Grammar Extraction and Refinement from an HPSG Corpus*

Kiril 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

May 15, 2002

Abstract

Grammar learning and refinement on the basis of language resources is very appealing in comparison with manual development of formal grammar. But in order to learn a complex grammar a complex resource is needed. Thus the creation of language resources and learning of grammars from them have to be aware of each other. In this paper we define a formal basis for annotation of corpora with respect to a contemporary linguistic theory — HPSG and a methodology for extraction of grammars from such corpora. Also we describe an approach to incremental refinement of an HPSG grammar in the process of annotation of an HPSG Corpus.

1 Introduction

In recent years most of the language resources have been developed on rigorous linguistic and formal background. Such resources have become a challenge not only during the process of their development but also when they are used for different NLP tasks. In our view these two areas of research and technology have to be more aware of each other in order to have a greater mutual benefit. In our project we aim at the creation of a 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. In our annotation architecture we envisage two sources of linguistic knowledge in order to minimize the possible analyses of the annotated sentences:

*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 *"Cooperation with Natural and Engineering Scientists in Central and Eastern Europe"* contract I/76 887.

- 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 an 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: (1) their intersection is calculated; (2) on the base of the differences, a set of constraints over the intersection is calculated; (3) during the actual annotation step, the annatotator 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 next choices.

This architecture is being currently implemented by establishing an interface between two systems: one for XML based corpora development [Simov et. al. 2001] and one for HPSG grammar development [Götz and Meurers 1997]. The result of the project will be an HPSG corpus based on feature graphs and reliable grammars. One of the intended applications of these language resources is using the corpus for improving the accuracy of the implemented HPSG grammar.

In the annotation process, described above, the HPSG grammar serves as a constraint over the possible output syntactic structures. However, it will substantially overgenerate. Thus the most important contribution of the HPSG grammar is that the chosen analyses always meet certain pre-determined requirements. It is annotator's duty to mark the acceptable syntactic structures out of a large number of possible analyses. In this way all the interesting and more detailed linguistic facts, which lack in the general HPSG grammar, will be presented in the treebank on the basis of annotators' local decisions. Another important point in our work is that the analyses have a formal interpretation. This formal interpretation ensures the detection of some annotators' errors and further exploration of the treebank. For example, one could write logical theories over the representation of the treebank and then check whether they are satisfied by the structures in the treebank or not.

This paper describes an approach for extraction and incremental fine-graining of an HPSG grammar from such an HPSG corpus. For this specific task we rely on Rens Bod's Data-Oriented Parsing Model.

In a number of works Rens Bod defines a methodology for grammar learning from corpora. The main details of this methodology can be found in [Bod 1998]. He presents a grammar learning model from corpora which comprises the following elements:

1. Defining the grammar formalism for the target grammar;
2. Establishing a procedure for the construction of sentence analyses in the chosen grammar formalism;
3. Establishing a decomposition procedure, which extracts a grammar in the target grammar formalism from the structures in the corpus;
4. A performance model guiding the analysis of new sentences with respect to some desirable conditions.

Two additional unspoken assumptions are:

5. The structures in the corpus are decomposable into the grammar formalism;
6. The extracted grammar should neither overgenerate, nor undergenerate with respect to the training corpus.

The last assumption refers to the quality of the 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 its all acceptable syntactic structures. Thus a good grammar will not assign more analyses to the sentences than the analyses, which already exist in the corpus.

We report here only the formal basis of our work. A corpus of Bulgarian within this formal framework is under development. Thus, the practical feasibility of the approach has to be proved in future.

The structure of the paper is as follows: in the next section we define the notion of an HPSG corpus. Section 3 presents a logical formalism for HPSG, a definition of feature graphs based on this formalism and how each finite HPSG grammar can be represented as a set of feature graphs. Section 4 concentrates on the extraction of feature graphs from an HPSG corpus and on the construction of an HPSG grammar on the base of the extracted graphs. Section 5 discusses the refinement of HPSG grammars using the mechanism described in the previous section. The last section outlines the conclusions and future work. Note that in this paper we do not aim at defining a performance model for the extracted grammar.

