sim \sigma \in \mathcal{D}$, (2) for each $\tau_1 \in \mathcal{TM}$ and $\tau_2 \in \mathcal{TM}$, $\tau_1 \approx \tau_2 \in \mathcal{D}$ and $\tau_1 \not\approx \tau_2 \in \mathcal{D}$, (3) for each $\delta \in \mathcal{D}$, $\neg\delta \in \mathcal{D}$, (4) for each $\delta_1 \in \mathcal{D}$ and $\delta_2 \in \mathcal{D}$, $[\delta_1 \wedge \delta_2] \in \mathcal{D}$, $[\delta_1 \vee \delta_2] \in \mathcal{D}$, and $[\delta_1 \rightarrow \delta_2] \in \mathcal{D}$. **Literals** are descriptions of the form $\tau \sim \sigma$, $\tau_1 \approx \tau_2$, $\tau_1 \not\approx \tau_2$ or their negation. For each $\Sigma$-interpretation $\mathcal{I}$, $\mathcal{D}_\mathcal{I}$ is a **description denotation function** over $\mathcal{I}$ iff $\mathcal{D}_\mathcal{I}$ is a total function from $\mathcal{D}$ to the powerset of $\mathcal{U}_\mathcal{I}$, such that

$$\begin{aligned}
&\mathcal{D}_\mathcal{I}(\tau \sim \sigma) = \{\upsilon \in \mathcal{U}_\mathcal{I} \mid \mathcal{P}_\mathcal{I}(\tau)(\upsilon)\downarrow,\\
&\qquad \mathcal{S}_\mathcal{I}(\mathcal{P}_\mathcal{I}(\tau)(\upsilon)) = \sigma\},\\
&\mathcal{D}_\mathcal{I}(\tau_1 \approx \tau_2) = \{\upsilon \in \mathcal{U}_\mathcal{I} \mid \mathcal{P}_\mathcal{I}(\tau_1)(\upsilon)\downarrow, \mathcal{P}_\mathcal{I}(\tau_2)(\upsilon)\downarrow,\\
&\qquad \text{and } \mathcal{P}_\mathcal{I}(\tau_1)(\upsilon) = \mathcal{P}_\mathcal{I}(\tau_2)(\upsilon)\},\\
&\mathcal{D}_\mathcal{I}(\tau_1 \not\approx \tau_2) = \{\upsilon \in \mathcal{U}_\mathcal{I} \mid \mathcal{P}_\mathcal{I}(\tau_1)(\upsilon)\downarrow, \mathcal{P}_\mathcal{I}(\tau_2)(\upsilon)\downarrow,\\
&\qquad \text{and } \mathcal{P}_\mathcal{I}(\tau_1)(\upsilon) \neq \mathcal{P}_\mathcal{I}(\tau_2)(\upsilon)\},\\
&\mathcal{D}_\mathcal{I}(\neg\delta) = \mathcal{U}_\mathcal{I} \setminus \mathcal{D}_\mathcal{I}(\delta),\\
&\mathcal{D}_\mathcal{I}([\delta_1 \wedge \delta_2]) = \mathcal{D}_\mathcal{I}(\delta_1) \cap \mathcal{D}_\mathcal{I}(\delta_2),\\
&\mathcal{D}_\mathcal{I}([\delta_1 \vee \delta_2]) = \mathcal{D}_\mathcal{I}(\delta_1) \cup \mathcal{D}_\mathcal{I}(\delta_2), \text{ and}\\
&\mathcal{D}_\mathcal{I}([\delta_1 \rightarrow \delta_2]) = (\mathcal{U}_\mathcal{I} \setminus \mathcal{D}_\mathcal{I}(\delta_1)) \cup \mathcal{D}_\mathcal{I}(\delta_2).
\end{aligned}$$

Each subset $\theta \subseteq \mathcal{D}$ is an **SRL theory**. For each $\Sigma$-interpretation $\mathcal{I}$, $\mathcal{T}_\mathcal{I}$ is a **theory denotation function** over $\mathcal{I}$ iff $\mathcal{T}_\mathcal{I}$ is a total function from the powerset of $\mathcal{D}$ to the powerset of $\mathcal{U}_\mathcal{I}$ such that for each $\theta \subseteq \mathcal{D}$, $\mathcal{T}_\mathcal{I}(\theta) = \cap\{\mathcal{D}_\mathcal{I}(\delta) | \delta \in \theta\}$. $\mathcal{T}_\mathcal{I}(\emptyset) = \mathcal{U}_\mathcal{I}$. A theory $\theta$ is **satisfiable** iff for some interpretation $\mathcal{I}$, $\mathcal{T}_\mathcal{I}(\theta) \neq \emptyset$. A theory $\theta$ is **modelable** iff for some interpretation $\mathcal{I}$, $\mathcal{T}_\mathcal{I}(\theta) = \mathcal{U}_\mathcal{I}$, $\mathcal{I}$ is called a **model** of $\theta$. The interpretation $\mathcal{I}$ **exhaustively models** $\theta$ iff

$\mathcal{I}$ is a model of $\theta$, and
for each $\theta' \subseteq \mathcal{D}$,
if for some model $\mathcal{I}'$ of $\theta$,
$\mathcal{T}_{\mathcal{I}'}(\theta') \neq \emptyset$,
then $\mathcal{T}_\mathcal{I}(\theta') \neq \emptyset$.

An HPSG grammar $\Gamma = \langle \Sigma, \theta \rangle$ in SRL consists of: (1) a signature $\Sigma$ which gives the ontology of entities that exist in the universe and the appropriateness conditions on them, and (2) a theory $\theta$ which gives the restrictions upon these entities.

## 3.2. Exclusive Matrices

Following (King and Simov, 1998) in this section we define a normal form for finite theories in SRL — called **exclusive matrix**. This normal form possesses some desirable properties for representation of grammars and corpora in HPSG.

