Lexical Issues
in Machine Translation

Research Report

Research Support Scheme of the Open Society Institute
Final Report on Project No. 340/93
September 1993 – March 1996

Alexandr Rosen
Institute of Theoretical and Computational Linguistics
Faculty of Philosophy, Charles University, Prague

Contents

1 Introduction: Machine translation system as a linguistic testbed

The discipline of machine translation is notorious for its mixture of theoretical and engineering aspects. Although this fact is responsible for a number of pitfalls in the history of this field, it also makes research efforts attractive because of the possibility of testing theoretical hypotheses in a multidimensional environment.

Some of the factors which make machine translation system attractive for linguistic research are as follows:

- The bilingual or multilingual environment is a guard against the temptation to present language-internal facts as linguistic universals. At the same time, such an environment is a strong motivation for searching as much common ground as possible among different languages.
- Representation of meaning, extracted from the input, is testable by its translation into the target language. If the system is reversible, a more profound test can be performed by translating the result back into the source language.
- The contrastive framework of a machine translation system is also an attractive field for modelling another important aspect of language: the relation between surface word order and the hierarchy of communicative dynamism with the topic/focus articulation. The comparison of Czech and English, different in factors determining the surface word order, is especially fruitful.
- Most problems and subfields of natural language processing are present in machine translation, which is important for the development of compatible formalisms and processing tools.

This realization of the importance of machine translation for linguistic theory was at the heart of the present enterprise. Although some components, especially lexical data, have been acquired and structured with the aim of maintaining re-usability in different, possibly practically-oriented, environments, the major part of work has been directed towards the construction of a flexible partial model of linguistic competence in the two languages.

The model rests firmly in a lexical setting due to the choice of a unification-based formalism and the powerful hierarchical organisation of lexical data with defaults and multiple inheritance. Lexical specifications are responsible for the solution of many more or less idiosyncratic phenomena. As a result, the grammar component consists of a few rather general rules manipulating structurally and informationally rich lexical descriptions.

The conceptual model is rather easy to modify even in its very basic assumptions and as such it is open to further development, even with other tools. Re-usability of lexical descriptions and the possibility of reconstructing other specifications is an important factor, supporting the 'testbed' view.

The integrated partial description of Czech and English consists of the following items:

- a lexicon of English lemmas, acquired from electronic sources, with more than 100.000 entries
- an efficiently organized description of English lexical classes, including inflectional paradigms, valency requirements and the specification of a uniform data structure for lexical items
- a grammar of English, covering a number of non-trivial phenomena within a monostratal unification-based context-free framework, representing 'linguistic meaning' of surface strings as structures isomorphic with annotated dependency trees
- a lexicon of Czech lemmas with a representative sample of words of different inflectional types and syntactic properties; acquisition of significant amounts of lexical data is previewed from available electronic sources
- a description of Czech lexical classes, parallel to the English description, with a substantial morphological component consisting of hierarchically organized definitions of inflectional types
- a grammar of Czech employing the same formalism and processing tool as that for English, doing justice at the same time to some fundamental properties of Czech: variations of word order, fixed position of clitics, complexities of analytical formation of tenses, rich agreement patterns

The components have been tested in parsing hundreds of made-up sentences including (combinations of) relevant phenomena and also some real texts (for English). The Czech component has been thoroughly evaluated both for parsing and for generation.

The basic strategy was to verify the proposed approach vis à vis the formalism and the processing tools for English. The availability of rich lexical data resources was an important factor in this choice. Next, a parallel Czech model was built with basic functions common to both languages and solutions for major distinctive Czech phenomena evaluated. As a further step, solutions to some phenomena present in both languages (coordination, unbounded dependencies) will be imported with some modifications from English to Czech.

As a consequence of this approach, coverage of the Czech description is more restricted than that of English, especially in the lexical part. Regrettably, the original plan to build and fully integrate a substantial transfer component had to be postponed. It remains to be seen whether the transfer link would require modifications in some previous assumptions.

The following Sections 2 – 5 are concerned with aspects which the two language descriptions have in common. Starting with an overview of the shared aspects in Section 2, a brief characteristic of the formalism is provided in Section 3. Next, in Section 4, the lexical component is presented. Special attention is devoted to its internal organisation and the lexical specifications it defines. Section 5 deals with the grammar component, its foundations, types of grammar rules, grammar principles applied across multiple rules and rule types, a proposal concerning the hierarchy of communicative dynamism, and the role of heuristics. Finally, in Sections 6 and 7, issues specific to English and Czech are presented, including a schematic description of grammar rules.

Throughout the present report, references are provided to samples of formal descriptions or computer outputs in the Appendix.

2 Reversible formal descriptions of Czech and English in a lexically-based framework

The formal partial descriptions of English and Czech share a number of common characteristics:

1. Description of lexical items, morphological paradigms and syntactic properties of both languages is conceived as a formally uniform **declarative specification** expressing the relation between the text string and its representation. A declarative description can be interpreted as a set of constraints applicable to two different processes: analysis of a string yielding the string's representation(s) and synthesis (or generation) of one or more strings from a representation. A grammar is called reversible precisely when this dual view of a single description is possible. Declarativity further supports other desirable characteristics, which bring the linguistic and computational aspects closer together. A formal declarative description can be structured in a way which a linguist considers intuitively plausible and which, at the same time, enables easy maintenance and development. In other words, problem-oriented modularity is easy to achieve. From the theoretical point of view, the most interesting feature of such an approach is the possibility to express linguistic generalizations of various types in a straightforward manner and to test them both for compatibility with the rest of the theory and for adequacy with respect to the data.

2. The declarative description is based on **constraints**. A constraint (or feature) is represented as an attribute and its value. Constraints describe sets of linguistic objects of any kind, from the most specific objects like an instance of a lexical form to more general objects like a word class category. Together with grammar rules, constraints describe properties of syntactic structures of certain types and general principles which these structures obey. A set of admissible linguistic objects is usually described by many constraints combined through **unification** into a coherent non-redundant specification. Such a specification is called **feature structure**. Values of attributes can be atomic or complex; feature structures are the most common example of a complex value. It is possible and desirable to define certain constraints as being appropriate for a given type of linguistic objects, thereby excluding any other constraint from descriptions of this type of objects. Feature structures for which a type has been declared as appropriate are called **typed feature structures**.

   An instance of a syntactic structure can be described more or less exhaustively, i.e. with a different degree of specificity. The most exhaustive (specific) description is its representation, a set of constraints given by unification of the surface string with the relevant specifications in the lexicon and the grammar. The set of objects which is described by the representation has usually a single member. An example of much less specific (more general) description of the same object is a specification of the syntactic structure in grammar. Such a specification can describe a potentially infinite number of objects.

Feature structures representing linguistic objects can relate two levels of description within a single formal framework, possibly with the inclusion of additional intermediate levels. A relation between the two levels is often interpreted as the relation between form and meaning, or between *signifié* and *signifiant* according to de Saussure. Such (typed) feature structures can then be viewed as **linguistic signs**.

3. Lexicon and morphology are integrated into a single **hierarchy of lexical classes**. The most specific classes are the lowest in the hierarchy and correspond to lemmas, while the most general class at the top is characterized (constrained) by features shared by any lexical item. A lexical form is identified through the interaction of specifications of the lemma and of all its superordinate classes in the hierarchy. Lower classes recursively inherit specifications from higher classes, unless a lower class specifies the same feature differently from a higher class. In such a case, the more specific lower class overrides the more general specification of the more general higher class in this feature, i.e., overrides a default value of this feature.

   Furthermore, a class may inherit specifications from more than one immediately dominating classes. Such multiple 'paths' of inheritance can be employed for the specification of orthogonal or mutually compatible information (e.g., inflectional type and valency are considered to be orthogonal). However, the system also provides a way of solving the cases of incompatible inherited information. A familiar example of such a case is the so-called 'Nixon diamond' (Touretzky 1986): the US president Nixon is both a Quaker and a Republican. While typical Quakers are pacifists and typical Republicans are not, the question is whether Nixon is a pacifist.

   If the property of being or not being pacifist is represented as value of an attribute of the classes Quaker and Republican which dominate the class Nixon, the conflict in different values of the same attribute inherited from two immediately dominating classes is resolved by specifying a precedence order of those classes. If the order of Nixon's superclasses is specified as `(Quaker Republican)`, the Quaker class is taken as more specific, overriding in effect the 'default' Republican class: Nixon ends up as being pacifist. To make Nixon a non-pacifist, the order of superclasses should be reversed. Thus, the solution of multiple inheritance of conflicting information is actually based on a mechanism similar to that used for single default inheritance.

4. While the lexicon with integrated morphology describes the relation between a word form and its representation, grammar is responsible for the relation between the representation of word forms and the representation of a sentence. The grammar is based on **context-free rules** which license, together with lexical specifications, various language structures and describe the relation between surface string and its **dependency-based representation** within a (formally) monostratal standard unification model. The combination of a standard context-free unification-based formalism with a dependency-oriented account of 'linguistic meaning' has its merit in the possibility to employ efficient processing methods and available linguistically-oriented formal apparatus, while preserving a transparent, sophisticated and adequately expressive representation. The grammar shares most of the properties of unification-based approaches, especially the crucial role of lexicon, which assumes certain tasks traditionally considered as belonging to the domain of grammar.

The following schematic account of the relationship between linguistic theory, grammar, formalism and implementation should provide a picture of the foundations and overall design of monolingual descriptions.

theory T**:** FGD – Functional Generative Description for the representation of linguistic forms by means of dependency graph with nodes standing for content words (see Sgall et al., 1986); description of surface-level syntactic phenomena and of the relation between the string of word forms and the representation in the spirit of HPSG – Head-Driven Phrase Structure Grammar (see Pollard & Sag, 1994)

grammar and lexicon G**:** a (partial) description of English and Czech by means of formalism F within theory T

formalism (metalanguage) F**:** syntax of the metalanguage is based on context-free rules with path equations stipulating the unification of values in complex categories (*signs*), resembling the notation of PATR-II with numerous extensions like list operations, atomic disjunctions, typed feature structures, hierarchical lexicon with multiple-inheritance and defaults

implementation I**:** compiler of the grammar and the lexicon G specified by means of the formalism F, extended Earley-style parser (prediction through multiple restrictors), head-driven generator (based on the algorithm described in Shieber et al., 1989), dictionary look-up module, transfer component

3 Formalism

The whole description is expressed in the formalism ELU – Environnement linguistique d'unification (see Estival, 1990, Russell et al., 1992), which is designed for the implementation of experimental machine translation systems. However, each of the three basic components of a machine translation system — analysis, transfer and synthesis (or 'generation') — can be developed and run independently. The basic properties of the formalism are as follows:

data structures: typed feature structures, lists, atomic disjunctions, atomic negations; a dependency tree is approximated within the general format of typed feature structures by attributes and their (list) values for the governing node and its dependent(s)

lexicon: specifications of lemmas and lexical classes in a tangled hierarchy with multiple inheritance and defaults; the same lexicon can be used reversibly for analysis and synthesis

grammar: context-free rules with constraints as path equations, list operations; the same grammar can be used reversibly for analysis and synthesis

transfer: the lexical and structural rules of transfer are an extension of the unification formalism, preserving the properties of declarativity, bidirectionality and locality, with an integrated control strategy for the recursive application of the rules, which is based on two relations: the relations of subsumption between the source description and the corresponding description by the rules, and the relation of specificity which imposes partial ordering on the transfer rules

macros: definition of recurrent sets of *constraints*;

the macros can have parameters, multiple definitions, and can be recursive

Both monolingual components consist of five modules:

general specifications common for lexicon and grammar: declaration of variables used for dictionary look-up, declaration of restrictors and paths of special significance for generation, declaration of feature structure types (see Sample 1)

definition of macros used both in lexicon and grammar either to express general principles or more specific but recurrent constraints (see Sample 2)

definition of lexical classes — a repository of lexical data which are not specific to a lemma, structured according to their specificity in a hierarchy (see Sample 3)

dictionary of lemmas including information specific to individual lexical items: base form, word class, inflection type, valency type (see Sample 4)

grammar rules — each covering a number of structural types and resting heavily on general principles invoked by macro calls (see Sample 5)

4 Lexicon

In a unification-based formalism, lexicon is not only a repository of lexical knowledge, but also the source of formal structural descriptions providing the basis for descriptions of higher units. Richly structured descriptions of lexical items, as provided by the lexicon, are isomorphic with data structures used throughout the system. However, the specification of lexical items in the lexicon need not amount to describing all intricacies required by the grammar component for every lexical item. Lexical knowledge which is not specific to a single item or lexeme can be expressed at an appropriate level of generality and inherited by all relevant individuals. This also applies to the way inflectional forms are related to the lemma.

4.1 Hierarchical organisation of lexical data

The lexicon combines specification of lemmas with a description of inflection paradigms and provides a representation of word forms, isomorphic with the representation of higher syntactic structures[1] Some of its functions are equivalent to lexical rules.

The lexicon consists of an alphabetically ordered list of base forms with word-item-specific lexical data (lemmas, see Sample 4) and definitions of lexical classes, ordered in a hierarchical structure (see Sample 3). Within this structure, every lexical class can inherit specifications (constraints) from higher (more general, populous) classes, while lemmas are treated as the lowest, most specific classes. If an inherited constraint does not unify (i.e., if it conflicts) with a constraint of the heir, the heir's constraint overrides the inherited data. If the constraints unify, the resulting constraint assumes the information of the higher class by default.

A single class can be immediately dominated by more than one class. This is the case of *multiple inheritance.* If constraints of two immediately dominant classes A and B are in conflict, the precedence order of classes stated in the heir's definition is taken into account. A constraint of class A overrides a conflicting constraint of class B just in case A precedes B in the list of classes which are above the heir. The constraint of B is therefore treated as default[2]

Based on these assumptions, at compile time the algorithm computes for every class a total ordering of all dominating classes. In (1) an ordering of classes dominating a few lexemes in the Czech lexicon is given. The first non-indented line for every lexeme shows a lexical entry with its immediate superclasses.

[1]Cf. a schema of the *lexical* type in (3); a *phrasal* type differs in that it has an additional attribute DTRS.
[2]See page 4 for the 'Nixon diamond' example.