2 The HPSG Corpus

First we start with a definition of a corpus in general. Such an ideal corpus has to ensure the above requirements and assumptions. A corpus C in a given grammatical formalism G is a sequence of syntactic structures of sentences where each structure is a member of the set of structures defined as a **strong generative capacity** (SGC) of a grammar Γ in this grammatical formalism:

Definition 1 (Grammar Formalism Corpus) *A corpus C in a 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 Γ in this grammatical formalism:*

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

where Γ 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 Γ to the phonological string $\sigma(S)$, then

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

Of course, the grammar Γ is unknown, but implicitly represented in the corpus C. The existence of such a grammar is a prerequisite for the above methodology to get to work. If the described grammar does not exist, then we will not be able to extract an appropriate grammar from the corpus — either the grammar will overgenerate over the corpus, or it will undergenerate.

We use this general definition in order to define a corpus in HPSG. Thus 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 corpus in HPSG containing sentences that are members of $\mathtt{SGC}(\Gamma)$ for some grammar Γ in SRL.

It is accepted that an HPSG grammar in SRL formally comprises two parts: signature and 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. The extraction of a signature from a corpus is a trivial task. In our view, however, it is very ambitious to aim at extracting an HPSG grammar as a set of implications directly from a corpus which is presented as a set of feature structures.

Hence we offer an intermediate format for the extracted grammar. We define a normal form for HPSG grammars. It is ideologically very close to the feature structures, which define the strong generative capacity in HPSG as it is proposed in the work of [King 1999] and [Pollard 1999]. We define both the corpus and the extracted grammar in terms of clauses (considered as graphs) in a special kind of matrices in SRL. We will present an extraction mechanism for grammars in this normal form. The construction of the 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 between the elements in normal form and the numerous 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 only. The possibility of them being semantic entities, i.e. elements of an interpretation, is excluded.

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 its representation as a set of feature graphs is shown. The main idea of the section is that the feature graphs as presented here (with subsumption relation based on isomorphism) can be used for a representation of grammars and corpora in HPSG. We define operations over these graphs (unification and etc.) which become the basis for an inference procedure. The procedure is logically correct with respect to the semantics of the logical formalism in question. Thus working only with feature graphs does not affect negatively neither the generality of the grammars that could be represented, nor the interpretation of these grammars. The only cost is the human readability, which is a serious loss, but we consider it as a second task to be solved within the process of grammar extraction.

3.1 King's Logic — SRL

This section presents the basic notions of Speciate Re-entrancy Logic (SRL) — for details see [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 $\mathcal{A} : \mathcal{S} \times \mathcal{F} \to Pow(\mathcal{S})$ 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 Σ (or Σ-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 $\upsilon \in \mathcal{U}_\mathcal{I}$,

if $\mathcal{F}_\mathcal{I}(\phi)(\upsilon)\downarrow$[1] then $\mathcal{S}_\mathcal{I}(\mathcal{F}_\mathcal{I}(\phi)(\upsilon)) \in \mathcal{A}(\mathcal{S}_\mathcal{I}(\upsilon), \phi)$, and

for each $\phi \in \mathcal{F}$ and each $\upsilon \in \mathcal{U}_\mathcal{I}$,

if $\mathcal{A}(\mathcal{S}_\mathcal{I}(\upsilon), \phi)$ is not empty then $\mathcal{F}_\mathcal{I}(\phi)(\upsilon)\downarrow$,

$\mathcal{F}_\mathcal{I}$ is called **feature interpretation function**.

τ is a term iff τ is a member of the smallest set $\mathcal{T}$ such that (1) $: \in \mathcal{T}$, and (2) for each $\phi \in \mathcal{F}$ and each $\tau \in \mathcal{T}$, $\tau\phi \in \mathcal{T}$. For each Σ-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 $\tau \in \mathcal{T}$, $\mathcal{P}_\mathcal{I}(\tau\phi)$ is the composition of the partial functions $\mathcal{P}_\mathcal{I}(\tau)$ and $\mathcal{F}_\mathcal{I}(\phi)$ if they are defined.

δ is a **description** iff δ is a member of the smallest set $\mathcal{D}$ such that (1) for each $\sigma \in \mathcal{S}$ and for each $\tau \in \mathcal{T}$, $\tau \sim \sigma \in \mathcal{D}$, (2) for each $\tau_1 \in \mathcal{T}$ and $\tau_2 \in \mathcal{T}$, $\tau_1 \approx \tau_2 \in \mathcal{D}$

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

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 Σ-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

$\mathcal{D}_\mathcal{I}(\tau \sim \sigma) = \{\upsilon \in \mathcal{U}_\mathcal{I} \mid \mathcal{P}_\mathcal{I}(\tau)(\upsilon)\downarrow, \mathcal{S}_\mathcal{I}(\mathcal{P}_\mathcal{I}(\tau)(\upsilon)) = \sigma\}$,