First, we define some technical notions. A **clause** is a finite set of literals interpreted *conjunctively*. A **matrix** is a finite set of clauses interpreted *disjunctively*.

A matrix $\mu$ is an **exclusive matrix** iff for each clause $\alpha \in \mu$,

(E0) if $\lambda \in \alpha$ then $\lambda$ is a positive literal,
(E1) $: \approx : \in \alpha$,
(E2) if $\tau_1 \approx \tau_2 \in \alpha$ then $\tau_2 \approx \tau_1 \in \alpha$,
(E3) if $\tau_1 \approx \tau_2 \in \alpha$ and $\tau_2 \approx \tau_3 \in \alpha$ then $\tau_1 \approx \tau_3 \in \alpha$,
(E4) if $\tau\phi \approx \tau\phi \in \alpha$ then $\tau \approx \tau \in \alpha$,
(E5) if $\tau_1 \approx \tau_2 \in \alpha$, $\tau_1\phi \approx \tau_1\phi \in \alpha$ and $\tau_2\phi \approx \tau_2\phi \in \alpha$ then $\tau_1\phi \approx \tau_2\phi \in \alpha$,
(E6) if $\tau \approx \tau \in \alpha$ then for some $\sigma \in \mathcal{S}$, $\tau \sim \sigma \in \alpha$,
(E7) if for some $\sigma \in \mathcal{S}$, $\tau \sim \sigma \in \alpha$ then $\tau \approx \tau \in \alpha$,
(E8) if $\tau_1 \approx \tau_2 \in \alpha$, $\tau_1 \sim \sigma_1 \in \alpha$ and $\tau_2 \sim \sigma_2 \in \alpha$ then $\sigma_1 = \sigma_2$,
(E9) if $\tau \sim \sigma_1 \in \alpha$ and $\tau\phi \sim \sigma_2 \in \alpha$ then $\sigma_2 \in \mathcal{A}(\sigma_1, \phi)$,
(E10) if $\tau \sim \sigma \in \alpha$, $\tau\phi \in \mathit{Term}(\mu)$ and $\mathcal{A}(\sigma, \phi) \neq \emptyset$ then $\tau\phi \approx \tau\phi \in \alpha$,
(E11) if $\tau_1 \not\approx \tau_2 \in \alpha$ then $\tau_1 \approx \tau_1 \in \alpha$ and $\tau_2 \approx \tau_2 \in \alpha$,
(E12) if $\tau_1 \approx \tau_1 \in \alpha$ and $\tau_2 \approx \tau_2 \in \alpha$ then $\tau_1 \approx \tau_2 \in \alpha$ or $\tau_1 \not\approx \tau_2 \in \alpha$, and
(E13) $\tau_1 \approx \tau_2 \notin \alpha$ or $\tau_1 \not\approx \tau_2 \notin \alpha$,

where $\{\sigma, \sigma_1, \sigma_2\} \subseteq \mathcal{S}$, $\phi \in \mathcal{F}$, and $\{\tau, \tau_1, \tau_2, \tau_3\} \subseteq \mathcal{TM}$, and `Term` is a function from the powerset of the sets of literals to the powerset of $\mathcal{TM}$ such that

$$\begin{aligned}
\texttt{Term}(\alpha) = &\{\tau \mid (\neg)\tau\phi \approx \tau' \in \alpha, \tau \in \mathcal{TM}, \phi \in \mathcal{F}^*\} \cup\\
&\{\tau \mid (\neg)\tau' \approx \tau\phi \in \alpha, \tau \in \mathcal{TM}, \phi \in \mathcal{F}^*\} \cup\\
&\{\tau \mid (\neg)\tau\phi \not\approx \tau' \in \alpha, \tau \in \mathcal{TM}, \phi \in \mathcal{F}^*\} \cup\\
&\{\tau \mid (\neg)\tau' \not\approx \tau\phi \in \alpha, \tau \in \mathcal{TM}, \phi \in \mathcal{F}^*\} \cup\\
&\{\tau \mid (\neg)\tau\phi \sim \sigma \in \alpha, \tau \in \mathcal{TM}, \phi \in \mathcal{F}^*\}.
\end{aligned}$$

There are two important properties of an exclusive matrix $\mu = \{\alpha_1, \ldots, \alpha_n\}$: (1) each clause $\alpha$ in $\mu$ is satisfiable (for some interpretation $\mathcal{I}$, $\mathcal{T}_\mathcal{I}(\alpha) \neq \emptyset$), and (2) each two clauses $\alpha_1$, $\alpha_2$ in $\mu$ have disjoint denotations (for each interpretation $\mathcal{I}$, $\mathcal{T}_\mathcal{I}(\alpha_1) \cap \mathcal{T}_\mathcal{I}(\alpha_1) = \emptyset$). Also in (King and Simov, 1998) it is shown that each finite theory with respect to a finite signature can be converted into an exclusive matrix which is semantically equivalent to the theory. Relying on the definition of model (where each object in the domain is described by the theory) and the property that each two clauses in an exclusive matrix have disjoint denotation, one can easy prove the following proposition.

**Proposition 2** *Let $\theta$ be a finite SRL theory with respect to a finite signature, $\mu$ be the corresponding exclusive matrix and $\mathcal{I} = \langle \mathcal{U}, \mathcal{S}, \mathcal{F} \rangle$ be a model of $\theta$. For each object $\upsilon \in \mathcal{U}$ there exists a unique clause $\alpha \in \mu$ such that $\upsilon \in \mathcal{T}(\alpha)$.*

## 3.3. Feature Graphs

As it was mentioned above, an HPSG corpus will comprise a set of feature structures representing the HPSG analyses of the sentences. We interpret these feature structures as descriptions in SRL (clauses in an exclusive matrix).