(1)

```
bratr (PN6 Noun)
  bratr PN6 PN MascAnim NounInfl Noun ContentWord Word
kost (KS Noun)
  kost KS Fem NounInfl Noun ContentWord Word
hluchy2 (MD Adj)
  hluchy2 MD MDbis AdjInfl Adj ContentWord Word
kve2st (Impf NE6 Verb)
  kve2st Impf Ng NE6 NE VerbInfl Verb GeneralVerb VerbFrame SubjectAgr ContentWord Word
moci (Ng PE3 ModalVerb)
  moci Ng PE3 VerbInfl ModalVerb GeneralVerb DefaultAuxVerbGov SubjectAgr FunctionWord Word
by2t (Ng BYT Verb)
  by2t Ng BYT VerbInfl Verb GeneralVerb VerbFrame SubjectAgr ContentWord Word
by2t (Ng BYT AuxVerb)
  by2t Ng BYT VerbInfl AuxVerb GeneralVerb DefaultAuxVerbGov SubjectAgr FunctionWord Word
z (Prep)
  z Prep FunctionWord Word
se (Noun)
  se Noun ContentWord Word
si (Noun)
  si Noun ContentWord Word
on (ON PersPron)
  on ON PronInfl PersPron Noun ContentWord Word
```

An interested reader is referred to Sample 3 in the Appendix, where definitions of some verb-related classes are presented. The classes `PN6`, `PN`, `KS`, `MD`, `NE6`, `NE`, `PE3`, `BYT` and `ON` denote inflectional (sub)types. The classes `MascAnim`, `Fem`, `Impf` (= imperfective), `ModalVerb`, `AuxVerb` and `PersPron` subdivide major parts of speech. The classes `ContentWord` and `FunctionWord` partition the most general class `Word`. The classes `VerbFrame` and `SubjectAgr` provide for adequate structuring of information on valency and on properties relevant for subject-predicate agreement. This information is accessible in variables instantiated either in lexical entries (valency) or by classes `NounInfl` and `VerbInfl` relating morphological categories to word forms (agreement). Other classes like `GeneralVerb` or `DefaultAuxVerbGov` have the purpose of denoting some general properties of a given category.

When a part of inherited constraints conflicts with the heir's constraints, the status of the relevant constraints of the higher class matters. The constraints can either be located in the so-called main set of constraints, or in one of the so-called variant constraint sets. A constraint in the main set which conflicts with a constraint from a lower class is overridden by the conflicting constraint from the lower class while other non-conflicting main set constraints from the higher class remain unchanged, i.e. are inherited by the lower class. However, if a constraint from a variant set of the higher class conflicts with a constraint from a lower class, a unification failure results with the effect that the whole relevant variant set of constraints cannot be inherited. If there is no other non-conflicting variant set, the constraints cannot be satisfied in any other way.

A simplified lexical specification in (2) should provide a picture of the approach[3].

(2)

```
# Class  Word ()
         <the most general specification of a lexical unit>

# Class  Verb (Word)
         <general specification of an intransitive verb>
         |
         Paradigm = nese
         <specification of the inflection paradigm ‘nese’>
         |
         Paradigm = bere
         <specification of the inflection paradigm ‘bere’>
         |
         etc.

# Class  Trans ()
         <general specification of a transitive verb>

# Word   lézt  (Verb)  Paradigm = nese
# Word   prát  (Trans Verb)  Paradigm = bere
```

The example defines three lexical classes and two lemmas. The class `Word` stands at the top of the hierarchy, dominating the class `Verb`. The lemmas `lézt` ‘to crawl’ and `prát` ‘to wash (laundry)’ stand below the class `Verb`. The lemma `prát` is dominated by another class `Trans`. Specification of a transitive verb in the class `Trans` overrides the default specification of an intransitive verb in the class `Verb`. The variable `Paradigm` is instantiated for `prát` with the value `bere`. This value selects the correct variant set of constraints describing the relevant inflexion paradigm in the class `Verb` (all other variant sets have a different value of the variable `Paradigm` and are excluded by unification failure).

The *multiple inheritance* property is typically used for specifying two orthogonal characteristics of lexical items: inflectional behaviour and valency requirements. The specifications inherited from two major dominating sets of classes, which correspond to the description of inflectional paradigms and valency patterns, are combined in a single description of a lexical form.

Before looking up a word form in the lexicon, the formal description has to be compiled into a list of all possible forms indexing a file of lexical information representing the form. This file can be saved onto external memory, which is a pre-requisite when the lexicon is to be used for generation. The whole lexical component has so far been tested on the maximum of 20.000 lemmas (for English).

4.2 The format of lexical data

After compilation, a lexical form can be related to its description provided by the lexical classes. In analysis, the input is a form and the output is one or more representations, in synthesis (generation) the input is a representation and the output is one or more forms.

The form of representation has a common structure both for English and Czech, which is illustrated below in (3).

[3] Here, the ordered list of superclasses for `lézt` is `lézt Verb Word`. For `prát` the list is `prát Trans Verb Word`

(3)
$$\begin{bmatrix} \text{ORTHO} & \langle \ldots \rangle \\ \text{CD} & \langle \ldots \rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \text{x} \\ \text{HEAD} & [\ldots]_{\textit{head}} \\ \text{SC} & \langle \ldots \rangle \\ \text{DEEP} & \begin{bmatrix} \text{GOV} & [\ldots]_{\textit{gov}} \\ \text{DEPS} & \langle \ldots \rangle \end{bmatrix}_{\textit{deep}} \end{bmatrix}_{\textit{local}} \\ \text{NONLOC} & \begin{bmatrix} \text{INHER} & \begin{bmatrix} \text{REL} & \langle \ldots \rangle \\ \text{SLASH} & \langle \ldots \rangle \end{bmatrix}_{\textit{inher}} \\ \text{TO-BIND} & \begin{bmatrix} \text{REL} & \langle \ldots \rangle \\ \text{SLASH} & \langle \ldots \rangle \end{bmatrix}_{\textit{to-bind}} \end{bmatrix}_{\textit{nonloc}} \end{bmatrix}_{\textit{synsem}} \end{bmatrix}_{\textit{lexical}}$$

At the top level of the structure, whose type is *lexical*, there are three attributes: ORTHO, CD and SYNSEM. Values of the former two attributes at the lexical level are singleton or empty lists, the item in the ORTHO list is the verb form, the item in the CD list is the base form, in case a content word is described, or the list is empty if it is a function word. In phrasal projections of the lexical category the lists grow with the number of non-head daughters: in the maximal projection of *has* as the syntactic head of the sentence *Hamish has been referring to you* (with *refer to* as its 'underlying' head) the two lists look as follows:

(4) ORTHO ⟨ Hamish has been referring to you ⟩
CD ⟨ Hamish refer_to you ⟩

The value of the SYNSEM attribute is a feature structure of the type *synsem*, in the jargon of typed feature structures a *synsem* object. The *synsem* object includes information which can be subcategorized-for, i.e. selected by a head as the required specification of a complement.

The *synsem* object consists of *local* and *nonloc*al structures. The former describes properties of the word or its projection, while the latter takes care of phenomena which go beyond the word's head domain, the so-called unbounded dependencies occurring in questions, relative clauses, constructions with adjectives of the *tough* type, and topicalized constructions.

The value of CAT is an atom and is used as the principle restrictor during parsing and generation. As such, it is a combination of the head word class and the bar level.

The value of HEAD is a feature structure, a *head* object, which consists of information passed by unification from the head daughter to all of its projections. The *head* object varies slightly according to word class and language.

The value of SC (subcategorization list) is a list of underspecified *synsem* objects, representing a complementation paradigm. The objects correspond to complements and the list must be satisfied in the sense that the corresponding complements must exist at some level of projection of the lexical head. The maximal projection of the lexical head must have an empty SC list — the higher the projection, the shorter its SC list. In English, the order in the list reflects the unmarked sequence of complements (including the subject, which is first). In Czech, the SC attribute is replaced by an attribute SUBJ for subject and COMPS for complements. Also, in Czech (a pro-drop language) the subject is not necessarily expressed and thus the SUBJECT attribute may not be empty at the maximal projection level.

So far, the structure is not much different from some standard approaches. The major departure comes with the attribute DEEP, whose value makes explicit reference to dependency-oriented accounts, approximating the theory of Functional Generative Description. The two objects, the values of GOV and DEPS, stand for the governing node and the dependents, respectively. The *deep* object is that part of the whole representation which is passed to the transfer module and from which a corresponding string in the target language is synthesized (generated).

Similarly as the *head* object, the value of GOV is different for different word classes. The example in (5) provides the GOV value for the English word form *she*, as specified by the lexicon.

(5)
$$\begin{bmatrix} \text{GOV} & \begin{bmatrix} \text{LEXEME} & \text{she} \\ \text{WCL} & \text{pers\_pron} \\ \text{INDEX} & \begin{bmatrix} \text{PERS} & 3 \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{fem} \end{bmatrix}_{index} \\ \text{PFORM} & \text{-nil-} \\ \text{DEFINITE} & \text{yes} \end{bmatrix}_{ngov} \\ \text{DEPS} & \langle\rangle \end{bmatrix}_{deep}$$

The *index* object groups together information which is important for expressing agreement and anaphoric relations. The PFORM attribute is a preliminary means for analysing prepositional marking. The value of DEPS is a list of objects typed as *dependent.* An underspecified representation of such an object is shown in (6), which is the DEEP value for the verb form *expires.*

(6)
$$\begin{bmatrix} \text{GOV} & \begin{bmatrix} \text{LEXEME} & \text{expires} \\ \text{WCL} & \text{verb} \\ \text{PFORM} & \text{-nil-} \\ \text{SCONJ} & \text{-nil-} \\ \text{VOICE} & \text{active} \\ \text{TENSE} & \text{present} \\ \text{MODALITY} & \text{-nil-} \\ \text{PROGRESSIVE} & \text{no} \\ \text{PERFECTIVE} & \text{no} \\ \text{POLARITY} & \text{positive} \end{bmatrix}_{ngov} \\ \text{DEPS} & \left\langle \begin{bmatrix} \text{FUN} & \text{actor} \\ \text{DIR} & \text{left} \\ \text{VAL} & \mid \text{GOV} \mid \text{INDEX} \begin{bmatrix} \text{PERS} & 3 \\ \text{NUM} & \text{sg} \\ \text{GEND} & \text{-} \end{bmatrix}_{index} \end{bmatrix}_{dependent} \right\rangle \end{bmatrix}_{deep}$$

The underspecified value of VAL is instantiated to a *deep* object of the verb's actor at its appropriate projection level, with the *index* object specified to take care of subject – predicate agreement.[4]

The proper *gov* object as the value for the attribute VAL is selected by sharing the *gov* values in *synsem* objects of the SC/SUBJ/COMPS lists with the corresponding VAL values.

[4]Passivization is treated by the lexicon — a different *lexical* object is produced for transitive verbs in the passive participle form.

The following example (7) shows relevant parts of the *lexical* description of the Czech verb form *pomohly* 'helped$_{pl,3rd,fem/masc_inanim}$' with two complements — a nominal and an infinitive.

(7)

$$\begin{bmatrix}
\mathrm{ORTHO} \left\langle \begin{bmatrix} \text{pomohly} \end{bmatrix} \right\rangle \\
\mathrm{CD} \left\langle \begin{bmatrix} \text{pomoci} \end{bmatrix} \right\rangle \\
\mathrm{SYNSEM} \mid \mathrm{LOC} \begin{bmatrix}
\mathrm{INFL} \begin{bmatrix} \mathrm{PERS} & 3 \\ \mathrm{NUM} & \text{pl} \\ \mathrm{GEND} & \text{f/mi} \\ \mathrm{VARIANT} & 1 \end{bmatrix}_{vinfl} \\
\mathrm{SUBJ} \left\langle \begin{bmatrix} \mathrm{LOC} \begin{bmatrix} \mathrm{CAT} & \text{n/nbar} \\ \mathrm{INFL} & \mid \mathrm{CASE}\ \text{nom} \\ \mathrm{COMPS} & \langle\rangle \\ \mathrm{SUBJ} & \langle\rangle \\ \mathrm{DEEP} & \boxed{1} \end{bmatrix} \end{bmatrix} \right\rangle \\
\mathrm{COMPS} \left\langle \begin{bmatrix} \mathrm{LOC} \begin{bmatrix} \mathrm{CAT}\ \text{n/nbar} \\ \mathrm{INFL} \mid \mathrm{CASE}\ \text{dat} \\ \mathrm{COMPS}\ \langle\rangle \\ \mathrm{SUBJ}\ \langle\rangle \\ \mathrm{DEEP}\ \boxed{2} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \mathrm{LOC} \begin{bmatrix} \mathrm{CAT}\ \text{v/vbar} \\ \mathrm{HEAD} \mid \mathrm{VFORM}\ \text{inf} \\ \mathrm{COMPS}\ \langle\rangle \\ \mathrm{SUBJ}\ \boxed{4} \\ \mathrm{DEEP}\ \boxed{3} \end{bmatrix} \end{bmatrix} \right\rangle \\
\mathrm{DEEP} \mid \mathrm{DEPS} \left\langle \begin{bmatrix} \mathrm{FUN} & \text{actor} \\ \mathrm{VAL} & \boxed{1} \end{bmatrix}, \begin{bmatrix} \mathrm{FUN} & \text{addressee} \\ \mathrm{VAL} & \boxed{2} \end{bmatrix}, \begin{bmatrix} \mathrm{FUN} & \text{patient} \\ \mathrm{VAL} & \boxed{3} \end{bmatrix} \right\rangle
\end{bmatrix}_{local}
\end{bmatrix}_{lexical}$$

$$\boxed{4} \left\langle \begin{bmatrix} \mathrm{LOC} \mid \mathrm{DEEP}\ \boxed{2} \end{bmatrix} \right\rangle$$

The numbers in boxes are indexes which stipulate sharing of values. Thus, the DEEP value $\boxed{1}$ for subject is shared with the actor's value for the attribute VAL. Similarly, the dative complement's DEEP value $\boxed{2}$ is instantiated as the addressee and the infinitive $\boxed{3}$ as the patient.

The dative object controls the subject of the infinitive, so the DEEP value $\boxed{2}$ of the dative object is shared with the DEEP value of the infinitive's subject (the value of the infinitive's SUBJECT attribute is shown separately ar $\boxed{4}$. Since the verb *pomoci* is of the *equi* type, the controlling complement occurs at the deep level twice: once as the addressee of the finite verb and once as a participant of the embedded verb (Actor, if the embedded infinitive is in the active voice, or Patient, if the voice is passive).

In Czech, indexical agreement categories are not present in the SUBJ value's *object*, because agreement is realized by a mechanism independent of satisfying subcategorization requirements.

The entire structure for *pomohly* in the format produced by ELU is given in the Appendix, Sample 6. For the English form *prevents*, see Sample 7.

5 Grammar

The grammar design is inspired by two linguistic traditions: (i) those constraint-based grammar theories which model the relation between the surface string and its representation by means of a monostratal description with a context-free backbone and the mechanism of unification, i.e. within a single formal apparatus (GPSG – Generalized Phrase Structure Grammar and HPSG – Head-driven Phrase Structure Grammar are examples of such a theory); (ii) the theory of FGD – Functional Generative Description with its richly structured representation of language expression.