$\mathcal{D}_\mathcal{I}(\tau_1 \approx \tau_2) = \{\upsilon \in \mathcal{U}_\mathcal{I} \mid \mathcal{P}_\mathcal{I}(\tau_1)(\upsilon)\downarrow, \mathcal{P}_\mathcal{I}(\tau_2)(\upsilon)\downarrow, \text{and } \mathcal{P}_\mathcal{I}(\tau_1)(\upsilon) = \mathcal{P}_\mathcal{I}(\tau_2)(\upsilon)\}$,

$\mathcal{D}_\mathcal{I}(\tau_1 \not\approx \tau_2) = \{\upsilon \in \mathcal{U}_\mathcal{I} \mid \mathcal{P}_\mathcal{I}(\tau_1)(\upsilon)\downarrow, \mathcal{P}_\mathcal{I}(\tau_2)(\upsilon)\downarrow, \text{and } \mathcal{P}_\mathcal{I}(\tau_1)(\upsilon) \neq \mathcal{P}_\mathcal{I}(\tau_2)(\upsilon)\}$,

$\mathcal{D}_\mathcal{I}(\neg\delta) = \mathcal{U}_\mathcal{I} \setminus \mathcal{D}_\mathcal{I}(\delta)$,

$\mathcal{D}_\mathcal{I}([\delta_1 \wedge \delta_2]) = \mathcal{D}_\mathcal{I}(\delta_1) \cap \mathcal{D}_\mathcal{I}(\delta_2)$,

$\mathcal{D}_\mathcal{I}([\delta_1 \vee \delta_2]) = \mathcal{D}_\mathcal{I}(\delta_1) \cup \mathcal{D}_\mathcal{I}(\delta_2)$, and

$\mathcal{D}_\mathcal{I}([\delta_1 \rightarrow \delta_2]) = (\mathcal{U}_\mathcal{I} \setminus \mathcal{D}_\mathcal{I}(\delta_1)) \cup \mathcal{D}_\mathcal{I}(\delta_2)$.

Each subset $\theta \subseteq \mathcal{D}$ is an **SRL theory**. For each Σ-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 θ is **satisfiable** iff for some interpretation $\mathcal{I}$, $\mathcal{T}_\mathcal{I}(\theta) \neq \emptyset$. A theory θ is **modelable** iff for some interpretation $\mathcal{I}$, $\mathcal{T}_\mathcal{I}(\theta) = \mathcal{U}_\mathcal{I}$, $\mathcal{I}$ is called a **model** of θ. The interpretation $\mathcal{I}$ **exhaustively models** θ iff

$\mathcal{I}$ is a model of θ, and

for each $\theta' \subseteq \mathcal{D}$,

if for some model $\mathcal{I}'$ of θ,

$\mathcal{T}_{\mathcal{I}'}(\theta') \neq \emptyset$,

then $\mathcal{T}_\mathcal{I}(\theta') \neq \emptyset$.

An HPSG grammar $\Gamma = \langle \Sigma, \theta \rangle$ in SRL consists of: (1) a signature Σ which gives the ontology of entities that exist in the universe and the appropriateness conditions on them, and (2) a theory θ which gives the restrictions upon these entities. As it was mentioned above, the exhaustive models are basis for the definition of strong generative capacity in HPSG ([King 1999] and [Pollard 1999]).

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

(E0) if $\lambda \in \alpha$ then λ 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 $\{\tau, \tau_1, \tau_2, \tau_3\} \subseteq \mathcal{T}$, and `Term` is a function from the powerset of the sets of literals to the powerset of $\mathcal{T}$ such that

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

Two properties of an exclusive matrix $\mu = \{\alpha_1, \ldots, \alpha_n\}$ are important: (1) each clause α in μ is satisfiable (for some interpretation $\mathcal{I}$, $\mathcal{T}_{\mathcal{I}}(\alpha) \neq \emptyset$), and (2) each two clauses α_1, α_2 in μ have disjoint denotations (for each interpretation $\mathcal{I}$, $\mathcal{T}_{\mathcal{I}}(\alpha_1) \cap \mathcal{T}_{\mathcal{I}}(\alpha_1) = \emptyset$). [King and Simov 1998] shows that each finite theory with respect to a finite signature can be converted into an exclusive matrix which is semantically equivalent to the theory. Using the above properties it is easy to prove the following proposition.