Let $\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$ be a finite signature. A directed, connected and rooted graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{V}, \rho, \mathcal{S} \rangle$ such that

$\mathcal{N}$ is a set of **nodes**,
$\mathcal{V} : \mathcal{N} \times \mathcal{F} \rightarrow \mathcal{N}$ is a partial **arc function**,
$\rho$ is a **root node**,
$\mathcal{S} : \mathcal{N} \rightarrow \mathcal{S}$ is a total **species assignment function**,
such that
for each $\nu_1, \nu_2 \in \mathcal{N}$ and each $\phi \in \mathcal{F}$
if $\mathcal{V}\langle \nu_1, \phi \rangle\downarrow$ and $\mathcal{V}\langle \nu_1, \phi \rangle = \nu_2$,
then $\mathcal{S}\langle \nu_2 \rangle \in \mathcal{A}\langle \mathcal{S}\langle \nu_1 \rangle, \phi \rangle$,

is a **feature graph** wrt $\Sigma$.

A feature graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{V}, \rho, \mathcal{S} \rangle$ such that for each node $\nu \in \mathcal{N}$ and each feature $\phi \in \mathcal{F}$ if $\mathcal{A}\langle \mathcal{S}\langle \nu \rangle, \phi \rangle \downarrow$ then $\mathcal{V}\langle \nu, \phi \rangle \downarrow$ is called a **complete feature graph** (or complete graph).

According to our definition feature graphs are a kind of feature structures which are treated syntactically rather than semantically. We use complete feature graphs for representing the analyses of the sentences in the corpus.

We say that the feature graph $\mathcal{G}$ is **finite** if and only if the set of nodes is finite.

For each graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{V}, \rho, \mathcal{S} \rangle$ and node $\nu$ in $\mathcal{N}$ with $\mathcal{G}\,|_{\nu} = \langle \mathcal{N}_{\nu}, \mathcal{V}\,|_{\mathcal{N}_{\nu}}, \rho_{\nu}, \mathcal{S}\,|_{\mathcal{N}_{\nu}} \rangle$ we denote the **subgraph** of $\mathcal{G}$ starting on node $\nu$.

Let $\mathcal{G}_1 = \langle \mathcal{N}_1, \mathcal{V}_1, \rho_1, \mathcal{S}_1 \rangle$ and $\mathcal{G}_2 = \langle \mathcal{N}_2, \mathcal{V}_2, \rho_2, \mathcal{S}_2 \rangle$ be two graphs. We say that graph $\mathcal{G}_1$ **subsumes** graph $\mathcal{G}_2$ ($\mathcal{G}_2 \sqsubseteq \mathcal{G}_1$) iff there is an *isomorphism* $\gamma : \mathcal{N}_1 \to \mathcal{N}_2'$, $\mathcal{N}_2' \subseteq \mathcal{N}_2$, such that

$\gamma(\rho_1) = \rho_2$,

for each $\nu, \nu' \in \mathcal{N}_1$ and each feature $\phi$,

$\mathcal{V}_1\langle \nu, \phi \rangle = \nu'$ iff $\mathcal{V}_2\langle \gamma(\nu), \phi \rangle = \gamma(\nu')$, and

for each $\nu \in \mathcal{N}_1$, $\mathcal{S}_1\langle \nu \rangle = \mathcal{S}_2\langle \gamma(\nu) \rangle$.

The intuition behind the definition of subsumption by isomorphism is that each graph describes "exactly" a chunk in some SRL interpretation in such a way that every two distinct nodes are always mapped to distinct objects in the interpretation.

For each two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ if $\mathcal{G}_2 \sqsubseteq \mathcal{G}_1$ and $\mathcal{G}_1 \sqsubseteq \mathcal{G}_2$ we say that $\mathcal{G}_1$ and $\mathcal{G}_2$ are **equivalent.** For convenience, in the following text we consider each two equivalent graphs equal.

For a finite feature graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{V}, \rho, \mathcal{S} \rangle$, we define a translation to a clause. Let

$Term(\mathcal{G}) = \{:\} \cup \{\tau \mid \tau \doteq :\phi_1 \ldots \phi_n, n \leq \|\mathcal{N}\|, \mathcal{V}\langle \rho, \tau \rangle \downarrow\}$[2]

be a set of terms. We define a clause $\alpha_{\mathcal{G}}$:

$\alpha_{\mathcal{G}} = \{\tau \sim \sigma \mid \tau \in Term(\mathcal{G}), \mathcal{V}\langle \rho, \tau \rangle \downarrow, \mathcal{S}\langle \mathcal{V}\langle \rho, \tau \rangle \rangle = \sigma\} \cup$
$\{\tau_1 \approx \tau_2 \mid \tau_1 \in Term(\mathcal{G}), \tau_2 \in Term(\mathcal{G}),$
$\mathcal{V}\langle \rho, \tau_1 \rangle \downarrow, \mathcal{V}\langle \rho, \tau_2 \rangle \downarrow$, and $\mathcal{V}\langle \rho, \tau_1 \rangle = \mathcal{V}\langle \rho, \tau_2 \rangle\} \cup$
$\{\tau_1 \not\approx \tau_2 \mid \tau_1 \in Term(\mathcal{G}), \tau_2 \in Term(\mathcal{G}),$
$\mathcal{V}\langle \rho, \tau_1 \rangle \downarrow, \mathcal{V}\langle \rho, \tau_2 \rangle \downarrow$, and $\mathcal{V}\langle \rho, \tau_1 \rangle \neq \mathcal{V}\langle \rho, \tau_2 \rangle\}$.

We interpret a finite feature graph via the interpretation of the corresponding clauses

$\mathcal{R}_{\mathcal{I}}(\mathcal{G}) = \mathcal{T}_{\mathcal{I}}(\alpha_{\mathcal{G}})$.