This combination of context-free unification grammar and well-motivated dependency representation is desirable for at least two reasons:

1. Within monostratal theories and their constraint-based apparatus, a number of formal and easily implementable solutions exist for the description of many non-trivial syntactic phenomena. This applies especially to English, more recently to German, French, Japanese and a few other languages, but may be true, after some modification of the theory, also about Slavic languages.

2. FGD offers a theory of the underlying level of language expressions, where many aspects of 'linguistic meaning' are explicitly represented in a relatively transparent fashion. The aspects include topology and types of structural relations between expressions (represented as labelled dependency links between content words), properties of the individual expressions (like number, person, tense), degree of communicative dynamism and topic-focus articulation.

The grammar models a language by describing the relationship between the surface string of language expressions and their dependency-based representation via context-free rules annotated with constraints. The context-free rules and their dependency interpretation can be viewed as two relatively independent levels of linguistic description. This relative independence can be employed for the description of a number of language phenomena. The following English and Czech examples should illustrate some cases where this relative independence of DL ('deep level') and SL ('surface level') is used.

- In the DL dependency graph, function words are represented as features annotating relevant content words, rather than as independent nodes.

- Relations between heads and their modifiers (complements and adjuncts) are explicitly labelled by function names (Actor, Patient, Addressee, Locative, Time, ...).

- In a context-free rule, a daughter always has to be selected as the head of the construction. However, this 'SL' head need not correspond to a node governing the construction on DL. So, in a sentence *Hamish has been referring to you* (see 8), *has* is the SL head of the sentence, while the DL governor is the content verb *refer to* (see 9).[5] In the following, the notion of *head* will be reserved for heads on SL, while a governing

[5]The examples are simplified: only structural links are shown, nodes are labelled only by categorial symbol or lexeme. Another simplification is the omission of the topmost SL node which is labelled by `Sentence`. The complete DL representation in the form of feature structure is given in Sample 8.

node on DL will be called *governor*.

(8)

```
   ____S_____
  /          \
Hamish   ___AuxV___
        /          \
       has    ___AuxV____
             /           \
            been      __VP1__
                     /       \
                 referring  PP_
                           /   \
                          to you
```

(9)

```
          __refer~to__
         /            \
       Hamish         you
```

Similarly, in a Czech sentence *Nebudu moci pást husy* 'I won't be able to graze geese' (see 10), *nebudu* is the SL head of the sentence, while the DL governor is the content verb *pást* (see 11).[6]

(10)

```
       S
       |
   ___VC____
  /         \
nebudu   __VC___
        /       \
       moci    _VC_
              /    \
            pa2st husy
```

(11)

```
          ___pást___
         /          \
        já          husa
```

[6] Again, the complete DL representation is given in Sample 8. In formal specifications as well as in computer inputs and outputs, the Czech diacritics are transliterated as follows: á = a2, č = c3, ů = u4.

Auxiliary and modal verbs in English and Czech are treated as members of the *raising* verb class, together with verbs like *seem* or *expect* (in the usage *he seems to come* and *she expects him to come*). Unlike auxiliaries, the latter verbs have a node of their own on DL. The Czech past tense auxiliary *být* is an exception — it is treated as a *marker* as in (12): *Pást husy jsem nemohl* 'I could not graze geese'.

(12)

```
         S
         |
     ____CV____
    /          \
   _VC_      AuxV_
  /    \    /     \
pa2st husy jsem nemohl
```

- Due to the SL/DL distinction, non-projective constructions (discontinuous constituents, long-distance, crossing or unbounded dependencies) can be represented as regular trees (for SL see (13), for DL see (14) and Sample 9). On SL, slash mechanism is used to control the relation between a gap (a trace) and its filler, while on DL the filler depends projectively on its governor. Except for cases when this projectivity restriction could be violated, the left-to-right ordering of DL tree nodes reflects the scale of communicative dynamism of the sentence, which tends to correspond to the order of content words in the surface string (see Section 5.3 below).

(13)

```
    ___FH____
   /         \
Drogheda   __S___
          /      \
         she   _VP1__
              /      \
              is   _PP__
                  /     \
                 from Trace
                        |

      ________S________
     /                 \
   DetNP          ____AuxV____
  /     \        /            \
 the  fence    VPcle       ___AuxV___
              /     \     /          \
            could  not  have    ___AuxV____
                               /           \
                             been     _AdjComp1_
                                     /          \
                                   easy     __AuxV__
                                           /        \
                                          to      _VP1_
                                                 /     \
                                              refer   _PP_
                                                     /    \
                                                    to Trace
                                                          |
```

(14)

```
                ____is
               /    /
   from~Drogheda  she

               easy
                   \
                   refer~to
                   /
                 fence
```

- A single constituent can be shared (co-indexed) on DL by two or more nodes. The specification of such a constituent, or its relevant part, is referred to wherever necessary, without interfering with the tree structure of the representation. A typical example is an *equi* object-control type construction like *he persuaded me to stay* or *Honzovi jsme hnůj kydat nepomohli* 'We didn't help John to muck manure out' (see (15) and Sample 11), where *me* and *Honzovi* are interpreted both as inner participants (complements) of the finite verb and as the Actor of the infinitive (a PRO element in GB terms).

(15)

```
         S                                    __pomoci_______________
         |                                   /         \              \
    _____CV_____                           my     [1]Honza       __kydat__
   /            \                                                /         \
Honzovi    ____AuxV_____                                       [1]       hnůj
          /             \
        jsme        ____CV____
                   /          \
                _CV__    nepomohli
               /     \
             hnu4j kydat
```

A similar approach can be used for representing anaphoric relations (including those involving zero pronominals).

- Syntactic categories in context-free rules do not have to correspond to categories of DL governors (a deverbative nominal can be represented as a verb on DL).
- The number of non-head daughters is usually different from the number of dependents on the governor representing the head. Most rules are binary-branching while the number of dependents on a single governor on DL is determined by its valency[7] and the actually used number of optional free modifications (adjuncts).
- Although there is no explicit DL valency frame, a distinction can be made between SL complementation requirements and corresponding DL valency frames. On SL, obligatory complementations of various types (complements, adjuncts, particles) can be realized in several ways:
    1. as a SUBCAT list (in the English grammar) which specifies all (obligatory) complementations realized as non-heads; in the case of a verb the list typically includes a subject and (other) complements, but may also include specification of a particle; nouns typically specify requirements for a determiner;

[7] The term valency is used here to denote subcategorization requirements of a lexical item imposed on non-auxiliary elements. For a detailed discussion of the (Tesnière-inspired) approach to valency within the theory of FGD, see Sgall et al. 1986, 111ff. and Panevová

2. as SUBJECT and COMPS features (in the Czech grammar), a parallel to the SUBCAT list split into the feature specifying the subject and the COMPS feature specifying all other (non-head) items;
3. as a SPEC feature used in *markers* (e.g., prepositions, subordinating conjuncts) and adjuncts to specify possible heads.

The difference between the SL complementation requirements and a corresponding DL valency frame is (i) in the number of subcategorized-for items (DL lacks requirements on particles, raising words on DL lack a complement), and also (ii) in that auxiliary words (verbs, prepositions) have SUBCAT or SPEC features, but do not exist as nodes on DL.

5.1 Types of Grammar Rules

The grammar contains a relatively small number of rules of several types. There are a few general principles which apply to all rules and other principles apply to all rules of a certain type. Both rule types and principles are defined as macros which the individual rules invoke.

The grammars of both English and Czech share the following rule types:

head_only_rule:
Rules of this type license a single (head) daughter. Dependency interpretation: specification of a DL feature (a *grammateme*) or some other DL information when it is licensed due to the absence of some other lexical item(s)[8]

head_complement_rule:
Rules of this type license a head plus one or two obligatory complementations.[9] Dependency interpretation: specification of a structure consisting of a head and its obligatory modification(s), be it inner participants or free modifications.[10]

head_adjunct_rule:
Rules of this type license a head and an adjunct, i.e. an optional free modification. Dependency interpretation: specification of an optional free modification.

head_marker_rule:
Rules of this type license a head and a function word (preposition, subordinating conjunction, past tense auxiliary verb in Czech). Dependency interpretation: specification of a grammateme due to the presence of a function word.

aux_head_complement_rule:
Rules of this type license an auxiliary verb as the head and another item listed in the frame of the auxiliary, i.e. an appropriate form of a content verb or another auxiliary.

coord_rule:
A single rule of this type licenses the first member of a coordinated structure, a conjunctive element (conjunction or comma) and the second member. Coordination of more than two items is treated by repeated application of the same rule. The rule stipulates conditions for coordination of various categories by invoking macros which have the effect of generating a family of coordination rules from a single schematic rule. Interpretation in the dependency-based framework: specification of an entire coordinated structure (if it consists of two items), or of the rightmost two members of a coordinated structure, or of the third or next member from the right (if it consists of more than two items).

[8]An example is the treatment of bare plural in English, where the head noun is interpreted as non-definite. Similarly, a Czech clause with unexpressed subject assumes its clausal status while allowing for later additions to its underspecified subject description.

[9]The Czech grammar generates almost exclusively binary-branching structures. Accordingly, these rules allow only a single non-head daughter.

[10]In English, there is an exceptional rule of this type. The rule licenses a sequence 'article, nominal group' when the mother node is marked by an appropriate grammateme of determination. The head noun must subcategorize for a determiner.

trace_rule:
A single rule of this type licenses a gap (trace) when the gap is bound by a filler. Dependency interpretation: detection of a potential non-projective (crossing) dependency.[11]

head_filler_rule:
A single rule of this type licenses a head daughter which carries a specification of a gap to be filled by a filler daughter. Dependency interpretation: specification of an unprojectively dependent complement.

5.2 Principles of Grammar

In addition to the types of rules, the grammars of English and Czech share some general principles, which apply across a number of different rules.

HFP: Head Feature Principle
This principle unifies (provides token identity between) *head features* of the mother and *head features* of the head daughter. *Head features* are those constraints which do not change (i.e., which unify) their values in head daughters on the path from the lexical head up to its maximal projection.

HDeep: deep Head Feature Principle
This principle unifies the DL representation of the governor and its dependents as provided by the head daughter with the DL representation of the mother. This principle applies in head-complement rules. Head-adjunct rules use a modified version of the principle due to the unpredictability in the occurrence of adjuncts. Head-marker rules do not obey this principle at all since they modify the deep head features of the governor.

HDtrs: daughters principle
This principle has a few variants according to the rule type and stipulates token identity between the value of the attribute HEAD_DTR and the head of the phrase, and also between the values of the attributes COMP_DTR, ADJUNCT_DTR, MARKER_DTR, FILLER_DTR and COORD_DTR and the respective daughters.

SCP: Subcategorisation Principle
This principle stipulates a relation between the surface frame of the head daughter, its sister complement(s) and the surface frame of the mother. The frame of the mother equals the frame of the head daughter minus the item which unifies with the SYNSEM value of the complement(s). The surface frame is realized as a single subcategorization list for English (including the subject) and as a complements set[12] and a subject feature for Czech.

SpecP: specifier principle
This principle unifies the value of the attribute SPEC in the adjunct daughter or the marker daughter with the SYNSEM value of the head.

NlocFP: Non-local Feature Principle
This principle controls the distribution of the non-local features SLASH and REL among the mother and daughters in the rule.

The following table gives an overview of principles and their use in the rule types.

	h_o	h_c	aux	h_a	h_f	h_m	coor
HFP	+	+	+	+	+	+	+
HDeep	+	+	-	-	-	-	+
HDtrs	+	+	+	+	+	+	+
SCP	-	+	+	-	-	-	-
SpecP	-	+	+	-	-	+	-
NlocFP	+	+	+	+	-	-	+

[11]The dependency tree retains projectivity if an item is dislocated out of its constituent but remains within the domain of the dependency subtree whose root is its immediate governor, like in the sentence *Those funny people I like so much.*).
[12]The set is merely a data type simulated by list operations available in ELU. As a matter of fact, the value of the COMPS attribute is a list.

5.3 Surface and deep word order

In the theory of FGD, the nodes in the underlying dependency representation of a sentence are ordered from left to right according to the rising degree of communicative dynamism. According to a recent version of the theory, this ordering is total for sister nodes, i.e. for nodes immediately governed by a common head. This solution has the merit in that it preserves projectivity (non-crossing dependency links), but, on the other hand, cannot be used to represent distinctions between some sentences which are not synonymous in a way consistent with general assumptions.

The two Czech sentences in (16) and (17) illustrate this difficulty.

(16) Tu knížku Pavel slíbil vrátit do zítřka.
the book Pavel promised to return till tomorrow

(17) Pavel slíbil tu knížku vrátit do zítřka.
Pavel promised the book to return till tomorrow.

In both cases, the nodes of the tree are ordered in the same way, as in (18).

(18)

```
        ___slíbil______
       /               \
    Pavel            __vrátit__
                    /          \
              tu knížku     do zítřka
```

Even though the unexpressed (PRO) subject *Pavel* of the imbedded verb can be ordered either after or before the object *tu knížku* 'the book' to represent the distinction, the object of the embedded verb still has to follow the main verb in order to save the projectivity requirement. This does not reflect the intuition that in (16 the "embedded" object *tu knížku* 'the book' is less dynamic than the finite verb *slíbil* 'promised', while in (17) just the opposite is the case. The situation is similar in English examples like (19) — represented by the tree in (20), where the raised object of 'break' is less dynamic than the matrix clause predicate 'difficult'.

(19) The fence has been difficult to break.

(20)

```
difficult
     \
    break
    /
 fence
```

However, if the ordering of nodes should be defined over the whole sentence, including embedded structures, and if a member of an embedded structure like 'the book' or 'the fence' is treated as the least dynamic element of the whole clause, the tree is no longer projective. If the projectivity condition is to be retained and all elements of the sentence ordered, including elements of embedded structures, the degrees of communicative dynamism have to be reflected independently of the ordering of nodes in the dependency tree.

A proposal along these lines might be based on the introduction of a separately specified dimension of the DL representation, an attribute CD for *communicative dynamism*. The value of CD might be a list of references to items participating in the communicative dynamism hierarchy. In the current version of the system, the list consists of lexemes of content words in a sequence which should correspond to the communicative dynamism hierarchy.

However, this solution is not without problems, both formal and linguistic. If the question of how to relate a surface string with this list is put aside, the following issues remain:

- The proposal does not solve the problem of the partitioning of the list into topic and focus.

- According to its conceptual status, the CD list should be located inside the *deep* object. However, a CD list should not be subcategorized, i.e. should not be a part of a *synsem* object.

- Items from the CD list should point to GOV values. However, GOV values are not correct until a level of projection covering all function words and possibly some adjuncts as well. (GOV is not a head feature.) Therefore, the list points just to lexemes.