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

3.3 Feature Graphs

Recall that a corpus in HPSG is supposed to 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). As it was mentioned above, in order to be distinguished among other numerous kinds of feature structures, they are called feature graphs.

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**,

ρ is a **root node**,

$\mathcal{S} : \mathcal{N} \to \mathcal{S}$ 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 Σ.

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**.

Feature graphs as defined here are a kind of feature structures which we treat syntactically rather than semantically. We use complete feature graphs for representation of 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.

A path π is a finite sequence of features $\pi \in \mathcal{F}^*$. ϵ is the empty path.

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

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

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

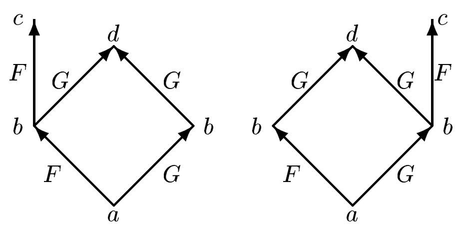
for each $\nu, \nu' \in \mathcal{N}_1$ and each feature ϕ,

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

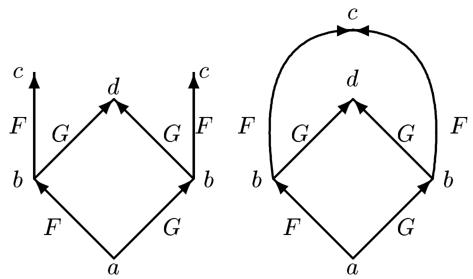
for each $\nu \in \mathcal{N}_1$, $\mathcal{S}_1\langle \nu \rangle = \mathcal{S}_2\langle \gamma(\nu) \rangle$.

The graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{V}, \rho, \mathcal{S} \rangle$ is a **unifier** of $\mathcal{G}_1$ and $\mathcal{G}_2$ iff $\mathcal{G} \sqsubseteq \mathcal{G}_1$ and $\mathcal{G} \sqsubseteq \mathcal{G}_2$. $\mathcal{G}_{mgu}$ is **a most general unifier** of $\mathcal{G}_1$ and $\mathcal{G}_2$ iff $\mathcal{G}_{mgu}$ is their unifier and for each another their unifier $\mathcal{G}'$, $\mathcal{G}_{mgu} \sqsubseteq \mathcal{G}'$ iff $\mathcal{G}' \sqsubseteq \mathcal{G}_{mgu}$. There is no unique most general unifier of two graphs even up to node renaming. $\mathtt{MGU}\langle \mathcal{G}_1, \mathcal{G}_2 \rangle$ is the set of all most general unifiers for the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ modulo equivalence of graphs, i.e. for each two graphs $\mathcal{G}'$ and $\mathcal{G}''$ in $\mathtt{MGU}\langle \mathcal{G}_1, \mathcal{G}_2 \rangle$ it holds that $\mathcal{G}' \not\sqsubseteq \mathcal{G}''$ and $\mathcal{G}'' \not\sqsubseteq \mathcal{G}'$. If $\mathtt{MGU}\langle \mathcal{G}_1, \mathcal{G}_2 \rangle = \emptyset$ we say that $\mathcal{G}_1$ and $\mathcal{G}_2$ are incompatible.

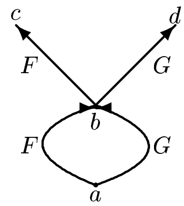
The intuition behind the definition of subsumption by isomorphism is that each graph describes "exactly" a chunk in some SRL interpretation in such a way that every two distinct nodes are always mapped to distinct objects in the interpretation. For instance, let us consider the following two graphs:



Their can be unified to the following two graphs:



But they can not unify to the following graph, which is possible only if the subsumption was defined by homomorphism instead of isomorphism:



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 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 $\alpha_{\mathcal{G}}$:

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

We interpret a graph by 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 graph then we interpret it as the intersection of the interpretations of all finite 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 μ can be represented as feature graphs. Let μ 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,

$\mathcal{V}_\alpha : \mathcal{N}_\alpha \times \mathcal{F} \rightarrow \mathcal{N}_\alpha$ 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 $\tau_1\phi \approx \tau_2 \in \alpha$,

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

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

3.4 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 θ 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 θ^e into an exclusive matrix which in turn is converted into a set of graphs $\mathcal{GR}$ called **graph representation** of θ.