Let $\mathcal{G}$ be an infinite feature graph. Then we interpret it as the intersection of the interpretations of all finite feature graphs that subsume it:

$\mathcal{R}_{\mathcal{I}}(\mathcal{G}) = \cap_{\mathcal{G} \sqsubseteq \mathcal{G}', \mathcal{G}' <_{\omega}} \mathcal{R}_{\mathcal{I}}(\alpha_{\mathcal{G}'})$.

The clauses in an exclusive matrix $\mu$ can be represented as feature graphs. Let $\mu$ be an exclusive matrix and $\alpha \in \mu$, then

$\mathcal{G}_{\alpha} = \langle \mathcal{N}_{\alpha}, \mathcal{V}_{\alpha}, \rho_{\alpha}, \mathcal{S}_{\alpha} \rangle$ is a feature graph such that

$\mathcal{N}_{\alpha} = \{|\tau|_{\alpha} \mid \tau \approx \tau \in \alpha\}$ is a set of nodes,

$\mathcal{V}_{\alpha} : \mathcal{N}_{\alpha} \times \mathcal{F} \to \mathcal{N}_{\alpha}$ is a partial **arc function**, such that

$\mathcal{V}_{\alpha}\langle |\tau_1|_{\alpha}, \phi \rangle \downarrow$ and $\mathcal{V}_{\alpha}\langle |\tau_1|_{\alpha}, \phi \rangle = |\tau_2|_{\alpha}$ iff

$\tau_1 \approx \tau_1 \in \alpha, \tau_2 \approx \tau_2 \in \alpha$, $\phi \in \mathcal{F}$, and $\tau_1\phi \approx \tau_2 \in \alpha$,

$\rho_{\alpha}$ is the root node $|:|_{\alpha}$, and

$\mathcal{S}_{\alpha} : \mathcal{N}_{\alpha} \to \mathcal{S}$ is a **species assignment function**, such that

$\mathcal{S}_{\alpha}\langle |\tau|_{\alpha} \rangle = \sigma$ iff $\tau \sim \sigma \in \alpha$.

**Proposition 3** *Let $\mu$ be an exclusive matrix and $\alpha \in \mu$. Then the graph $\mathcal{G}_{\alpha}$ is semantically equivalent to $\alpha$.*

## 3.4. Inference with Feature Graphs

In this paper we do not present a concrete inference mechanism exploiting feature graphs. As it was mentioned above, one can use the general inference mechanisms of SRL in order to construct sentence analyses. However, a much better solution is to employ an inference mechanism, which uses directly the graph representation of a theory. Such an inference mechanism can be defined along the lines of *Breadth-First Parallel Resolution* in (Carpener 1992) despite the difference in the treatment of the feature structure in (Carpener 1992) (Note that (Carpener 1992) treats feature structures as semantic entities, but we consider our feature graphs syntactic elements.). One has to keep in mind that finding models in SRL is undecidable (see (King, Simov and Aldag 1999)) and some restrictions in terms of time or memory will be necessary in order to use Breadth-First Parallel Resolution-like algorithm. A presentation of such an algorithm is beyond the scope of this paper.

[2]$\|X\|$ is the cardinality of the set $X$

## 3.5. Graph Representation of an SRL Theory

Each finite SRL theory can be represented as a set of feature graphs. In order to make this graph transformation of a theory completely independent from the SRL particulars, we also need to incorporate within the graphs the information from the signature that is not present in the theory yet. For each species the signature encodes the defined features as well as the species of their possible values. We explicate this information in the signature by constructing a special theory:

$$\theta_{\Sigma} = \{ \bigvee_{\sigma \in \mathcal{S}} [ \bigwedge_{\mathcal{A}(\sigma,\phi) \neq \emptyset, \phi \in \mathcal{F}} [\, :\phi \approx :\phi]]\}.$$

Then for each theory $\theta$ we form the theory $\theta^{e} = \theta \cup \theta_{\Sigma}$ which is semantically equivalent to the original theory (because we add only the information from the signature which is always taken into account, when a theory is interpreted). We convert the theory $\theta^{e}$ into an exclusive matrix which in turn is converted into a set of graphs $\mathcal{GR}$ called **graph representation** of $\theta$.

The graph representation of a theory inherits from the exclusive matrixes their properties: (1) each graph $\mathcal{G}$ in $\mathcal{GR}$ is satisfiable (for some interpretation $\mathcal{I}$, $\mathcal{R}_{\mathcal{I}}(\mathcal{G}) \neq \emptyset$), and (2) each two graphs $\mathcal{G}_1$, $\mathcal{G}_2$ in $\mathcal{GR}$ have disjoint denotations (for each interpretation $\mathcal{I}$, $\mathcal{R}_{\mathcal{I}}(\mathcal{G}_1) \cap \mathcal{R}_{\mathcal{I}}(\mathcal{G}_2) = \emptyset$). We can reformulate here also the Prop. 2.

**Proposition 4** *Let $\theta$ be a finite SRL theory with respect to a finite signature, $\mu$ be the corresponding exclusive matrix, $\mathcal{GR}$ be the graph representation of $\theta$ and $\mathcal{I} = \langle \mathcal{U}_{\mathcal{I}}, \mathcal{S}_{\mathcal{I}}, \mathcal{F}_{\mathcal{I}} \rangle$ be a model of $\theta$. For each object $\upsilon \in \mathcal{U}$ there exists a unique graph $\mathcal{G} \in \mathcal{GR}$ such that $\upsilon \in \mathcal{R}(\mathcal{G})$.*

There exists also a correspondence between complete graphs with respect to a finite signature and the objects in an interpretation of the signature.