- Although in typical cases the scale of communicative dynamism and the topic/focus partitioning concerns content words, function words and even dependent morphemes (like inflectional endings) may also be involved, as in cases of verum focus *You haven't watered the plants, have you? — Of course I* **HAVE** *watered the plants.*[13] So far, this issue has also not been dealt with.

In spite of these problems, the proposal has been implemented as the first approximation of an alternative representation of the hierarchy of communicative dynamism.

5.4 Heuristics

Linguistic knowledge encoded in the lexicon and grammar is not sufficient to delimit the boundary between correct and incorrect usage and to constrain the relationships between forms and (representations of) their meanings. A substantial part of such knowledge is of semantic nature and very difficult to formalize. Furthermore, the whole formal description models the competence aspect of language and its use. Pragmatic considerations and other aspects of performance are beyond the model's descriptive capacity.

While extraction of data from large corpora and their use in combination with rule-based systems is previewed as a way of approximating human abilities in language production and understanding, the current model uses much less sophisticated techniques in cases of necessity.

As a substitute for a more flexible and profound approach, heuristics is used where necessary. Such cases typically occur whenever there is not enough information to constrain a linguistic object through a more general device. (To illustrate this fact, just consider that without heuristic constraints, a simple nominal group *the freedoms defined in Section I* has four different representations.) Heuristic constraints sometimes fail in that they are too restrictive in unpredictable contexts. In order to make debugging in such cases of 'overkill' easier, heuristic constraints are distinctly separated from regular constraints.

Some examples of heuristic constraints:

1. An optional adjunct is often at a position where its preceding or following head cannot be unambiguously determined. In such cases, the nearest neighbouring candidate is selected. (This is an analogy of the *right association* principle for English.)

2. An optional adjunct adjacently following or preceding a coordinated structure can modify either the whole coordination or only its adjacent member. In such cases, the whole coordination is treated as the head.

3. In English, if an *of*- or *by*-prepositional phrase follows the verb and some other constituent dependent on the verb, and if it does not satisfy a subcategorization requirement of the verb,[14] it is not treated as the verb's modifier.

6 Specifics of English

English has been and still is by far the language most often subjected to formal descriptions within various theoretical frameworks. This means that it is often possible to choose between various already existing solutions to many problems, especially of English syntax. Another important factor is the bias inherent in many theories, formalisms and software tools towards this specific language, especially towards grammaticalized word order, restricted amount of inflectional variations and, last but not least, the absence of diacritics. Finally, there is

[13] The cases of metalinguistic correction like *I said EXtension, not INtension* probably do not provide sufficient argument for structuring the communicative dynamism and topic/focus distinction at an even finer level than morphemes.

[14] The actor of a passive verb realized as a *by*-phrase is subcategorized by the passive participle.

a number freely available large electronic sources of lexical data, more or less suitable for natural language processing tasks.

In spite of this, the existing known solutions have to be reconstructed rather than simply imported as ready-made, and there are many phenomena, some of them rather frequent, which have for various reasons escaped the attention of theoretical linguistics. In the following, the framework presented in the preceding sections is instantiated for the description of English.

6.1 The English lexicon

The lexicon for English has been compiled from several electronic sources with a considerable degree of human involvement and semi-automatic processing. The core of lexical data has been extracted from an electronic version of OALD (Hornby 1974), described in Mitton (1986), and extended by acquisitions from other sources. The number of lemmas is currently slightly over 100.000. All of them have been transformed into the canonical format which conforms to the ELU formalism. 20.000 the entries have been tested and compiled and are available for parsing and generation. A detailed description of the format of entries is available as a part of the current report.

6.2 Grammar rules for English

This section makes reference to the previous general section on grammar rule types.

head_only_rule: Sentence, NP

Sentence: sentence $\rightarrow$ **s** punct[15]
EXAMPLE: $[_{sentence}$ $[_s$ Hamish has been referring to you$]$ $[_{punct}$.$]$ $]$

NP: np $\rightarrow$ **n/nbar/nbbar**
EXAMPLE: Hamish broke $[_{np}$ $[_{nbbar}$ two new fences $]$ $]$.

head_complement_rule: S, VInv, VP1, VP2, DetNP, AdjComp1, AdjComp2

S: s $\rightarrow$ np **vp**
EXAMPLE: $[_s$ $[_{np}$ Hamish$]$ $[_{vp}$ has been referring to you$]$ $]$

VInv: v $\rightarrow$ **v** np
EXAMPLE: $[_v$ $[_v$ did $]$ $[_{np}$ you$]$ $]$ break the fence

VP1: vp $\rightarrow$ **vp/v** X[16]
EXAMPLE: nothing in Articles 10, 11 and 14 shall be $[_{vp}$ $[_v$ regarded$]$ $[_{pp}$ as preventing the high contracting parties from imposing restrictions on the political activity of aliens$]$ $]$.

VP2: vp $\rightarrow$ **vp/v** XY
EXAMPLE: Nothing in Articles 10, 11 and 14 shall be regarded as $[_{vp}$ $[_v$ preventing $]$ $[_{np}$ the high contracting parties $]$ $[_{pp}$ from imposing restrictions on the political activity of aliens$]$.

DetNP: np $\rightarrow$ det **n/nbar/nbbar**
EXAMPLE: $[_{np}$ $[_{det}$ the $]$ $[_{nbar}$ high contracting parties $]$ $]$

AdjComp1: adj $\rightarrow$ **adj** X
EXAMPLE: the fence was $[_{adj}$ $[_{adj}$ easy $]$ $[_{vp}$ to break ϵ $]$ $]$

AdjComp2: adj $\rightarrow$ **adj** XY
EXAMPLE: the fence was $[_{adj}$ $[_{adj}$ easy $[_{pp}$ for John $]$ $[_{vp}$ to break ϵ $]$ $]$

[15] The symbols given in the schematized representation of the rules are values of the attribute CAT. This feature is the main restrictor in parsing and generation and can be viewed as an abbreviation of some important properties of the constituent. The values, however important from the algorithm's point of view, have a limited linguistic role, since they merely express in a compact manner information present at other locations (word class of the head, status of the subcategorization list, left modification of the noun). The head of the structure licensed by the rule is printed in bold type. — Although the rule Sentence has two items on the right-hand side, a punctuation mark is not treated as a real modifier. Therefore this rule is of the head_only_rule type.

[16] The category of the complement is specified by subcategorization requirements of the head daughter rather than directly by the rule.

head_adjunct_rule: VPAdvb, AdvbVP, SCommaAdvb, AdvbS, AdvbCommaS, AdjN, PreDetN, NA, NRel, AdvAdjP

VPAdvb: vp → **v/vp** pp/adv/scls
EXAMPLE: Hamish $[_{vp}$ $[_{vp}$ hugged Sinéad] $[_{pp}$ by the fence]]. The commission shall $[_{vp}$ meet $[_{scls}$ as the circumstances require]].

AdvbVP: vp → adv **vp**
EXAMPLE: Hamish $[_{vp}$ $[_{adv}$ already] $[_{vp}$ must have been here]].

SCommaAdvb: s → **s** "," pp/adv/scls
EXAMPLE: $[_{s}$ $[_{s}$ Men and women of marriageable age have the right to marry and to found a family], $[_{pp}$ according to the national laws governing the exercise of this right]].

AdvbS: s → pp/adv/scls **s**
EXAMPLE: $[_{s}$ $[_{adv}$ Yesterday] $[_{s}$ Hamish kissed Sinéad]].

AdvbCommaS: s → pp/adv/scls "," **s**
EXAMPLE: $[_{s}$ $[_{pp}$ Subject to the provisions of Article 29], $[_{s}$ the commission shall take its decisions by a majority of the members present and voting]].

AdjN: nbar → adj/adjp/n **n/nbar**
EXAMPLE: the two $[_{nbar}$ $[_{adjp}$ extremely happy] $[_{n}$ winners]]

PreDetN: nbbar → predet **n/nbar**
EXAMPLE: the $[_{nbbar}$ $[_{predet}$ two] $[_{nbar}$ extremely happy winners]]

NA: n → **n** pp/vp/app/adj/adjp
EXAMPLE: the $[_{n}$ $[_{n}$ rights and freedoms] $[_{vp}$ defined in Section I of this convention]]

NRel: n → **n** s
EXAMPLE: The $[_{n}$ $[_{n}$ members] $[_{s}$ whose terms are to expire at the end of the initial period of three years]] shall be chosen by lot.

AdvAdjP: adjp → adv **adj/adjp**
EXAMPLE: $[_{adjp}$ $[_{adv}$ very] $[_{adj}$ difficult]]

head_marker_rule: SCls, VPcle, PP

SCls: scls → subord_conj **s**
EXAMPLE: The court may only deal with a case $[_{scls}$ $[_{subord\_conj}$ after] $[_{s}$ the commission has acknowledged the failure of efforts for a friendly settlement]] and within the period of three months provided for in Article 32.

VPcle: v → **v** pcle
EXAMPLE: The court shall $[_{v}$ $[_{v}$ draw] $[_{pcle}$ up]] its own rules and shall determine its own procedure.

PP: pp → prep **np/vp**[17]
EXAMPLE: This article shall not prevent states $[_{pp}$ $[_{prep}$ from] $[_{vp}$ requiring the licensing of broadcasting, television or cinema enterprises]].

aux_head_complement_rule: AuxV

AuxV: vp → **vp/v** X
EXAMPLE: $[_{vp}$ $[_{v}$ has] $[_{vp}$ been referring to you]]

coord_rule: Coord

[17] Should English prepositions be *markers* or *heads*? Some arguments might favour the alternative solution of prepositions as heads:

- Heads are never traces, except when the head of prepositional phrase is the nominal.
- Conceptually, the information about prepositional marking (PFORM) should be of the same type as the information about case. However, unlike CASE, PFORM cannot be a head feature.

Coord: $X \rightarrow X$ coord_conj X

EXAMPLE: Nothing in Articles $[_{app}$ $[_{app}$ 10 $]$ $[_{coord\_conj}$, $]$ $[_{app}$ $[_{app}$ 11 $]$ $[_{coord\_conj}$ and $]$ $[_{app}$ 14 $]$ $]$ $]$ shall be regarded as preventing the high contracting parties from imposing restrictions on the political activity of aliens.

trace_rule: Trace

Trace: np/pp $\rightarrow \epsilon$

EXAMPLE: The jurisdiction of the court shall extend to all cases$_1$ concerning the interpretation and application of the present convention which$_1$:2 the high contracting parties or the commission shall refer $[_{np}$ ϵ $]_1$:2 to it in accordance with Article 48.

head_filler_rule: FH

FH: s $\rightarrow$ np/pp s

EXAMPLE: The jurisdiction of the court shall extend to all cases$_1$ concerning the interpretation and application of the present convention $[_s$ $[_{np}$ which$_1$:2 $]$ $[_s$ the high contracting parties or the commission shall refer ϵ_1:2 to it in accordance with Article 48 $]$ $]$.

6.3 Overview of results

The English component covers the following linguistic phenomena:

nominal group:

- (pre)determination, prepositional marking
- modification by adjectives, participles, prepositional phrases, relative clauses

verbal group:

- past tense, passive voice, progressive and perfective tenses, modality, negation, inversion
- complementation of verbs by any categories specified in subcategorization requirements, modification by free adjuncts at any position within the verb's head domain
- distinction of constructions involving controlled embedded predicates into *raising* and *equi* types on the basis of the information in the controlling verb's subcategorization list
- *unbounded dependencies*: topicalisations, relative expressions, embedded relative expressions (noun phrases, prepositional phrases, noun phrases extracted from prepositional phrases), adjectives of the *tough/easy* type

coordination of phrasal and lexical categories

other linguistic phenomena related to unbounded dependencies:

- Complex NP Constraint (a relative clause cannot include a trace)[18]
- Trace Principle (every trace must be subcategorized, but not as a subject)[19]
- Subject Condition (a trace may occur inside a subject only when it is subcategorized by another item in the SC list.[20].
- Coordination Principle ('strong version' – identity of NONLOC values in conjuncts and the mother)
- prepositional phrase cannot be a trace in a *that-less* relative clause (**I loved the book*$_1$ *[to* $\_\__1$*]:2 she referred* $\_\__1$*:2.*)

[18] Counterexamples exist: *Moby Dick*$_1$*, I was the only person that hadn't read* $\_\__1$.

[19] In an ungrammatical sentence **How*$_1$ *do you think the building is* $\_\__1$ *tall?* the trace is not subcategorized by *tall*. In Czech this limitation does not apply: *Jak*$_1$ *myslíš, že je ta budova* $\_\__1$ *vysoká?*

[20] In a sentence *Who*$_1$ *did my talking to* $\_\__1$ *bother* $\_\__1$*?*, the trace is in the subject *my talking to* _ and, at the same time, the whole object of *bother* is a trace, unlike in a sentence: **Who*$_1$ *did my talking to* $\_\__1$ *bother Hillary?* Similarly *The leader*$_1$ *who rivals of* $\_\__1$ *assassinated* $\_\__1$, unlike **The leader*$_1$ *who rivals of* $\_\__1$ *assassinated the consul*

The English component has been tested on made-up examples which should verify its functionality on specific phenomena and on a part of a legal document *Declaration of Human Rights*.

Thirteen sentences from this document have been processed by the English component (ELUENG) and by two publicly accessible parsers with good references: (ENGCG, Karlsson et al. 1994 and Link Grammar, Sleator & Temperley, 1991) and finally by the system APAČ (Kirschner, 1987). The results are listed in the following table:

	ENGCG	LINK GRAMMAR	APAČ	ELUENG
number of correctly analyzed sentences	5	6	3	6
number of ambiguous sentences included	2	0	0	4
number of wrong analyses	8	7	10	7

In spite of the very small amount of data and the fact that the English component could have been well predisposed for the enterprise, the results may give some argument for cautious optimism.

7 Specifics of Czech

Viewed from the perspective of formal linguistics, Czech is an interesting challenge. The language as such is rich in morphology and has a relatively free word order. These two properties alone disqualify a number of standard formal approaches and software tools. On the other hand, Czech has been investigated in a linguistic tradition which bears fruit in a number of theoretical aspects of language-universal nature, one of the most prominent aspects being, e.g., the field of topic–focus articulation. Another consequence of the local but, at the same time, more universally oriented tradition has been the awareness of the need for a more flexible alternative to the English-centred constituency-based formal grammars in the Chomskyan tradition. With the years passing, this realization began to grow in some overseas scientific communities as well. Today's linguistic theories which have their roots in strict constituency exhibit many properties which make for a distinctly converging trend.