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 θ be a finite SRL theory with respect to a finite signature, μ be the corresponding exclusive matrix, $\mathcal{GR}$ be the graph representation of θ and $\mathcal{I} = \langle \mathcal{U}_\mathcal{I}, \mathcal{S}_\mathcal{I}, \mathcal{F}_\mathcal{I} \rangle$ be a model of θ. 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 Σ and υ be an object in $\mathcal{U}$, then the graph $\mathcal{G}_\upsilon = \langle \mathcal{N}, \mathcal{V}, \rho, \mathcal{S} \rangle$, where*

$\mathcal{N} = \{\upsilon' \in \mathcal{U}_\mathcal{I} \mid \exists \tau \in \mathcal{T}$ *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. In order to prove this one could rely on the definition of interpretation in SRL and on the fact that each object in an SRL interpretation is "complete" with respect to the requirements of the signature. The connection between the graphs in the graph representation of a theory and the object graphs of objects in a model of the theory is also easily detectable.

Proposition 6 *Let θ be a finite SRL theory with respect to a finite signature, $\mathcal{GR}$ be the graph representation of θ, $\mathcal{I} = \langle \mathcal{U}_{\mathcal{I}}, \mathcal{S}_{\mathcal{I}}, \mathcal{F}_{\mathcal{I}} \rangle$ be a model of θ, υ 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.

3.5 Inference with Feature Graphs

In this paper we do not present a concrete inference mechanism using feature graphs. Of course one can use the general inference mechanisms of SRL in order to construct analyses of new sentences. However, a much better solution is to have an inference mechanism, which uses the graph representation of a theory directly. Such inference mechanism can be defined along the lines of *Breadth-First Parallel Resolution* of [Carpener 1992] despite the difference in the treatment of the feature structure in [Carpener 1992] ([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-liked algorithm. A presentation of such an algorithm goes beyond the scope of this paper.

3.6 Feature Graphs for HPSG Grammar and Corpus

Thus we can sum up that feature graphs are applicable 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 establish a formal connection between a grammar and a corpus exploiting the properties of feature graphs.

Definition 7 (Corpus Grammar) *Let C be an HPSG corpus and Γ be an HPSG grammar with graph representation $\mathcal{GR}$. We say that Γ 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 $\mathcal{G}_\Gamma$ in $\mathcal{GR}$ such that $\mathcal{G}_C\,|_\nu \sqsubseteq \mathcal{G}_\Gamma$.*

It follows from the definition that if C is an HPSG corpus and Γ is a corpus grammar of C then Γ accepts all analyses in C.

4 Extracting Grammars

In this section we present a fragmentation of an HPSG corpus into a set of feature graphs such that its subsets with particular properties can comprise HPSG grammars. In his work ([Bod 1998]) Rens Bod uses the whole set of fragments as a grammar with respect to other grammar formalisms. In more recent work ([Bod 2000], [Bod 2000]) he studies the impact of different subsets of fragments on the properties of the extracted grammar in order to minimise the redundancy in the grammar. We present both possibilities — how to construct the whole set of fragments and how to extract a particular grammar from this set.

Now we define the set of fragments extracted from the corpus C. We construct a set $\mathcal{GRF}$ of feature graphs such that

1. For each graph $\mathcal{G} \in \mathcal{GRF}$, $\mathcal{V}\langle\rho,\phi\rangle\downarrow$ iff $\mathcal{A}\langle\mathcal{S}\langle\rho\rangle,\phi\rangle\downarrow$, and
2. For each graph $\mathcal{G} \in \mathcal{GRF}$, there is a graph $\mathcal{G}_C$ in C and there is a node $\nu \in \mathcal{G}_C$ such that $\mathcal{G}_C \mid \nu \sqsubseteq \mathcal{G}$.

The first condition ensures that all features appropriate for a given species will be presented at the root node for each feature graph in $\mathcal{GRF}$ whose root is labelled by this feature. The second condition ensures that each feature graph in $\mathcal{GRF}$ is really a fragment of at least one feature graph in the corpus. As there are no more constraints over the feature graphs in $\mathcal{GRF}$ then $\mathcal{GRF}$ contains feature graphs with different size (not just the root and paths with length one). $\mathcal{GRF}$ also includes all complete graphs from the corpus.

We ordered $\mathcal{GRF}$ according to the subsumption relation over feature graphs. The set ordered in this way is a set of partial orders over the features graphs in $\mathcal{GRF}$. The minimal graphs in these partial orders are exactly the graphs from the corpus. The maximal graphs are very close to the signature.