**Definition 5 (Object Graph)** *Let $\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$ be a finite signature, $\mathcal{I} = \langle \mathcal{U}_{\mathcal{I}}, \mathcal{S}_{\mathcal{I}}, \mathcal{F}_{\mathcal{I}} \rangle$ be an interpretation of $\Sigma$ and $\upsilon$ be an object in $\mathcal{U}$, then the graph $\mathcal{G}_{\upsilon} = \langle \mathcal{N}, \mathcal{V}, \rho, \mathcal{S} \rangle$, where*

$\mathcal{N} = \{\upsilon' \in \mathcal{U} \mid \exists \tau \in \mathcal{TM}$ *and* $\mathcal{P}(\tau)(\upsilon) = \upsilon'\}$

$\mathcal{V} : \mathcal{N} \times \mathcal{F} \to \mathcal{N}$ *is a partial* **arc function**, *such that*

$\mathcal{V}\langle \upsilon_1, \phi \rangle \downarrow$ *and* $\mathcal{V}_{\langle}\upsilon_1, \phi \rangle = \upsilon_2$ *iff*

$\upsilon_1 \in \mathcal{N}, \upsilon_2 \in \mathcal{N}$, $\phi \in \mathcal{F}$, *and* $\mathcal{F}_{\mathcal{I}}(\phi)(\upsilon_1) = \upsilon_2$,

$\rho = \upsilon$ *is the root node, and*

$\mathcal{S} : \mathcal{N} \to \mathcal{S}$ *is a* **species assignment function**, *such that*

$\mathcal{S}\langle \upsilon' \rangle = \mathcal{S}_{\mathcal{I}}\langle \upsilon' \rangle$,

*is called* **object graph**.

It is trivial to check that each object graph is a complete feature graph. Also, one easy can see the connection between the graphs in the graph representation of a theory and object graphs of objects in a model of the theory.

**Proposition 6** *Let $\theta$ be a finite SRL theory with respect to a finite signature, $\mathcal{GR}$ be the graph representation of $\theta$, $\mathcal{I} = \langle \mathcal{U}_\mathcal{I}, \mathcal{S}_\mathcal{I}, \mathcal{F}_\mathcal{I} \rangle$ be a model of $\theta$, $\upsilon$ be an object in $\mathcal{U}_\mathcal{I}$, and $\mathcal{G}_\upsilon = \langle \mathcal{N}, \mathcal{V}, \rho, \mathcal{S} \rangle$ be its object graph. For each node $\nu \in \mathcal{N}$, there exists a graph $\mathcal{G}_i \in \mathcal{GR}$, such that $\mathcal{G}\,|_\nu \sqsubseteq \mathcal{G}_i$.*

This can be proved by using the definition of a model of a theory, the Prop. 4 and the definition of a subgraph started at a node.

### 3.6. Outcomes: Feature Graphs for HPSG Grammar and Corpus

Thus we can sum up that feature graphs can be used for both:

- Representation of an HPSG corpus. Each sentence in the corpus is represented as a complete feature graph. One can easily establish a correspondence between the objects in an exhaustive model of (King 1999) and complete feature graphs or a correspondence between the elements of strong generative capacity of (Pollard 1999) and complete feature graphs. Thus complete feature graphs are a good representation for an HPSG corpus;

- Representation of an HPSG grammar as a set of feature graphs. The construction of a graph representation of a finite theory demonstrates that using feature graphs as grammar representation does not impose any restrictions over the class of possible finite grammars in SRL. Therefore we can use feature graphs as a representation of the grammar used during the construction of an HPSG corpus, as described above.

Additionally, we can use the properties of feature graphs in order to establish a formal connection between a grammar and a corpus.

**Definition 7 (Corpus Grammar)** *Let $C$ be an HPSG corpus and $\Gamma$ be an HPSG grammar. We say that $\Gamma$ is a* **grammar of the corpus** *$C$ if and only if for each graph $\mathcal{G}_C$ in $C$ and each node $\nu \in \mathcal{G}_C$ there is a graph $\mathcal{G}_G$ in $G$ such that $\mathcal{G}_C\,|_\nu \sqsubseteq \mathcal{G}_G$.*

It follows by the definition that if $C$ is an HPSG corpus and $\Gamma$ is a corpus grammar of $C$ then $\Gamma$ accepts all analyses in $C$.

## 4. Incremental Specialization using Negative Information

Let us now return to the annotation process. We start with an HPSG grammar which together with the signature determines the annotation scheme. We convert this grammar into a graph representation $\mathcal{GR}_0$. In the project we rely on the existing system (TRALE) for processing of HPSG grammars (TRALE is based on (Götz and Meurers, 1997)). TRALE works with HPSG grammars represented as general descriptions, but the result from the sentence processing is equivalent to a complete feature graph. It is also relatively easy to convert the grammar into a set of feature graphs.

Having $\mathcal{GR}_0$ we can analyze partial analyses of the sentences as it was described in the introduction. The partial analyses are used in order to reduce the number of the possible analyses. Let us suppose that the set of complete feature graphs $\mathcal{GRA}$ is returned by the TRALE system. Then these graphs are processed by the annotator within the CLaRK system and some of the analyses are accepted to be true for the sentence. Thus, they are added to the corpus and the rest of the analyses are rejected. Let $\mathcal{GRN}$ be the set of rejected analyses and $\mathcal{GRC}$ be the set of all analyses in the corpus up to now plus the new accepted ones. Our goal now is to specialize the initial grammar $\mathcal{GR}_0$ into a grammar $\mathcal{GR}_1$ such that it is still a grammar of the corpus $\mathcal{GRC}$ and it does not derive any of the graphs in $\mathcal{GRN}$. Using Prop. 6 we can rely on a very simple test for acceptance or rejection of a complete graph by the grammar: "If for each node in a complete graph there exists a graph in the grammar that subsumes the subgraph started at the same node, then the complete graph is accepted by the grammar." So, in order to reject a graph $\mathcal{G}$ in $\mathcal{GRN}$ it is enough to find a node $\nu$ in $\mathcal{G}$ such that for the subgraph $\mathcal{G}\,|_\nu$ there is no graph $\mathcal{G}' \in \mathcal{GR}_1$ such that $\mathcal{G}\,|_\nu \sqsubseteq \mathcal{G}'$. We will use this dependency in the process of guiding the specialization of the initial grammar.