As in the case of English, the present approach towards the description of Czech is based on a combination of context-free rules with a dependency-based representation, all within a monostratal unification formalism. The marked difference consists in that the context-free rules as such have very little linguistic content in the sense that the geometry of the derivation tree is determined by a formal strategy rather than by considerations of purely linguistic nature, such as which constituents form a more cohesive unit. Typically, a categorial label used in a rule is more a reflection of the formal derivation strategy than of the linguistic properties of the constituent. Consequently, the whole formal description rests even more heavily than in the case of English on lexical specifications.

Within unification-based context-free framework, solutions for languages with a significant amount of word-order variations (mostly for German) have been proposed, i.a., by Uszkoreit (1987), Reape (1989), Hinrichs & Nakazawa (1993), and Oliva (1992). The issues at stake are the following:

- Should CF rules license syntactic constituents in the traditional sense (as in Reape) or should the rules merely reflect a formal strategy?
- Solution to the previous issue can be related to the problem whether the rules should allow for more than two daughters or whether they should be restricted to binary branching.
- Should the structure be both right and left-branching or should it allow only a single direction? This problem is related to language-specific considerations (German tends to be a head-final language, at least at clausal level), but unidirectional branching can be imposed also on languages which are much loosely specified in this respect (for Czech, see Oliva, 1992).
- How should the varied constraints on word order be imposed? Word order constraints expressed independently of the order of sisters in a CF rule are easy to impose when they do not go beyond the single rules. This condition is difficult to maintain if the structure is binary-branching. In such cases a less straightforward solution must be found.
- Is there a special mapping component necessary between surface string and leaves of the derivation tree (as in Reape)? If the structure is rather flexible, as in binary approaches, there is less need for this split, which in effect adds another stratum to the framework.

The present approach is based on binary bidirectional branching structure. This way, a standard formalism and processing tool can be employed and a straightforward transparent derivational history maintained. On the other hand, constraints on word order have to be specified in a more complex fashion.

7.1 The Czech lexicon

Unlike for English, no electronic source of Czech lexical data has been used. Currently, the available sources provide word class and inflectional information. Valency information is available only for a handful of items.

This situation is changing, so the approach towards the construction of lexicon for the Czech component has been based on the following considerations:

- It would be unrealistic to expect that the amount of Czech lexical data could approach the figures for English without recourse to available electronic sources. Once adequate sources are available, integration into the system should be possible, based on the experience with the English data.
- The adopted strategy was to verify those parts of lexical description which are not lexeme-specific: the definition of inflectional paradigms and the definition of more general classes specifying valency requirements and data structure.
- An option is open whether a morphological component should be imported together with the lexeme-specific description. In any case, an integrated morphological component has been tested for a significant number of paradigms.

The specification of inflectional paradigms is a major component of the lexical description of Czech. Here, as well as in other parts of the description, the power of default inheritance hierarchy has been employed.

Ideally, such a description of morphology should profit from any case of similarity across inflectional paradigms. Thus, the plural instrumental ending *-mi*, which is realized in most nominal paradigms, could be specified only once for the whole class of the paradigms. However theoretically interesting this solution might be, problems of multiplicity of lexical descriptions with locally restricted differences in morphological categories would arise. Given the present formalism, the flaw consists in that a homographical ending within a paradigm would give rise to multiple lexical items. This is due to the impossibility to reconcile the two aspects of morphological description: the aspect of shared endings for identical category values among paradigms and the aspect of homograph endings for different category values within a single paradigm.[21]

An additional problem, which would lead to spurious ambiguities under a theoretically plausible organization, is the necessity to specify variant forms not as atomic disjuncts, but rather as marked and unmarked (due to the resulting uncertainty in generation).

For these reasons, a more pragmatic approach has been selected. There are a few major classes of inflectional paradigms which dominate more specific paradigms. These 'minor' paradigms may differ very slightly from their 'majors' and here the power of default inheritance is fully utilized.

The possibility to describe morphological processes like palatalization without recourse to multiple paradigms has been verified for adjectives. Other word classes could be treated similarly.

So far, most nominal and adjectival declensions have been described. Conjugational paradigms have been specified for a few major classes.

Syntactic properties of lexical items have been specified to a degree sufficient for evaluating functionality of the description for a subset of linguistic phenomena. Most notably, description of major types of valency requirements have been verified with a space open for the specification of other types. The described types include controlled embedded verbal structures and combinatory properties of auxiliaries and modals (see Sample 3).

For a list of sample lemmas see Sample 4.

[21] If the description of an ending is distributed over a hierarchy of classes according to the ending's specificity to a lesser or larger number of paradigms, a possible homography within a single paradigm is impossible to express as a local disjunction of values of a morphological category. As a result, multiple structures, differing only in the value of a morphological category, are assigned to a single word form.

The restriction of disjunction to atomic values in the present formalism further complicates an attempt to preclude multiplicity of structures.

7.2 Grammar rules for Czech

The grammar describes the core of syntactic phenomena, preserving the methods used for English where it is useful. The strategy was to concentrate on the differences between English and Czech syntax, in order to verify viability of the approach. Solutions for other phenomena are expected to be relatively easy importable/adaptable.

All rules are binary rules, except for the rule S, which is necessary for the description of the unexpressed subject. Three rules specify a fixed precedence of daughters: the rule for prepositional marking, adjectival modification and modification of noun by a prepositional phrase. Other rules occur in two versions, reflecting the two possible orderings of daughters: head-final (left-branching structure) or head-initial (right-branching structure). Starting upwards from the lexical head, the lower head projections first include all modifiers (complements or adjunct) standing to the right (if any), while the higher projections include modifiers to the left (if any). This means that in generation of a single head domain, left-branching (head-final) rules are applied before right-branching (head-initial) rules. For parsing, the order is reversed.

The values of CAT are used primarily for controlling this strategy. Head-initial rules do not alter the category symbol of the head, allowing for recursive application. Head-final rules assign a distinct symbol to the mother, appending the suffix `-bar` to the category symbol of the head (e.g. `vbar, nbar`). Recursive application of head-final rules is made possible by disjunctive specification of the head's category (e.g. `v/vbar`).

Sample 12 shows derivational trees for different word order variants of the same sentence.

The following list describes schematically instantiations of some rule types for Czech.

head_only_rule: S

S: s → **v/vbar**
EXAMPLE: $[_s$ $[_{vbar}$ Honza je na poli]], $[_s$ $[_v$ Pomáhal jsem Honzovi pást husy]]

head_complement_rule: VC, CV

VC: v → **v** X[22]
EXAMPLE: Honza $[_v$ $[_v$ dal Marii] pusu]], $[_v$ $[_v$ nemohl jsem] přijít]]

CV: vbar → X **v/vbar**
EXAMPLE: $[_{vbar}$ Honza $[_v$ dal Marii pusu]]

head_adjunct_rule: VA, AV, NA, AdjN

VA: v → **v** pp/adv
EXAMPLE: $[_v$ $[_v$ Potkal jsem ji] $[_{pp}$ v divadle]]

AV: vbar → pp/adv **v/vbar**
EXAMPLE: $[_{vbar}$ $[_{pp}$ s těžkou nůší] $[_v$ pomohl ženě předseda]]

NA: n → **n** pp
EXAMPLE: $[_n$ $[_n$ slepice] $[_{pp}$ s kuřaty]]

AdjN: nbar → adj **n/nbar**
EXAMPLE: $[_{nbar}$ velká $[_{nbar}$ rybí $[_n$ kost od Marie]]]

head_marker_rule: VAux, AuxV, PP

VAux: v → **v** vaux
EXAMPLE: $[_v$ $[_v$ nemohla] $[_{vaux}$ jsem]] mu pomoci

AuxV: vbar → vaux **v/vbar**
EXAMPLE: v lese $[_{vbar}$ $[_{vaux}$ jsem] $[_{vbar}$ se třásl strachy]]

PP: pp → prep **n/nbar**
EXAMPLE: $[_{pp}$ $[_{prep}$ na] $[_{nbar}$ vysokém stavení]]

[22]The category of the complement is specified by subcategorization requirements of the head daughter rather than directly by the rule.

7.3 A solution to an interesting problem: clitics in Czech

There is a class of (arguably) sentential clitics in Czech which tend to occur at what has traditionally been called Wackernagel's position, i.e. the second position following first unit in a clause. Some prominent members of this class are: past tense auxiliary, reflexive particles, weak dative and accusative personal pronouns. If more of them are present, they form a cluster and obey the order given above. The nature of the first unit is a matter of argument. (For some proposals see Koktová (to appear), Avgustinová & Oliva 1995.) Although there are examples like (21), where the first unit is not a single syntactic unit (subtree or constituent in the usual sense), the approach described below takes a syntactic unit as the first approximation.

(21) a. Z Prahy do Košic se jede skoro celý den
from Praha to Košice REFL goes almost whole day

b. Včera večer na chodníku před naším domem jsem potkal ježka
last night on pavement in front of our house AUX_{1sg} met hedgehog

Similarly, speakers' acceptability judgments do not provide a clearcut argument in all cases. Although the second position is overwhelmingly preferred, it is not an iron rule. Still, the degree of acceptability decreases significantly when the clitic is placed adjacently to the verb at other than Wackernagel's position and even more markedly when it is at some other position in the verb's non-recursive head domain (between its immediate dependents). Some positions are impossible: the initial position in the clause, as well as positions within embedded constituents.

This applies to the matrix (finite) clause. Clitics from embedded (mostly infinitival) clauses may climb to the second position of the matrix clause, but may also stay within the embedded clause. Again, acceptability judgments are not unanimous, although climbing versions seem to be preferred.[23]

If two or more clitics of the same type should meet in the same cluster as a result of the climbing, all but one are obligatorily deleted.[24]

---

[23]This is true mainly about infinitival clauses as complements of the matrix verb, e.g.:

(22) a. Honza se odmítá oholit
John REFL refuses to shave

b. Honza odmítá se oholit
John refuses REFL to shave

When the infinitive modifies a noun of the matrix clause, the climbing version is less acceptable or even excluded, depending probably on the degree of idiomaticity of the connection between the matrix verb and the modified noun. In (23) and (24) both versions are acceptable (*dostat se* 'to get [somewhere]' and *napít se* 'to have a drink' are true reflexive verbs).

(23) a. Honza se tam ještě má šanci dostat
John REFL there still has chance to get

b. Honza má ještě šanci se tam dostat
John has still chance REFL there to get

(24) a. Honza se má chuť napít
John REFL has taste to have a drink

b. Honza má chuť se napít
John has taste REFL to have a drink

However, in (25) and (26) the clitic climbing versions are impossible.

(25) a. * Honza se dostal chuť napít
John REFL got taste to have a drink

b. Honza dostal chuť se napít
John got taste REFL to have a drink

(26) a. * Honza se pojal úmysl učit anglicky
John REFL acquired intention to learn English

b. Honza pojal úmysl se učit anglicky
John acquired intention REFL to learn English

I owe this observation to Petr Sgall.

[24]As in (27), whose (intended) meaning is 'John (has) decided to shave himself', where both verbs are reflexive (*rozhodnout se* 'to decide' is a true reflexive):

So far, the clitic climbing versions are not licensed by the grammar.

To control word-order phenomena of this type, which are combined with 'topological' requirements (second position), within a binary branching context-free framework, the virtually wide-scope constraints have to be translated into local ones, not extending beyond binary-branching rules. This approach has been used previously e.g. in Engelkamp et al. (1992), Oliva (1992). An additional complexity is added due to the fact that the grammar is both left and right branching. In effect, every constraint has two versions: one for the head-final rules, one for the head-initial rules. The essence of the solution is in a combination of two types of constraints: (i) constraints controlling the second position as a possible location for a clitic cluster, e.g. a constraint requiring that a clitic may not occur as a non-head daughter in a continuous sequence of head-final local trees from a maximal verbal projection downwards, (ii) constraints controlling the sequence of clitics in a cluster; such constraints specify requirements which every non-head daughter at lower projection levels must satisfy once the non-head daughter at the current level is a clitic of a certain type. These constraints are implemented by macros which control the interaction of several dedicated features.

Sample 13 provides some examples of sentences with clitics.

8 Conclusion

The effort has been fruitful in the following points:

- A substantial re-usable lexical data base for English has been accumulated.
- A common lexically based approach, combining context-free grammar with dependency-oriented representation has been verified for two typologically different languages in the process of parsing and generation.
- The viability of the treatment of Czech morphology within a hierarchically structured lexical component has been demonstrated.
- A number of standard and some unorthodox theoretically relevant solutions to various phenomena have been implemented and tested with satisfactory results.
- The whole model has proved to be flexible and open to further development as a kind of linguistic testbed.

The unfulfilled plans include a richer Czech lexical data base and the missing transfer link. Together with a substantial grammar development and refinement of lexical data, these plans constitute the obvious major points in the envisaged continuation of the project.

It is expected that Czech lexical data will be acquired from an electronic source. The option of whether a morphological component should be imported into the system with the lexical data, replacing the existing component, remains open. As originally intended, data for the transfer component should be extracted from a range of sources, including bilingual dictionaries in the electronic form and parallel bilingual texts, employing some available methodologies.

In a more distant perspective, the rule-based description of linguistic competence for the two languages should be augmented with a model of simulated competence and performance knowledge, based on data extracted from corpora by statistical techniques.

---

(27) a. * Honza se se rozhodl oholit
John REFL REFL decided to shave

b. Honza se rozhodl oholit
John REFL decided to shave

c. Honza se rozhodl oholit se
John REFL decided to shave REFL

References

Avgustinova, T. and K. Oliva (1995) 'On the Nature of the Wackernagel Position in Czech'. *Proceedings of FDSL 95*, University of Leipzig

Estival, D. (1990), *ELU User Manual* Technical Report 1, ISSCO, Geneva.

Engelkamp, J., G. Erbach and H. Uszkoreit (1992) Handling Linear Precedence Constraints by Unification. In *Proceedings of ACL '92*, Newark.

Estival, D., A. Ballim, G. Russell and S. Warwick (1990) "A Syntax and Semantics for Feature Structure Transfer". *Proceedings of the 3rd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, University of Texas at Austin.

Hajičová, E. & A. Rosen (1994) 'Machine Readable Dictionary as a Source of Grammatical Information', in A. Zampolli *et al.* (eds.) *Current Issues in Computational Linguistics: In Honor of Don Walker*, Kluwer Academic Publishers.

Hinrichs, E. & Ts. Nakazawa (1993) *Aspects of German VP Structure – An HPSG Account.* SfS–Report–01-93, Seminar für Sprachwissenschaft, Computerlinguistik, Eberhard-Karls-Universität, Tübingen.

Hornby, A.S. (1974) *Oxford Advanced Learner's Dictionary of Current English*, 3rd edition, Oxford University Press, London.