Let G be a set of feature graphs such that for each minimal feature graph M in $\mathcal{GRF}$ there is at least one feature graph in G that subsumes M and G contains only feature graphs from $\mathcal{GRF}$. Again using the graph representation of a grammar and Prop. 4, it is easy to prove that a grammar G constructed in this way is a grammar of the corpus C.

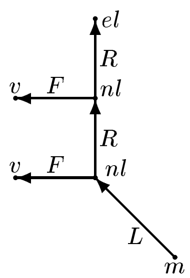
Choosing different feature graphs in $\mathcal{GRF}$ we can construct different grammars. We can compare two grammars on the base of their strong generative capacity. Let G_i and G_j be two grammars, then we say that the grammar G_i is more general than the grammar G_j iff $\mathtt{SGC}(G_j) \subseteq \mathtt{SGC}(G_i)$. In this case we call G_i a **generalization** of G_j and we call G_j a **specialization** of G_i. Determining this very relation between grammars is decidable in SRL. Thus we can order the possible grammars extracted from the corpus C by means of $\mathcal{GRF}$ in a hierarchy.

A grammar from this hierarchy can be chosen by specifying additional constraints over it. For instance, using the corpus C we can check whether a grammar overgenerates or undergenerates and thus we can choose the more general grammar with such properties. We might accept that this grammar generalises in best way over the corpus. Of course, in general there will be more than one such grammar. But which grammar is the best, constructed on the base of the corpus, depends on some external requirements. Such requirements are, for instance, the following conditions: (1) extracted grammar ensures the shortest inferences for the sentences in the corpus; (2) or it contains the most frequent graphs extracted from the corpus.

Example:

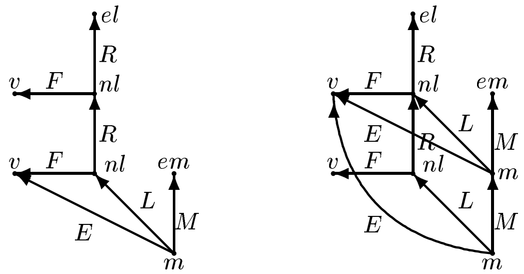
Here we present a very simple example. This example is based on the notion of list and member relation encoded as feature graphs. The lists are encoded by two species: nl for non-empty lists and el for empty lists. Two features are defined for non-empty lists: F for the first element of the list and 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. We accept that analysis in this 'grammar' is a graph in which one of the elements of the list is picked up by the feature E from all recursion steps of the member relation. Thus we could imagine that a typical query for analysis is a graph like the following:

Query graph:



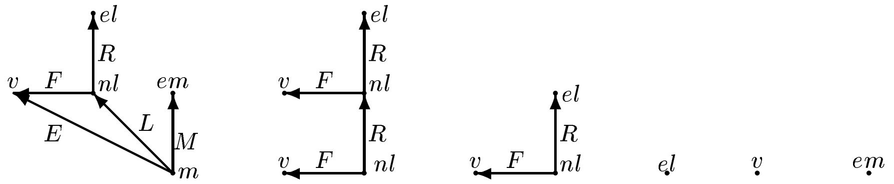
A possible corpus in this setup might contain the following two graphs: one for the first element of a two-element list and one for the second element of the same list. Therefore we could consider the query graph ambiguous with respect to the member relation and this fact is contained in the corpus.

Corpus:

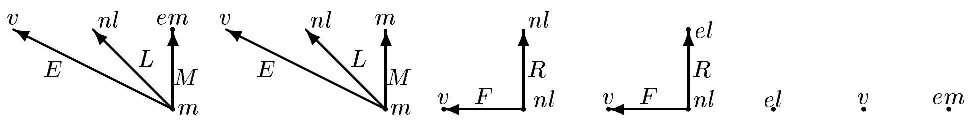


The number of fragments extracted from these two graphs are as follows: 19 from the left one and 113 from the right one (some of them are the same in the two graphs). We list here only the complete graphs that are different from the two graphs in the corpus and the most general fragments extracted from the corpus. Then we give a grammar of the corpus which does not undergenerate or overgenerate over the corpus.