In order to apply this test we have to consider not only the graphs in $\mathcal{GRC}$ and $\mathcal{GRN}$, but also their complete subgraphs. We process further the graphs in $\mathcal{GRN}$ and $\mathcal{GRC}$ in order to determine which information encoded in these graphs is crucial for the rejection of the graphs in $\mathcal{GRN}$. Let $sub(\mathcal{GRN})$ be the set of the complete graphs in $\mathcal{GRN}$ and their complete subgraphs and let $sub(\mathcal{GRC})$ be the set of the complete graphs in $\mathcal{GRC}$ and their complete subgraphs. We divide the set $sub(\mathcal{GRN})$ into two sets: $\mathcal{GRN}^+$ and $\mathcal{GRN}^-$, where $\mathcal{GRN}^+ = sub(\mathcal{GRN}) \cap sub(\mathcal{GRC})$ contains all graphs that are equivalent to some graph as well in $\mathcal{GRP}$[3] and $\mathcal{GRN}^- = sub(\mathcal{GRN}) \setminus sub(\mathcal{GRC})$ contains subgraphs that are presented only in $sub(\mathcal{GRN})$.

Then we choose all graphs $\mathcal{G}$ in $\mathcal{GR}_0$ such that for some $\mathcal{G}' \in \mathcal{GRN}^-$ it holds $\mathcal{G}' \sqsubseteq \mathcal{G}$. Let this set be $\mathcal{GR}_0^-$. This is the set of graphs in the grammar $\mathcal{GR}_0$ which we have to modify in order to achieve our goal.

Then we select from $sub(\mathcal{GRC})$ all graphs such that they are subsumed by some graph from $\mathcal{GR}_0^-$. Let this set be $\mathcal{GRP}$. These are the graphs that might be rejected by the modified grammar. Thus, the algorithm has to disallow such a rejection.

Thus our task is to specialize the graphs in the set $\mathcal{GR}_0^-$ in such a way that the new grammar (after substitution of $\mathcal{GR}_0^-$ with the new set of more specific graph into $\mathcal{GR}_0$) accepts all graphs in $\mathcal{GRP}$ and rejects all graphs in $\mathcal{GRN}$.

The algorithm works by performing the following steps:

[3]This is based on the fact that the accepted analyses can share some subgraphs with the rejected analyses.

1. It calculates the set $\mathcal{GRN}^-$;
2. It selects a subset $\mathcal{GR}_0^-$ of $\mathcal{GR}_0$;
3. It calculates the set $\mathcal{GRP}$;
4. It tries to calculate a new set of graphs $\mathcal{GR}_1^-$ such that each graph $\mathcal{G}$ in the new set $\mathcal{GR}_1^-$ is either member of $\mathcal{GR}_0^-$ or it is subsumed by a graph in $\mathcal{GR}_0^-$. Each new graph in $\mathcal{GR}_1^-$ can not have more nodes than the nodes in the biggest graph in the sets $\mathcal{GRP}$ and $\mathcal{GRN}$. This condition ensures the algorithm termination. If the algorithm succeeds to calculate a new set $\mathcal{GR}_1^-$ then it proceeds with the next step. Otherwise it stops without producing a specialization of the initial grammar.
5. It checks whether each graph in $\mathcal{GRP}$ is subsumed by a graph in $\mathcal{GR}_1^-$. If 'yes' then it prolongs the execution with the next step. Otherwise it returns to step 4 and calculates a new set $\mathcal{GR}_1^-$.
6. It checks whether there is a graph in $\mathcal{GRN}$ such that it is subsumed by a graph in $\mathcal{GR}_1^-$ and all its complete subgraphs in $\mathcal{GRN}^-$ are subsumed by a graph in $\mathcal{GR}_1^-$. If 'yes' then it returns to step 4 and calculates a new set $\mathcal{GR}_1^-$. Otherwise it returns the set $\mathcal{GR}_1^-$ as a specialization of the grammar $\mathcal{GR}_0^-$.

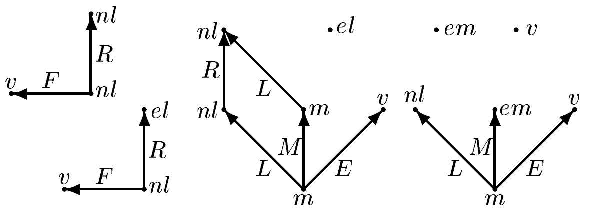
When the algorithm returns a new set of graphs $\mathcal{GR}_1^-$ which is a specialization of the graph set $\mathcal{GR}_0^-$, then we substitute the graph set $\mathcal{GR}_0^-$ with $\mathcal{GR}_1^-$ in the grammar $\mathcal{GR}_0$ and the result is a new, more specific grammar $\mathcal{GR}_1$ such that it accepts all graphs in the corpus $\mathcal{GRC}$ and rejects all graphs in $\mathcal{GRN}$.

In general, of course, there exist more than one specialization. Deciding which one is a good one becomes a problem, which cannot be solved only on the base of the graphs in the two sets $\mathcal{GRP}$ and $\mathcal{GRN}$. In this case two repairing strategies are possible: either additional definition of criteria for choosing the best extension, or the application of some statistical evaluations.