Karlsson, F., A. Voutilainen, J. Hikkilä & A. Anttila (1994) *Constraint Grammar: a Language–Independent System for Parsing Unrestricted Text.* Mouton de Gruyter.

Kirschner, Z. (1987) APAC3-2: An English-to-Czech Machine Translation System, EBSAT XIII, MFF UK Praha.

Koktová, E. (to appear) *Word-Order Based Grammar*, Mouton de Gruyter.

Mitton, R. (1986) 'A partial dictionary of English in Computer-Usable Form', in *Literary and Linguistic Computing* 1:214-215.

Oliva, K. (1992) 'The proper treatment of word order in HPSG.' In *Proceedings of COLING 1992*, Nantes.

Pollard, K., and I. Sag (1994) *Head-Driven Phrase Structure Grammar.* Stanford: Center for the Study of Language and Information and Chicago & London: The University of Chicago Press.

Reape, M. (1989) 'A Logical Treatment of Semi-Free Word Order and Bounded Discontinuous Constituency'. *Proceedings of the 4th Meeting of the European Chapter of the ACL.* Manchester.

Russell G., A. Ballim, J. Carroll, S. Warwick-Armstrong (1992) 'A Practical Approach to Multiple Default Inheritance for Unification-Based Lexicons', Special Issue on Inheritance II, *Computational Linguistics*, Vol. 18, No. 3, 311-337.

Sgall P., E. Hajičová and J. Panevová (1986) *The Meaning of the Sentence in its Semantic and Pragmatic Aspects* (edited by J. Mey), Reidel, Dordrecht / Academia, Praha.

Shieber, S., van Noord, G., R.C. Moore and F.C.N. Pereira (1989). "A Semantic-Head-Driven Generation Algorithm for Unification-Based Formalisms". In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver.

Sleator, D. & D. Temperley (1991) Parsing English with a link grammar. Technical Report CMI–CS–91–196, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA

Touretzky, D. S. (1986). *The Mathematics of Inheritance Systems.* Pitman Publishing.

Uszkoreit, H. (1987). *Word Order and Constituent Structure in German.* CSLI Lecture Notes No. 8. CSLI, Stanford.

9 Appendix

9.1 Sample 1: common definitions

```
## declaration of variables for dictionary look-up
# Declare
      Form = <synsem loc head form>
      Category = <synsem loc cat>
      Identifier = <synsem loc head base>

## declaration of restrictors
# Restrict Cat Form Trace Base Mod
      <* synsem loc cat> = Cat
      <* synsem loc head form> = Form
      <* trace> = Trace
      <* synsem loc head base> = Base
      <* synsem loc modified left> = Mod

## definition of types
# Typedef
      Phrase = (ortho,cd,trace,synsem,dtrs)
      Synsem = (loc,nonloc)
      Loc = (cat,coord,head,sc,modified,bare,deep)
      Nonloc = (inher,to-bind)
      Deep = (gov,deps)
      Dependent = (fun,dir,val)
      Lexical = (ortho,cd,trace,synsem)
      NHead = (form,base,wtype,wcl,ntype,predicative,attributive,count,case,mark,spec)
      NGov = (lexeme,wcl,index,pform,definite,conj)
      VHead = (form,base,wtype,wcl,vform,predicative,attributive,inv,negated,
               voice,tense,aspect,spec,frame)
      VGov = (lexeme,wcl,pform,sconj,voice,tense,modality,progressive,
               perfective,polarity,conj)
      Index = (pers,num,gend)
      Head_Only_Dtr = (head_dtr)
      Head_Complement_Dtrs = (head_dtr,comp_dtr)
      Head_Adjunct_Dtrs = (head_dtr,adjunct_dtr)
      Head_Filler_Dtrs = (head_dtr,filler_dtr)
      Head_Marker_Dtrs = (head_dtr,marker_dtr)
      Coord_Dtrs = (left_coord_dtr,conj_dtr,right_coord_dtr)
```

9.2 Sample 2: sample definitions of macros

English

```
## Subject control

SubjControl(Subj,VComp,Vform)
      Vform = ~adj
      <synsem loc sc> = [Subj VComp]
      <VComp loc cat> = v/vp
      <VComp loc head vform> = Vform
      <VComp loc sc> = [Subj]

SubjControl(Subj,VComp,adj)
      <synsem loc sc> = [Subj VComp]
      <VComp loc cat> = adj/adjp
      <VComp loc sc> = [Subj]

## Object control, equi

SOE(Cler,ClsComp)
      <ClsComp loc sc> = [Clee]
      <Clee loc deep> = <Cler loc deep>

## Subject condition

SubjectCondition(C,H)
      <C synsem nonloc inher slash> = []

SubjectCondition(C,H)
      <C synsem nonloc inher slash> = [_]
      <H synsem nonloc inher slash> = [_]

## Head feature principle

HFP(H)
      <* synsem loc head> = <H synsem loc head>
      <H trace> = no

## Subcategorization principle

SCP(H,C)
      <H synsem loc sc>
        = <* synsem loc sc> ++ [<C synsem>]

SCP(H,C1,C2)
      <H synsem loc sc>
        = <* synsem loc sc> ++ [<C1 synsem> <C2 synsem>]

Head_Complement_Rule(H,C)
      !HFP(H)
      !HDeep(H)
      !MotherTypes
      !HCDtrs(H,C)
      !SCP(H,C)
```

```
    !NonlocFP(C,H)
```

Czech

```
## Subcategorization principle

Cz_SCP(H)
     <* synsem loc subj> = <H synsem loc subj>
     <* synsem loc comps> = <H synsem loc comps>

Cz_SCP(H,C)
     <C synsem> = X
     <* synsem loc comps>
       = <H synsem loc comps> -- X
     <* synsem loc subj> = <H synsem loc subj>
     !Cz_Aux_Test(H,C)

Cz_SCP(H,C)
     <H synsem loc head vform> = fin
     <C synsem> = X
     <H synsem loc subj> = [X]
     !SubjVerbAgr(X,H)
     <* synsem loc subj> = []
     <* synsem loc comps> = <H synsem loc comps>
     !HDeep(H)

Cz_Aux_Test(H,C)
     <H synsem loc head wtype> = content
     !HDeep(H)

Cz_Aux_Test(H,C)
     <H synsem loc head wtype> = function
     <synsem loc deep gov> = <H synsem loc deep gov>
     <synsem loc deep deps>
       = <C synsem loc deep deps>
         ++ <H synsem loc deep deps>

## Subject control
SubjControl_Cz(Subj,VComp,Vform)
     <synsem loc subj> = [Subj]
     <synsem loc comps> = [VComp]
     <VComp loc cat> = v/vbar
     <VComp loc head vform> = Vform
     <VComp loc subj> = [Subj]
     <VComp loc comps> = []

## Object control, Equi
SOE_Cz(Cler,ClsComp)
     <ClsComp loc subj> = [Clee]
     <ClsComp loc comps> = []
     <Clee loc deep> = <Cler loc deep>
```

9.3 Sample 3: sample definitions of lexical classes

Czech

```
## VERBS

# Class Verb (GeneralVerb VerbFrame SubjectAgr ContentWord)
     <Gov sconj> = '-NIL-'
     <Gov voice> = active
     <Gov modality> = '-NIL-'
     <Gov progressive> = no
     <Gov perfective> = no
     <Gov tense> = <Head tense>
     <Gov polarity> = Polarity
     <Gov pform> = '-NIL-'

# Class GeneralVerb ()
     <* synsem loc cat> = v
     Head == VHead
     <Head wcl> = verb
     <Loc infl> == VInfl
     Gov == VGov
     <Gov pform> = '-NIL-'
     ObjCase = acc

# Class SubjectAgr ()
     <Loc subj> = [Subj]
     <Subj loc cat> = n/nbar
     <Subj loc subj> = []
     <Subj loc comps> = []
     |
     <synsem loc head vform> = fin
     <Subj loc head case> = nom
     |
     <synsem loc head vform> = ~fin

# Class VerbFrame ()
     <synsem loc cat> = v
     <synsem loc head frame> = Frame
     |
     Frame=i
     !VComps([])
     !Actants(Subj,actor)
     |
     Frame=n
     !VComps([Obj])
     !Actants(Subj,actor,Obj,patient)
     !Complement(Obj,n/nbar,ObjCase)
     |
     Frame=refl
     !VComps([ReflPcle])
     !Actants(Subj,actor)
     <ReflPcle loc cat>= n
     <ReflPcle loc head wscl> = refl_pcle
     <ReflPcle loc head case> = ReflCase
```

```
     <ReflPcle loc head base> = ReflForm
     !LexSuffix(ReflForm)
     |
     Frame=iobj_n
     !VComps([IObj Obj])
     !Actants(Subj,actor,IObj,addressee,Obj,patient)
     !Complement(IObj,n/nbar,dat)
     !Complement(Obj,n/nbar,acc)
     |
     Frame=iobj_inf
     !VComps([IObj Obj])
     !Actants(Subj,actor,IObj,addressee,Obj,patient)
     !Complement(IObj,n/nbar,dat)
     !Complement(Obj,inf)
     !SOE_Cz(IObj,Obj)
     |
     Frame=pred
     !VComps([Pred])
     !Actants(Subj,actor,Pred,patient)
     !Complement(Pred,n/nbar/adj,nom/ins)
     <Pred loc head predicative> = yes
     <Pred loc head spec> = Subj
     |
     Frame=pred
     !VComps([Pred])
     !Actants(Subj,actor,Pred,patient)
     <Pred loc cat> = pp/adv
     <Pred loc head predicative> = yes
     !Empty_sc(Pred)

# Class Perf (Ng)
     <Gov perfective> = yes
     <Head aspect> = perfective

# Class Impf (Ng)
     <Gov perfective> = no
     <Head aspect> = imperfective

## VERBINFL

# Class VerbInfl ()

# Class Ng ()
     |
     <synsem loc head form> = F
     <synsem loc head negated> = no
     Polarity = positive
     |
     <synsem loc head form> = 'ne' && F
     <synsem loc head negated> = yes
     Polarity = negative

## ne2st
```

```
# Class NE (VerbInfl)
     !StemIt(Inf)
     | !FVerbInfl(F,Stem,u,fin,pres,1,sg,_,1)
     | !FVerbInfl(F,Stem,es3,fin,pres,2,sg,_,1)
     | !FVerbInfl(F,Stem,e,fin,pres,3,sg,_,1)
     | !FVerbInfl(F,Stem,eme,fin,pres,1,pl,_,1)
     | !FVerbInfl(F,Stem,ete,fin,pres,2,pl,_,1)
     | !FVerbInfl(F,Stem,ou,fin,pres,3,pl,_,1)
     | !FVerbInfl(F,Inf,'',inf,x,x,x,x,1)
     | !FVerbInfl(F,Inf,'i',inf,x,x,x,x,2)
     | !FVerbInfl(F,PStem,l,fin,past,3,sg,ma/mi,1)
     | !FVerbInfl(F,PStem,la,fin,past,3,sg,f,1)
     | !FVerbInfl(F,PStem,lo,fin,past,3,sg,n,1)
     | !FVerbInfl(F,PStem,li,fin,past,3,pl,ma,1)
     | !FVerbInfl(F,PStem,ly,fin,past,3,pl,mi/f,1)
     | !FVerbInfl(F,PStem,la,fin,past,3,pl,n,1)

## ne2st
# Class NE1 (NE)
     !StemIt(Stem,s,'2st')
     PStem = Stem

## ve2zt
# Class NE2 (NE)
     !StemIt(Stem,z,'2zt')
     PStem = Stem

## tr3a2st
# Class NE3 (NE)
     !StemIt(Stem,es,a2st)
     !StemIt(PStem,'t')

## za2bst
# Class NE4 (NE)
     !StemIt(Stem,eb,a2bst)
     !StemIt(PStem,'st')

## kla2st, kra2st, ve2st
# Class NE5 (NE)
     !StemIt(Stem,d,'2st')
     PStem = Stem

## me2st, kve2st
# Class NE6 (NE)
     !StemIt(Stem,t,'2st')
     PStem = Stem

## pa2st
# Class NE7 (NE)
     !StemIt(Stem,s,'2st')
     !StemIt(PStem,s,'st')

## ru4st
# Class NE8 (NE)
```

```
!StemIt(Stem,ost,'u4st')
PStem = Stem
```