Complete graphs:

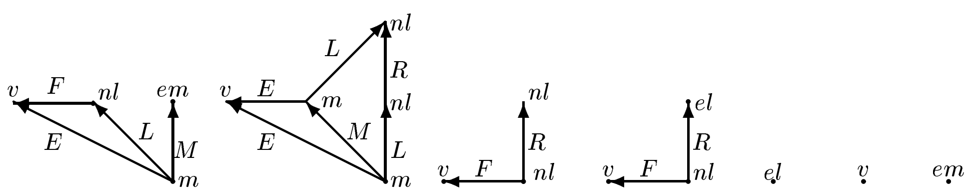


The most general fragments:



From these seven graphs the last four are at the same time complete and the most general. Therefore they belong to each grammar of the corpus. The first three graphs are the graphs which we can specialize in order to create different corpus grammars. If we take them as a part of a grammar then the result grammar will overgenerate over the corpus because there is no constraint over the behavior of the feature E and it could point to a node that is not connected with the elements of the list. Thus we have to take their more specific graphs in order to extract a grammar which does not overgenerate over the corpus.

A grammar of the corpus:



In this grammar the first two graphs are chosen to be more specific in comparison with the most general graphs above. The first graph ensures that the last recursive step is pointing to the first element of the associated list. The second graph ensures two things: first, the element of one recursive step of the relation member is the same as the element of the next recursive step and if there is a next recursive step, then it is connected with a non-empty list.

End of the example.

An alternative to the grammar construction, described above, is to use the whole set $\mathcal{GRF}$ as a grammar. It was mentioned above that this approach is taken by Rens Bod in his work ([Bod 1998]) with respect to other grammar formalisms. However, this alternative has some positive and some negative sides. First at all, the set $\mathcal{GRF}$ represents all generations over the corpus, but at the same time it will overgenerate over the corpus if one uses graphs from the top parts of the partial orders over $\mathcal{GRF}$. Thus, in order not to allow too general analyses, we have to design a special inference mechanism which ensures that possibly most specific feature graphs are actually used in the inference. The presence of very general graphs in the grammar can be used for analyses of ungrammatical sentences with some indications of the degree of the ungrammaticality. Another problem is the size of $\mathcal{GRF}$. As we can see in the example above, from a corpus of two graphs one can get more that a hundred of fragments. Thus, viewing the whole set $\mathcal{GRF}$ as a grammar is very attractive but it has to be approached carefully.

The general exploration of this topic is left for future work, but in the next section we discuss the possibility of using the initial HPSG grammar for the construction of the corpus in order to minimise the set of graphs which are considered during the analysis of the new sentences.

5 Grammar Refinement

As it was stated in the introduction, the development of an HPSG corpus is based on the usage of several grammars. In fact, we cannot imagine a completely manual annotation of a reasonable-size corpus, where a detailed annotation is aimed. Thus we could assume that together with the corpus we have also at our disposal the initial HPSG grammar. We envisage to use this grammar as a restriction over the process of extracting the graph fragments from the corpus.

Before describing the refinement process, two notions have to be introduced. We call the grammar G **exclusive grammar** if the matrix corresponding to the graphs in G is an exclusive matrix, otherwise the grammar is **non-exclusive grammar**.

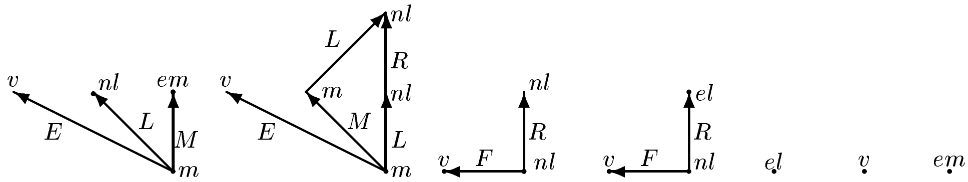
First, we construct a graph representation of the initial HPSG grammar. Let G_0 be the set of the graphs in this representation. G_0 is a grammar of the corpus because the graphs in G_0 are used during the annotation of the corpus and thus they subsume all the complete graphs presented in the corpus. In general G_0 will overgenerate over the corpus.

The grammar G_0 is an exclusive grammar because it is constructed by the algorithm for producing exclusive matrixes. As a **refinement** we consider each grammar G' such that G' is a specialization of G_0 and G' is an exclusive grammar. The specialization of a grammar is defined above on the basis of the strong generative capacity. Recall that there is no guarantee for the existence of exactly one most general refinement of a grammar. Thus our task now is to find the most general refinement(s) of G_0 which do not overgenerate over the corpus.

Example

Here we reconsider the example from the previous section. The following set of graphs constitutes an incomplete grammar for a member relation on lists. Let us suppose that it was used for the creation of the corpus given above. The missing information in this grammar is about the behavior of the feature E.