If the algorithm fails to produce a new set of graphs $\mathcal{GR}_1^-$ then there is an inconsistency in the acceptance of the graphs in $\mathcal{GRC}$ and/or in the rejection of the graphs in $\mathcal{GRN}$. This could happen if the annotator marks as wrong an analysis (or a part of it) which was marked as true for some previous analysis.

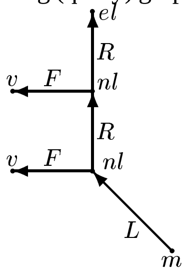
## 5. Example

In this section we present an example. This example is based on the notion of list and member relation encoded as feature graphs. The lists are encoded by two species: $nl$ for non-empty lists and $el$ for empty lists. Two features are defined for non-empty lists: $F$ for the first element of the list and $R$ for the rest of the list. The elements of a list are of species $v$. The member relation is also encoded by two species: $m$ for the recursive step of the relation and $em$ for the non-recursive step. For the recursive step of the relation (species $m$) three features are defined: $L$ pointing to the list, $E$ for the element which is a member of the list and $M$ for the next step in the recursion of the relation. The next set of graphs constitutes an incomplete grammar for member relation on lists. The incompleteness results from the fact that there is no restriction on the feature $E$.



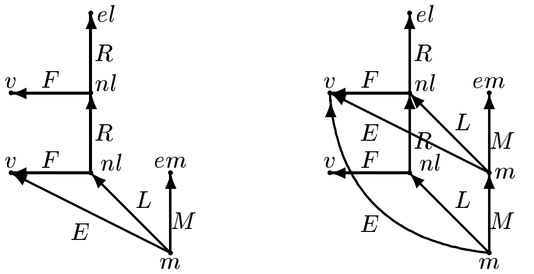
Here the two graphs on the left represent the fact that the rest of a non-empty list could be a non-empty list or an empty list. They also state that each non-empty list has a value. Then there are two graphs for the species $m$. The first states that the relation member can have a recursive step as a value for the feature $M$ if and only if the list of the second recursive step is the rest of the list of the first recursive step. The second graph just completes the appropriateness for the species $m$ saying that the value of the feature $L$ is also of species non-empty list when the value of the feature $M$ is non-recursive step of the member relation. There are also three graphs with single nodes for the case of empty lists, non-recursive steps of member relations and for the values of the lists. They are presented at the top right part of the picture. Now let us suppose that the annotator would like to enumerate all members of a two-element list by evaluation of the following (query) graph with respect to the above grammar.

**Query graph:**



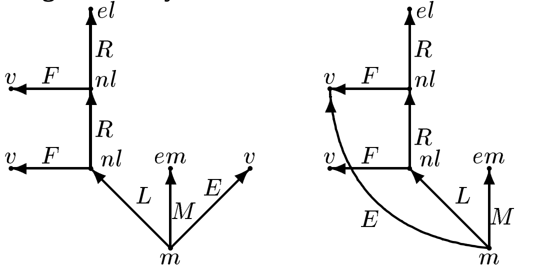
The grammar returns two acceptable analyses. One for the first element of the list and one for the second element of the list.

**Positive analyses:**

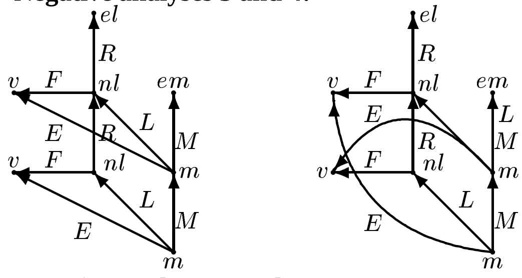


The grammar also accepts 11 wrong analyses in which the $E$ features either point to wrong elements of the list or they are not connected with element of the list at all. Here are the wrong analyses.