9.4 Sample 4: sample definitions of lemmas

English

```
#Word reach                (SibStemV Verb) Frame=advl./pp-obj_/n/n.ppfor-iobj_n/n_pcle_
                           VP=[2C,3A,6A,12B,13B,15B] TI=ti Status=1 Usage=[s,s] RevDat=950118
#Word receipt              (CStemN CountUncount Noun) Status=1 Usage=[s,s] RevDat=930118
#Word receive              (EStemV Verb) Frame=n VP=[2A,6A] TI=ti Status=1 Usage=[s,s]
                           RevDat=950531
#Word reciprocity          (NoPlN Uncount Noun) Status=1 Usage=[s,s] RevDat=940118
#Word refer                (DblStemV Verb) Frame=pp-obj_/n_pp_./n_pcle_ Prep-Obj=to
                           Pcle-Obj=back VP=[3A,14,15B] TI=ti Status=1 Usage=[s,s] RevDat=950927
#Word regard               (CStemN CountUncount Noun) Status=1 Usage=[s,s] RevDat=930118
#Word regard               (CStemV Verb) Frame=n/n_pp_./n_ping-objc_/n_pp-objc_
                           Prep-Objc=as VP=[6A,14,16B] TI=t Status=1 Usage=[s,s] RevDat=950426
#Word regardless           (NoInflJ Adj) Status=1 Usage=[s,s] RevDat=930118
#Word regardless_of        (Prep) Status=1 Usage=[s,s] RevDat=950620
#Word remainder            (CStemN Count Noun) Status=1 Usage=[s,s] RevDat=930118
#Word replace              (EStemV Verb) Frame=n/n_pp_./n_advl VP=[6A,14,15A] TI=t
                           Status=1 Usage=[s,s] RevDat=930118
#Word reputation           (CStemN CountUncount Noun) Status=1 Usage=[s,s] RevDat=930118
#Word request              (CStemN CountUncount Noun) Status=1 Usage=[s,s] RevDat=930118
#Word request              (CStemV Verb) Frame=n/thcls/n_inf. VP=[6A,9,17] TI=t Status=1
                           Usage=[s,s] RevDat=930118
#Word require              (EStemV Verb) Frame=n/ing/thcls/n_pp_./n_inf. VP=[6A,6D,9,14,17]
                           TI=t Status=1 Usage=[s,s] RevDat=930118
#Word responsibility       (YStemN CountUncount Noun) Status=1 Usage=[s,s] RevDat=930118
#Word restriction          (CStemN CountUncount Noun) Status=1 Usage=[s,s] RevDat=930118
#Word right                (Adv) Status=1 Usage=[s,s] RevDat=930118
#Word right                (CStemN CountUncount Noun) Status=1 Usage=[s,s] RevDat=930118
#Word right                (NoInflJ Adj) Status=1 Usage=[s,s] RevDat=930118
#Word rule                 (CStemN CountUncount Noun) Status=1 Usage=[s,s] RevDat=930118
#Word rule                 (EStemV Verb)
                           Frame=i/advlmsr/advl./pp-obj_/n/thcls/n_advl/n_pcle_/n_inf-objc.
                           VP=[2A,2B,2C,3A,6A,9,15A,15B,25] TI=ti Status=1 Usage=[s,s]
                           RevDat=930118
#Word safety               (NoPlN Uncount Noun) Status=1 Usage=[s,s] RevDat=930118
#Word same                 (NoInflJ Adj) Status=1 Usage=[s,s] RevDat=930118
#Word say                  (CStemV Verb) Sg3=says/saith Pt=said Pp=said
                           Frame=n/thcls/whcls/n_pp_. VP=[6A,9,10,14] TI=ti Status=1
                           Usage=[s,s] RevDat=931124
#Word secretariat          (CStemN Count Noun) Status=1 Usage=[s,s] RevDat=930118
#Word section              (CStemN Count Noun) Status=1 Usage=[s,s] RevDat=930118
#Word Section              (CStemN CountUncount Noun) Status=0 Usage=[s,s] RevDat=950621
#Word secure               (EStemV Verb) Frame=n/n.ppfor-iobj_n/n.ppto-iobj_n/n_pp_.
                           VP=[6A,12B,13B,14] TI=t Status=1 Usage=[s,s] RevDat=950426
#Word secure               (NoInflJ Adj) Status=1 Usage=[s,s] RevDat=930118
#Word security             (YStemN CountUncount Noun) Status=1 Usage=[s,s] RevDat=930118
#Word settlement           (CStemN CountUncount Noun) Status=1 Usage=[s,s] RevDat=930118
#Word several              (NoInflJ Adj) Status=1 Usage=[s,s] RevDat=930118
#Word since                (SubordConj) Status=1 Usage=[s,s] RevDat=950621
#Word since                (Prep) Status=1 Usage=[s,s] RevDat=930118
#Word sit                  (DblStemV Verb) Pt=sat Pp=sat Frame=i/advl./pp-obj_/n/n_pcle_
                           VP=[2A,2C,3A,6A,15B] TI=ti Status=1 Usage=[s,s] RevDat=930118
#Word society              (YStemN CountUncount Noun) Status=1 Usage=[s,s] RevDat=930118
```

```
#Word special               (NoInflJ Adj) Status=1 Usage=[s,s] RevDat=930118
#Word specify               (YStemV Verb) Frame=n/thcls VP=[6A,9] TI=t Status=1 Usage=[s,s]
                            RevDat=930118
#Word state                 (CStemN CountUncount Noun) Status=1 Usage=[s,s] RevDat=930118
#Word state                 (EStemV Verb) Frame=n/thcls VP=[6A,9] TI=t Status=1 Usage=[s,s]
                            RevDat=930118
#Word statute               (CStemN Count Noun) Status=1 Usage=[s,s] RevDat=930118
#Word subject               (CStemN Count Noun) Status=1 Usage=[s,s] RevDat=930118
#Word subject               (CStemV Verb) Frame=iobj_pp-obj_ Prep-2Obj=to VP=[14] TI=t Status=1
                            Usage=[s,s] RevDat=950425
#Word subject               (NoInflJ Adj) Status=1 Usage=[s,s] RevDat=930118
#Word subject_to            (Prep) Status=0 Usage=[s,s] RevDat=950622
#Word such                  (Det) DetMorph=no Status=1 Usage=[s,s] RevDat=950427
#Word supervise             (EStemV Verb) Frame=i/n VP=[2A,6A] TI=ti Status=1
                            Usage=[s,s] RevDat=930118
```

Czech

```
#Word o                 (Prep) PCase=loc/acc
#Word on                 (ON PersPron) Pers=3 Num=sg Gend=ma/mi/n
#Word pa2st              (Impf NE7 Verb) Frame=n/i
#Word pe2ci              (Impf PE1 Verb) Frame=n
#Word pi2sen3            (PS1 Noun)
#Word pod (Prep) PCase=ins/acc
#Word pole               (MR1 Noun)
#Word poma2hat           (Impf DE1 Verb) Frame=n/iobj_inf ObjCase=dat
#Word pomoci             (Perf PE3 Verb) Frame=n/iobj_inf ObjCase=dat
#Word pr3edseda          (PD1 Noun)
#Word pr3i2s3ti2         (JR Adj)
#Word pro                (Prep) PCase=acc
#Word r3ecky2            (MD Adj)
#Word r3idic3            (MZ3 Noun)
#Word ranec              (SJ3 Noun)
#Word rozkos3            (PS2 Noun)
#Word ru4st              (Impf NE8 Verb) Frame=i
#Word rybi2              (JR Adj)
#Word s                  (Prep) PCase=ins/gen
#Word se                 (Noun) <Head wscl>=refl_pcle <Head case>=acc
                         <Loc infl num>=sg/pl <Loc infl variant>=1
                         <synsem loc head clitic>=refl !Form_is_base
#Word si                 (Noun) <Head wscl>=refl_pcle <Head case>=dat
                         <Loc infl num>=sg/pl <Loc infl variant>=1
                         <sysnem loc head clitic>=refl !Form_is_base
#Word si2la              (ZN2 Noun)
#Word slepice            (NS2 Noun)
#Word socha              (ZN4 Noun)
#Word soudce             (SC1 Noun)
#Word staveni2           (ST  Noun)
#Word stroj              (SJ1 Noun)
#Word svine3             (NS3 Noun)
#Word te2ci              (Impf PE1 Verb) Frame=i
#Word tlouci             (Impf PE2 Verb) Frame=n
```

```
#Word tr3a2st           (Impf NE3 Verb) Frame=n/refl ObjCase=ins ReflCase=acc
#Word u                 (Prep) PCase=gen
#Word uc3itel           (MZ2 Noun)
#Word v                 (Prep) PCase=loc/acc
#Word ve2st             (Impf NE5 Verb) Frame=n/i
#Word ve2zt             (Impf NE2 Verb) Frame=n
#Word vejce             (MR  Noun) Pl2=vajec
```

9.5 Sample 5: sample grammar rules

English

```
# Rule VP1

VP -> H C
      !Head_Complement_Rule(H,C)
      !Categories(H,C,vp,vp/v,_,no)
      !Orthography(H,C)
      !CDyn(H,C)
      <H synsem loc head wtype> = content
      !NonLastComp
      !Mod(H,C)
      !NoLeftMod(H)
      !Empty_Bind(H)


# Rule VP2

VP -> H C1 C2
      !Head_Complement_Rule(H,C1,C2)
      !Categories(H,C1,C2,vp,vp/v,_,_,no)
      !Orthography(H,C1,C2)
      !CDyn(H,C1,C2)
      <H synsem loc head wtype> = content
      !NonLastComp
      !Mod(H,C2)
      !NoLeftMod(H)
      !Empty_Bind(H)

# Rule AuxV
AuxV -> H C
      !Aux_Head_Complement_Rule(H,C)
      !Categories(H,C,vp,vp/v,vp/v/adj/adjp,no)
      !Orthography(H,C)
      !CDyn(H,C)
      <H synsem loc head wtype> = function
      <C trace> = no
      !SCListCheck
      !Mod(H,C)
      !NoLeftMod(H)
      !Empty_Bind(H)
```

Czech

```
# Rule VA

VA -> H A
     !Head_Adjunct_Rule(H,A,free_modification)
     !Categories(H,A,v,v,pp/adv,no)
     !Orthography(H,A)
     !CDyn(H,A)
```

```
     !Empty_Bind(H)
     <A synsem nonloc inher slash> = []
     <* synsem loc modified right> = yes
     <* synsem loc modified left>
       = <H synsem loc modified left>
     !LH_Clitics(H,A)


# Rule AV

AV -> A H
     !Head_Adjunct_Rule(H,A,free_modification)
     !Categories(H,A,vbar,v/vbar,pp/adv,no)
     !Orthography(A,H)
     !CDyn(H,A)
     !Empty_Bind(H)
     <A synsem nonloc inher slash> = []
     <* synsem loc modified left> = yes
     <* synsem loc modified right>
       = <H synsem loc modified right>
     !RH_Clitics(A,H)

# Rule CV

CV -> C H
     !Categories(H,C,vbar,v/vbar,_,no)
     !Cz_Head_Complement_Rule(H,C)
     !CDyn(C,H)
     !Orthography(C,H)
     !Empty_Bind(H)
     !Mod(C,H)
     !RH_Clitics(C,H)
## heuristics
     <C synsem loc modified right> = no
     <C synsem loc modified left> = no

# Rule VC

VC -> H C
     !Categories(H,C,v,v,_,no)
     !Cz_Head_Complement_Rule(H,C)
     !CDyn(H,C)
     !Orthography(H,C)
     !Empty_Bind(H)
     !Mod(H,C)
     !LH_Clitics(H,C)
## heuristics
     <C synsem loc modified left> = no
     <C synsem loc modified right> = no
```

9.6 Sample 6: a Czech lexical item

The full structure for the Czech form *pomohly* with two complements — a nominal and an infinitive.

```
<4797> (Lexical)
ortho : [ pomohly ]
cd = [ pomoci ]
synsem : (Synsem)
         loc : (Loc)
               cat = v
               coord = no
               head : (VHead)
                      form = pomohly
                      base = pomoci
                      wtype = content
                      wcl = verb
                      vform = fin
                      negated = no
                      tense = past
                      aspect = perfective
                      clitic = no
                      frame = iobj_inf
               infl : (VInfl)
                      pers = 3
                      num = pl
                      gend = f/mi
                      variant = 1
               subj : [ <4842> loc : cat = n/nbar
                                     comps = []
                                     deep = #4846
                                     head : case = nom
                                     subj = [] ]
               comps = [ <4840> loc : cat = n/nbar
                                      comps = []
                                      deep = #4849
                                      infl : case = dat
                                      subj = []
                         <4841> loc : cat = v/vbar
                                      comps = []
                                      deep = #4852
                                      head : vform = inf
                                      subj = [ <4860> loc : deep -> #4849 ] ]
               lex = yes
               deep : (Deep)
                      gov : (VGov)
                            lexeme = pomoci
                            wcl = verb
                            pform = -NIL-
                            sconj = -NIL-
                            voice = active
                            tense = past
                            modality = -NIL-
                            progressive = no
                            perfective = yes
                            polarity = positive
```

```
                    deps = [ <4844> (Dependent)
                                    fun = actor
                                    val -> #4846
                             <4847> (Dependent)
                                    fun = addressee
                                    val -> #4849
                             <4850> (Dependent)
                                    fun = patient
                                    val -> #4852 ]
         nonloc : (Nonloc)
                  inher : rel = []
                          slash = []
                  to-bind = #4882
dtrs = []
```

9.7 Sample 7: an English lexical item

The full structure for the English form *prevents* with two complements — a nominal and a gerund marked by the preposition.

```
<1> (Lexical)
ortho : [ prevents ]
cd = [ prevent_from ]
trace = no
synsem : (Synsem)
         loc : (Loc)
               cat = v
               coord = no
               head : (VHead)
                      form = prevents
                      base = prevent
                      wtype = content
                      wcl = verb
                      vform = fin
                      inv = no
                      negated = no
                      voice = active
                      tense = present
                      frame = iobj_ping-obj_
                      altform = no
               sc = [ <42> loc : cat = np
                                 deep : <44>
                                        gov : index : (Index)
                                                      pers = 3
                                                      num = sg
                                                      gend = #49
                                 head : case = nom
                                 sc = []
                      <52> loc : cat = np
                                 deep = #61
                                 head : case = acc
                                 sc = []
                      <53> loc : cat = pp
                                 deep : <65>
                                        gov : pform = from
                                 head : vform = presp
                                 sc = [ <72> loc : deep -> #61 ] ]
               deep : (Deep)
                      gov : (VGov)
                            lexeme = prevent_from
                            wcl = verb
                            pform = -NIL-
                            sconj = -NIL-
                            voice = active
                            tense = present
                            modality = -NIL-
                            progressive = no
                            perfective = no
                            polarity = positive
                      deps = [ <54> (Dependent)
```

```
                              fun = actor
                              dir = left
                              val => #44
                          <58> (Dependent)
                              fun = addressee
                              dir = right
                              val -> #61
                         <62> (Dependent)
                              fun = patient
                              dir = right
                              val => #65 ]
nonloc : (Nonloc)
         inher : rel = []
                 slash = []
         to-bind = #86
```

9.8 Sample 8: sample analysis output – auxiliaries

English and Czech examples of clauses with auxiliary or modal verbs which figure as heads in CF rules but not as governors on the deep level. The examples show the derivation tree and the (deep) dependency representation. This and all other results were produced on a Sun ELC work station.