Initial grammar for member relation.



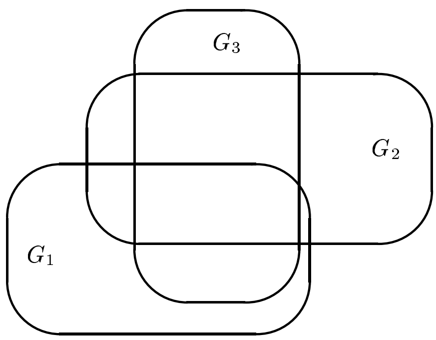
It is easy to see that the grammar given on page 14 is one of the most general refinements over this initial grammar.

End of the example.

One can easily construct a naive algorithm for enumeration of all refinements of G_0 and which returns the most specific ones that do not overgenerate or undergenerate over the corpus. In this paper we do not discuss such an algorithm.

Let $R = \{G_1, \ldots, G_n\}$ be the set of all refinements of the initial grammar G_0 . After the construction of R there are different ways to proceed. First, we can choose one of the grammars on the base of some other criteria, as it was mentioned above.

Another possibility is again using all the graphs in these grammars as a grammar - $G_r = \bigcup_{G_i \in R} G_i$. This grammar will be non-exclusive and in the general case it will overgenerate over the corpus. If we would like to save this property, we will have to index the graphs in G_r by the grammars they belong to and then in the inference we will use only graphs that belong to the same grammar. In general if such condition is not imposed over the use of G_r the result could be no refinement of G_0 at all. This is so because there is a possibility that G_r might contain all the graphs from G_0 and thus it will be equivalent to G_0. For instance, if $R = \{G_1, G_2, G_3\}$, we can present the three different grammars graphically as follows:



Although we can not ensure a unique refinement of the initial grammar, we can use the set R as a means for reducing the number of graph fragments that we have to extract and store. Also such a set could be a good point for a linguist to search for a refinement of G_0.

There are also other possibilities we cannot present here. They include such sources of additional information as the partial grammars, the rejected analyses during the construction of the corpus, which could serve as negative information (for details see [Simov et al., 2002]).

6 Acknowledgements

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

7 Conclusion and Future Work

In the paper we defined the notion of a corpus in HPSG. We outlined a common representation of such kind of corpus and of the grammars, that can be extracted from it. Then we presented an approach for grammar extraction from such a corpus. We also presented a mechanism for refinement of the initial HPSG grammar (used during the annotation of the corpus).

There are still open questions. If one would like to use the whole set $\mathcal{GR}$ as a grammar, then a procedure for construction of sentence analyses would be necessary. Also we have to define a performance model for the extracted grammar from such a corpus. Another problem is the conversion of the extracted grammar in more human oriented representation like implications in SRL. These will be the subject of our future work.

References

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

[Bod 2000] Rens Bod. *Parsing with the Shortest Derivation.* In the Proceedings COLING'2000. http://arxiv.org/abs/cs.CL/0011040, 2000.

[Bod 2000] Rens Bod. *Do All Fragments Count?.* Technical Report COMP-11-12, http://arxiv.org/abs/cs.CL/0011040, 2000.

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

[Götz and Meurers 1997] Thilo Götz and W. Detmar Meurers. *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. 1997.

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

[King 1999] Paul J. King. *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. 1999.

[King and Simov 1998] Paul J. King and Kiril Iv. Simov. The automatic deduction of classificatory systems from linguistic theories. In *Grammars*, volume 1, number 2, pages 103-153. Kluwer Academic Publishers, The Netherlands. 1998.

[King, Simov and Aldag 1999] Paul J. King, Kiril Iv. Simov and Bjørn Aldag. 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. 1998.

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

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

[Pollard 1999] Carl Pollard. *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. 1999.

[Simov et. al. 2001] Kiril Simov, Zdravko Peev, Milen Kouylekov, Alexander Simov, Marin Dimitrov, Atanas Kiryakov. *CLaRK - an XML-based System for Corpora Development.* In: *Proc. of the Corpus Linguistics 2001 Conference*, pages: 558-560. 2001.

[Simov et al., 2002] K.Simov, M.Kouylekov, A.Simov. *Incremental Specialization of an HPSG-Based Annotation Scheme.* In: Proceedings of the Workshop on "Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Language Data", the LREC conference, Canary Islands, Spain. 2002.