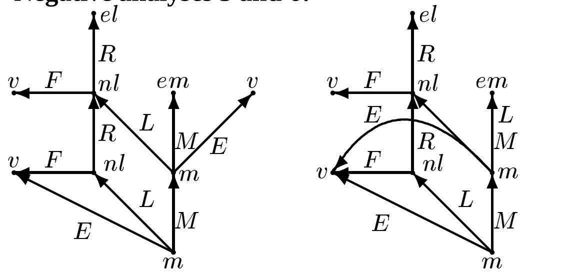
**Negative analyses 1 and 2:**



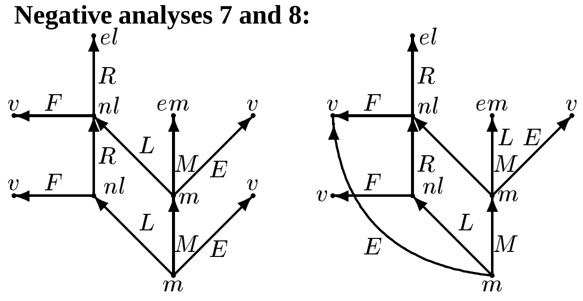
**Negative analyses 3 and 4:**



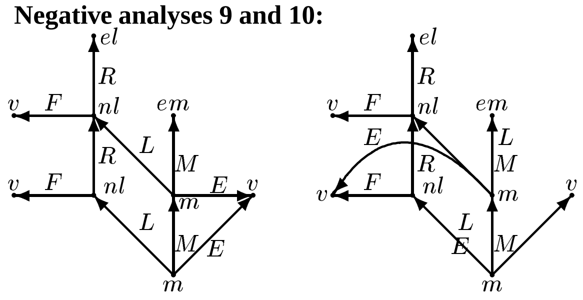
**Negative analyses 5 and 6:**



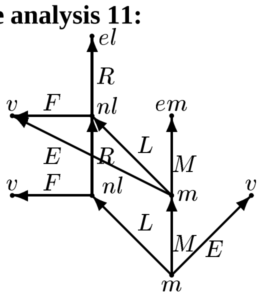
**Negative analyses 7 and 8:**



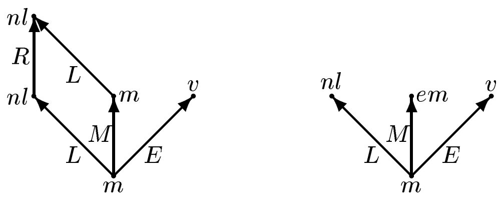
**Negative analyses 9 and 10:**



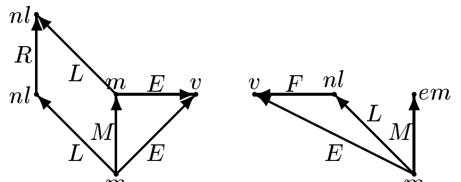
**Negative analysis 11:**



The next step is to determine the set $\mathcal{GRN}^-$. This set contains 12 complete graphs: all graphs in the set $\mathcal{GRN}$ and one subgraph that is not used in the positive analyses. We will not list these graphs here. The graphs from the grammar that subsumes the graphs in $\mathcal{GRN}^-$ are the two graphs for the member relation. We repeat them here.



Now we have to make them more specific in order to reject the negative examples from $\mathcal{GRN}^-$ but still to accept the two positive examples. The next two graphs are an example of such more specific graphs.



By the first graph the negative examples 3, 4, 5, 7, 8, 10 and 11 are rejected, and by the second graph the negative examples 1, 2, 5, 6, 7, 8, 9, 10 are rejected. Thus both specializations are necessary in order to reject all negative examples. The new grammar still accepts the two positive examples.

## 6. Conclusion

The presented approach is still very general. It defines a declarative way to improve an annotation HPSG grammar represented as a set of feature graphs. At the moment we have implemented only partially the connection between TRALE system and CLaRK system. Thus, a demonstration of the practical feasibility of the approach remains for future work.

Similar approach can be established on the base of the positive information only (see (Simov, 2001) and (Simov, 2002)), but the use of the negative information can speed up the algorithm. Also, the negative as well as positive information can be used in creation of a performance model for the new grammar along the lines of (Bod, 1998).

## 7. Acknowledgements

The work reported here is done within the BulTreeBank project. The project is funded by the Volkswagen Stiftung, Federal Republic of Germany under the Programme *"Co-operation with Natural and Engineering Scientists in Central and Eastern Europe"* contract I/76 887.

We would like to thank Petya Osenova for her comments on earlier versions of the paper. All errors remain ours, of course.

## 8. References

Rens Bod. 1998. *Beyond Grammar: An Experience-Based Theory of Language*. CSLI Publications, CSLI, California, USA.

Bob Carpenter. 1992. *The Logic of Typed Feature Structures*. Cambridge Tracts in Theoretical Computer Science 32. Cambridge University Press.

T. Götz and D. Meurers. 1997. *The ConTroll system as large grammar development platform*. In *Proceedings of the ACL/EACL post-conference workshop on Computational Environments for Grammar Development and Linguistic Engineering*. Madrid, Spain.

P.J. King. 1989. *A Logical Formalism for Head-Driven Phrase Structure Grammar*. Doctoral thesis, Manchester University, Manchester, England.

P.J. King. 1999. *Towards Thruth in Head-Driven Phrase Structure Grammar*. In V. Kordoni (Ed.), *Tübingen Studies in HPSG,* Number 132 in Arbeitspapiere des SFB 340, pp 301-352. Germany.

P. King and K. Simov. 1998. The automatic deduction of classificatory systems from linguistic theories. In *Grammars*, volume 1, number 2, pages 103-153. Kluwer Academic Publishers, The Netherlands.

P. King, K. Simov and B. Aldag. 1999. The complexity of modelability in finite and computable signatures of a constraint logic for head-driven phrase structure grammar. In *The Journal of Logic, Language and Information*, volume 8, number 1, pages 83-110. Kluwer Academic Publishers, The Netherlands.

C.J. Pollard and I.A. Sag. 1987. *Information-Based Syntax and Semantics*, vol. 1. CSLI Lecture Notes 13. CSLI, Stanford, California, USA.

C.J. Pollard and I.A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, Illinois, USA.

C.J. Pollard. 1999. *Strong Generative Capacity in HPSG.* in Webelhuth, G., Koenig, J.-P., and Kathol, A., editors, *Lexical and Constructional Aspect of Linguistic Explanation,* pp 281-297. CSLI, Stanford, California, USA.

K. Simov. 2001. *Grammar Extraction from an HPSG Corpus.* In: Proc. of the RANLP 2001 Conference, Tzigov chark, Bulgaria, 5–7 Sept., pp. 285–287.

K. Simov, G. Popova, P. Osenova. 2001. *HPSG-based syntactic treebank of Bulgarian (BulTreeBank).* In: *"A Rainbow of Corpora: Corpus Linguistics and the Languages of the World",* edited by Andrew Wilson, Paul Rayson, and Tony McEnery; Lincom-Europa, Munich, pp. 135–142.

K. Simov, Z. Peev, M. Kouylekov, A. Simov, M. Dimitrov, A. Kiryakov. 2001. *CLaRK - an XML-based System for Corpora Development.* In: Proc. of the Corpus Linguistics 2001 Conference, pages: 558-560.

K. Simov. 2002. *Grammar Extraction and Refinement from an HPSG Corpus.* In: Proc. of ESSLLI-2002 Workshop on Machine Learning Approaches in Computational Linguistics, August 5-9.(to appear)

K.Simov, P.Osenova, M.Slavcheva, S.Kolkovska, E.Balabanova, D.Doikoff, K.Ivanova, A.Simov, M.Kouylekov. 2002. *Building a Linguistically Interpreted Corpus of Bulgarian: the BulTreeBank.* In: Proceedings from the LREC conference, Canary Islands, Spain.