English

```
Script file /home/rosen/elu/nlex/elueng/sample# started 23/10 1995 12:31:58
Created by elu version dated 9/7 1992 09:03:43

(0) Hamish (1) has (2) been (3) referring (4) to (5) you (6) . (7)

NUMBER OF PARSES: 1
Parse # 1

        __________Sentence__________
       /                             \
   ____S_____                         .
  /          \
Hamish   ___AuxV___
        /          \
       has     ___AuxV____
             /            \
            been       __VP1__
                      /       \
                 referring   PP_
                            /   \
                           to you


<5008> (Deep)
gov : (VGov)
      lexeme = refer_to
      wcl = verb
      pform = -NIL-
      sconj = -NIL-
      voice = active
      tense = present
      modality = -NIL-
      progressive = yes
      perfective = yes
      polarity = positive
      conj = #154
deps : [ <371> (Dependent)
               fun = actor
               dir = left
               val : (Deep)
                     gov : (NGov)
                            lexeme = Hamish
                           wcl = noun
                           index : (Index)
                                   pers = 3
                                   num = sg
```

```
                            gend = #21
                    pform = -NIL-
                    definite = yes
                    conj = #25
              deps = []
  <375> (Dependent)
        fun = patient
        dir = right
        val : (Deep)
              gov : (NGov)
                    lexeme = you
                    wcl = pers_pron
                    index : (Index)
                            pers = 2
                            num = pl/sg
                            gend = #519
                    pform = to
                    definite = yes
                    conj = #479
              deps = [] ]

Unresolved Bindings: 5 (21 25 154 479 519)

PARSES 1 | COMPLETIONS 83 | UNIFICATIONS 136426 | CREATIONS 201221 | FAILURES 16945
TOTAL PROCESSOR TIME: 76.75 + 7.38 = 84.13 seconds
TOTAL ELAPSED TIME: about 90 seconds
TOTAL WAITING TIME: about 5 seconds
```

Czech

The sentence *Nebudu moci pást husy.* can be translated as 'I will not be able to graze geese.'

```
Script file /home/rosen/elu/eluec/cz.res.F1 started 29/3 1996 11:44:49
Created by elu version dated 9/7 1992 09:03:43

(0) nebudu (1) moci (2) pa2st (3) husy (4)

NUMBER OF PARSES: 1
Parse # 1

       S
       |
   ___VC____
  /         \
nebudu   __VC___
        /       \
       moci    _VC_
              /    \
            pa2st husy


<743> (Deep)
gov : (VGov)
      lexeme = pa2st
```

```
      wcl = verb
      pform = -NIL-
      sconj = -NIL-
      voice = active
      tense = fut
      modality = moci
      progressive = no
      perfective = no
      polarity = negative
      conj = #131
deps : [ <308> (Dependent)
               fun = actor
               val : gov : base = -nil-
                           index : (Index)
                                   pers = 1
                                   num = sg
                                   gend = #95
         <311> (Dependent)
               fun = patient
               val : (Deep)
                     gov : (NGov)
                           lexeme = husa
                           wcl = noun
                           index : (Index)
                                   pers = 3
                                   num = pl
                                   gend = f
                           pform = -NIL-
                           conj = #384
                     deps = [] ]

Unresolved Bindings: 4 (95 131 383 384)

PARSES 1 | COMPLETIONS 11 | UNIFICATIONS 3089 | CREATIONS 4747 | FAILURES 349
TOTAL PROCESSOR TIME: 2.77 + 0.37 = 3.13 seconds
TOTAL ELAPSED TIME: about 5 seconds
TOTAL WAITING TIME: about 1 second
```

9.9 Sample 9: sample analysis output – unbounded dependency

The example shows the derivation tree and the dependency representation of a sentence *Drogheda she is from.*

```
(0) Drogheda (1) she (2) is (3) from (4) . (5)

NUMBER OF PARSES: 1
Parse # 1

         _______Sentence_______
        /                      \
    ___FH____                  .
   /         \
Drogheda   __S___
          /      \
         she   _VP1__
              /      \
              is   _PP__
                  /     \
                 from Trace
                        |

<630> (Deep)
gov : (VGov)
      lexeme = be
      wcl = verb
      pform = -NIL-
      sconj = -NIL-
      voice = active
      tense = present
      modality = -NIL-
      progressive = no
      perfective = no
      polarity = positive
      conj = #101
deps : [ <116> (Dependent)
               fun = actor
               dir = left
               val : (Deep)
                     gov : (NGov)
                           lexeme = she
                           wcl = pers_pron
                           index : (Index)
                                   pers = 3
                                   num = sg
                                   gend = fem
                           pform = -NIL-
                           definite = yes
                           conj = #62
                     deps = []
         <120> (Dependent)
               fun = patient
               dir = right
               val : (Deep)
                     gov : (NGov)
```

```
                    lexeme = Drogheda
                    wcl = noun
                    index : (Index)
                            pers = 3
                            num = sg
                            gend = #21
                    pform = from
                    definite = yes
                    conj = #25
              deps = [] ]

Unresolved Bindings: 4 (21 25 62 101)

PARSES 1 | COMPLETIONS 23 | UNIFICATIONS 6790 | CREATIONS 9982 | FAILURES 691
TOTAL PROCESSOR TIME: 4.70 + 0.52 = 5.22 seconds
TOTAL ELAPSED TIME: about 7 seconds
TOTAL WAITING TIME: about 1 second
```

9.10 Sample 10: sample analysis output – *tough*-adjectives

The example shows the derivation tree and the dependency representation of a sentence *The fence could not have been easy to refer to.*

```
(0) the (1) fence (2) could (3) not (4) have (5) been (6) easy
(7) to (8) refer (9) to (10) . (11)

NUMBER OF PARSES: 1
Parse # 1
               _____________________Sentence______________________
              /                                                   \
      ________S________                                           .
     /                 \
   DetNP          ____AuxV____
  /     \        /            \
 the  fence    VPcle      ___AuxV___
              /     \    /          \
            could  not  have   ___AuxV____
                              /           \
                             been     _AdjComp1_
                                     /          \
                                    easy     __AuxV__
                                            /        \
                                            to     _VP1_
                                                  /     \
                                                refer  _PP_
                                                      /    \
                                                      to Trace
                                                           |



<7825> (Deep)
gov : (VGov)
      lexeme = easy
      wcl = adj/verb
      pform = -NIL-
      sconj = -NIL-
      voice = active
      tense = past
      modality = can
      progressive = no
      perfective = yes
      polarity = negative
      conj = #110
deps : [ <510> (Dependent)
               fun = actor
               dir = left
               val : (Deep)
                     gov : (VGov)
                           lexeme = refer_to
                           wcl = verb
                           pform = -NIL-
```

```
                sconj = -NIL-
                voice = active
                tense = #567
                modality = -NIL-
                progressive = no
                perfective = no
                polarity = positive
                conj = #572
          deps : [ <691> (Dependent)
                         fun = actor
                         dir = left
                         val = #694
                   <695> (Dependent)
                         fun = patient
                         dir = right
                         val : (Deep)
                               gov : (NGov)
                                     lexeme = fence
                                     wcl = noun
                                     index : (Index)
                                             pers = 3
                                             num = sg
                                             gend = #50
                                     pform = to
                                     definite = yes
                                     conj = #55
                               deps = #5579 ] ]

Unresolved Bindings: 7 (50 55 110 567 572 694 5579)

PARSES 1 | COMPLETIONS 121 | UNIFICATIONS 189594 | CREATIONS 267669 | FAILURES 23041
TOTAL PROCESSOR TIME: 96.45 + 11.08 = 107.53 seconds
TOTAL ELAPSED TIME: about 129 seconds
TOTAL WAITING TIME: about 21 seconds
```

9.11 Sample 11: sample analysis output – object control *equi* construction

The example shows the derivation tree and the dependency representation of a sentence *Honzovi jsme hnůj kydat nepomohli* ‘We didn’t help John to muck manure out’.

```
(0) Honzovi (1) jsme (2) hnu4j (3) kydat (4) nepomohli (5)

NUMBER OF PARSES: 1
Parse # 1

         S
         |
    _____CV_____
   /            \
Honzovi   ____AuxV_____
         /             \
        jsme       ____CV____
                  /          \
                _CV__    nepomohli
               /     \
             hnu4j kydat


<985> (Deep)
gov : (VGov)
      lexeme = pomoci
      wcl = verb
      pform = -NIL-
      sconj = -NIL-
      voice = active
      tense = past
      modality = -NIL-
      progressive = no
      perfective = yes
      polarity = negative
      conj = #120
deps : [ <429> (Dependent)
               fun = actor
               val : gov : base = -nil-
                           index : (Index)
                                   pers = 1
                                   num = pl
                                   gend = ma
         <432> (Dependent)
               fun = addressee
               val : <48> (Deep)
                     gov : (NGov)
                           lexeme = Honza
                           wcl = noun
                           index : (Index)
                                   pers = 3
                                   num = sg
                                   gend = ma
                           pform = -NIL-
```

```
                        definite = #42
                        conj = #43
                  deps = []
      <435> (Dependent)
            fun = patient
            val : (Deep)
                  gov : (VGov)
                        lexeme = kydat
                        wcl = verb
                        pform = -NIL-
                        sconj = -NIL-
                        voice = active
                        tense = x
                        modality = -NIL-
                        progressive = no
                        perfective = no
                        polarity = positive
                        conj = #344
                  deps = [ <348> (Dependent)
                               fun = actor
                               val => #48
                           <351> (Dependent)
                               fun = patient
                               val : (Deep)
                                     gov : (NGov)
                                           lexeme = hnu4j
                                           wcl = noun
                                           index : (Index)
                                                   pers = 3
                                                   num = sg
                                                   gend = mi
                                           pform = -NIL-
                                           definite = #274
                                           conj = #275
                                     deps = [] ] ]

Unresolved Bindings: 6 (42 43 120 274 275 344)
Cross references: 1 (48)

PARSES 1 | COMPLETIONS 9 | UNIFICATIONS 3341 | CREATIONS 5344 | FAILURES 401
TOTAL PROCESSOR TIME: 3.20 + 0.27 = 3.47 seconds
TOTAL ELAPSED TIME: about 5 seconds
TOTAL WAITING TIME: about 1 second
```

9.12 Sample 12: derivation tree in Czech

The example shows derivation trees for six word order variants of a sentence *Honza dal Marii balíček* ('John gave Mary a/the packet'). Deep level representation is the same in all cases. The difference is captured by different orderings inside the CD list.[25]

```
<129> (Deep)
gov : (VGov)
      lexeme = da2t
      wcl = verb
      pform = -NIL-
      sconj = -NIL-
      voice = active
      tense = past
      modality = -NIL-
      progressive = no
      perfective = yes
      polarity = positive
deps = [ <120> (Dependent)
               fun = actor
               val : (Deep)
                     gov : (NGov)
                           lexeme = Honza
                           wcl = noun
                           index : (Index)
                                   pers = 3
                                   num = sg
                                   gend = ma
                           pform = -NIL-
                     deps = []
         <123> (Dependent)
               fun = addressee
               val : (Deep)
                     gov : (NGov)
                           lexeme = Marie
                           wcl = noun
                           index : (Index)
                                   pers = 3
                                   num = sg
                                   gend = f
                           pform = -NIL-
                     deps = []
         <126> (Dependent)
               fun = patient
               val : (Deep)
                     gov : (NGov)
                           lexeme = bali2c3ek
                           wcl = noun
                           index : (Index)
                                   pers = 3
                                   num = sg
                                   gend = mi
```

[25] The attribute CD is at the same (topmost) level within the structure as the attribute SYNSEM (see Section 5.3). E.g., the CD list for the first variant *Honza dal Marii balíček* is `cd : [ Honza,da2t,Marie,bali2c3ek ]` .

```
                    pform = -NIL-
              deps = [] ]




        S
        |
    ____CV_____
   /           \
Honza      ____VC____
          /          \
        _VC_    bali2c3ek
       /    \
      dal Marii

          S
          |
     _____CV_____
    /            \
bali2c3ek      ___VC___
              /        \
            _VC_    Marii
           /    \
          dal Honza

        S
        |
    _____CV_____
   /            \
Honza       ____CV____
           /          \
      bali2c3ek     _VC__
                   /     \
                 dal   Marii

          S
          |
     ____CV_____
    /           \
bali2c3ek    ___CV___
            /        \
          Marii     _VC__
                   /     \
                 dal   Honza

          S
          |
     ____CV_____
    /           \
bali2c3ek    ___CV___
            /        \
          Honza     __CV_
```

```
                 /     \
               Marii    dal

                 S
                 |
             ____VC___
            /         \
       ___VC___    bali2c3ek
      /        \
    _VC_    Marii
   /    \
dal Honza
```

9.13 Sample 13: clitics in Czech

The example shows analysis results with derivation trees for several sentences with clitics, some of them being correctly refused.

In some sentences the strong version 'jemu' for the dative case of the pronoun *on* 'he' occurs. Unlike its weak counterpart, 'jemu' is not a clitic.

```
(0) na (1) poli (2) jsem (3) mu (4) ho (5) nedal (6)

NUMBER OF PARSES: 1
Parse # 1

       S
       |
   ___AV____
  /         \
 PP_      AuxV_
/   \    /     \
na poli jsem  _CV_
             /    \
             mu  _CV_
                /    \
                ho nedal

PARSES 1 | COMPLETIONS 11 | UNIFICATIONS 3152 | CREATIONS 4932 | FAILURES 350
TOTAL PROCESSOR TIME: 3.10 + 0.25 = 3.35 seconds
TOTAL ELAPSED TIME: about 6 seconds
TOTAL WAITING TIME: about 2 seconds

(0) nedal (1) jsem (2) mu (3) ho (4) v (5) divadle (6)

NUMBER OF PARSES: 1
Parse # 1

               S
               |
          ___VA____
         /         \
       __VC__    _PP__
      /      \  /     \
    __VC__   ho v  divadle
   /      \
   VAux    mu
  /    \
nedal jsem

PARSES 1 | COMPLETIONS 7 | UNIFICATIONS 2558 | CREATIONS 4095 | FAILURES 271
TOTAL PROCESSOR TIME: 2.80 + 0.22 = 3.02 seconds
TOTAL ELAPSED TIME: about 5 seconds
TOTAL WAITING TIME: about 1 second

(0) nedal (1) jsem (2) jemu (3) ho (4) v (5) divadle (6)

NUMBER OF PARSES: 0
```

```
PARSES 0 | COMPLETIONS 3 | UNIFICATIONS 2396 | CREATIONS 3860 | FAILURES 257
TOTAL PROCESSOR TIME: 2.65 + 0.20 = 2.85 seconds
TOTAL ELAPSED TIME: about 4 seconds
TOTAL WAITING TIME: about 1 second

(0) jemu (1) jsem (2) ho (3) nedal (4)

NUMBER OF PARSES: 1
Parse # 1

     S
     |
  __CV___
 /       \
jemu   AuxV_
      /     \
     jsem  _CV_
          /    \
          ho nedal

PARSES 1 | COMPLETIONS 5 | UNIFICATIONS 2096 | CREATIONS 3225 | FAILURES 263
TOTAL PROCESSOR TIME: 2.23 + 0.10 = 2.33 seconds
TOTAL ELAPSED TIME: about 4 seconds
TOTAL WAITING TIME: about 1 second

(0) na (1) poli (2) jsem (3) se (4) tr3a2sl (5)

NUMBER OF PARSES: 1
Parse # 1

       S
       |
   ___AV____
  /         \
 PP_      _AuxV_
/   \    /      \
na poli jsem  _CV__
             /     \
             se tr3a2sl

PARSES 1 | COMPLETIONS 10 | UNIFICATIONS 2835 | CREATIONS 4505 | FAILURES 309
TOTAL PROCESSOR TIME: 2.77 + 0.23 = 3.00 seconds
TOTAL ELAPSED TIME: about 5 seconds
TOTAL WAITING TIME: about 1 second

(0) na (1) poli (2) se (3) jsem (4) tr3a2sl (5)

NUMBER OF PARSES: 0
PARSES 0 | COMPLETIONS 3 | UNIFICATIONS 2292 | CREATIONS 3732 | FAILURES 258
TOTAL PROCESSOR TIME: 2.37 + 0.23 = 2.60 seconds
TOTAL ELAPSED TIME: about 5 seconds
TOTAL WAITING TIME: about 2 seconds
```