# Machine translation system as a linguistic testbed*

Outline of a talk presented at the International Center Workshop
on Computational Lingistics at the University of Tübingen, 16–27 September 1996

Alexandr Rosen
Charles University, Prague
alexandr.rosen@ff.cuni.cz



## 1 Introduction

### Main features of the system:

- bidirectional, Czech and English
- declarative, constraint-based specification of linguistic data
- not an industrial prototype, ideally a "linguistic playground"
- under construction (unbalanced development of individual parts)
- the effort is justified only if you believe that your linguistic hypotheses can be expressed and verified in a constraint-based formalism

### Why MT and not just a parser?
### Why is MT interesting for a linguistic theory?

- language-internal facts are difficult to present as universals
- strong motivation for searching common features across languages
- representation of meaning is testable by (backward) translation

*The work has been supported by Research Support Scheme of the Open Society Institute (project No. 340/93) and in part by Grant Agency of the Czech Republic (within project No. 405/93/2173).

- and also: MT embraces most problems of NLP
- ... thus supports compatibility across theories, formalisms, processing tools

### An MT system conceived as a “linguistic playground” should be:

- able to reflect theoretical assumption as directly as possible
- highly modular (easy to modify, extend, parameterize, cut- and-paste along many dimensions)
- able to incorporate language-universal and language-specific features wherever appropriate

### Current state of the playground

– properties shared by the Czech and English parts:

**theory** $T$**:**

- constraint-based framework for the description of surface syntax as a relation between surface string and its representation
- FGD-like dependency representation (Functional Generative Description — see Sgall, Hajičová & Panevová 1996)

**grammar and lexicon** $G$**:** (partial) descriptions of English and Czech by means of formalism $F$ within theory $T$, employing similar concepts and solutions

**formalism** $F$**:** à la PATR-II with some extensions

**implementation** $I$**:** compiler of $G$, extended Earley parser (prediction through restrictors), head-driven generator (Shieber et al., 1989), dictionary look-up module, transfer component

## 2 Formalism

ELU Environnement linguistique d'unification, ISSCO, Geneva (Dominique Estival *ELU User Manual*, 1990)

### Extensions to PATR-II:

**lists and trees** are built-in data types

**types** can be defined for feature structures

**disjunction and negation** can be applied to atomic constants

**macros** (templates) allow for parameters, multiple and recursive definitions

**lexicon** as a hierarchy of classes with controlled multiple inheritance and defaults

**transfer** declarative, unification-based, may be reversible, locality is preserved and recursion controlled by general means

## Good things about ELU:

- parsing, generation and transfer within a uniform environment
- a powerful lexical component

## Bad things about ELU:

- no longer a state-of-the-art tool, no longer developed
- types are optional (completeness in generation), not used in compilation, no type hierarchy
- unsatisfactory interaction of the lexical component with multiple macro definitions and disjunctions
- no lexical rules

## Both E & Cz components consist of 5 modules:

1. general specifications
2. definition of macros
3. definition of lexical classes
4. dictionary of lemmas
5. grammar rules

**general specifications** – declarations of types and restrictors, ...

```
Phrase = (ortho,cd,trace,synsem,dtrs)
Lexical = (ortho,cd,trace,synsem)
Synsem = (loc,nonloc)
Loc = (cat,coord,head,sc,modified,bare,deep)
Nonloc = (inher,to-bind)
Deep = (gov,deps)
Dependent = (fun,dir,val)
```

**definition of macros** – to express general principles or recurrent constraints in lexicon and grammar

```
HFP(H)
  <* synsem loc head> = <H synsem loc head>

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

**definition of lexical classes**

- the classes are ordered according to specificity in a hierarchy
- for every class, a list of superclasses can be defined
- for every class, there can be:
  - a main (default) set of contraints

– multiple disjunctive sets of constraints

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

# Class  VerbInfl (Word)
  |
  Form = Inf  <vform> = inf
  |
  Form = Sg1  <vform> = fin  <agr num> = sg  <agr pers> = 1
  |
  Form = Sg2  <vform> = fin  <agr num> = sg  <agr pers> = 2
  ...

# Class  Verb (VerbInfl)
  <general specification of an intransitive verb>
  |
  Paradigm = nese
  <specification of the inflection paradigm 'nese'>
  |
  Paradigm = bere
  <specification of the inflection paradigm 'bere'>
  |
  ...

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

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

**dictionary of lemmas**

```
receipt     (CStemN CountUncount Noun)
receive     (EStemV Verb) Frame=n
reciprocity (NoPlN Uncount Noun)
refer       (DblStemV Verb) Frame=pp-obj_/n_pp_.
            /n_pcle_ Prep-Obj=to Pcle-Obj=back
regard      (CStemN CountUncount Noun)
regard      (CStemV Verb) Frame=n/n_pp_./n_ping-objc_
            /n_pp-objc_ Prep-Objc=as

pod         (Prep) PCase=ins/acc
pole        (MR1 Noun)
pomáhat     (Impf DE1 Verb) Frame=n/iobj_inf
            ObjCase=dat
pomoci      (Perf PE3 Verb) Frame=n/iobj_inf
            ObjCase=dat
předseda    (PD1 Noun)
příští      (JR Adj)
```

**grammar rules**

```
# 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
```

# 3 Lexicon

## Schema of a lexical item

$$\begin{bmatrix} \text{ORTHO} & \langle ... \rangle \\ \text{CD} & \langle ... \rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \mathit{cat} \\ \text{HEAD} & \begin{bmatrix} ... \end{bmatrix}_{\mathit{head}} \\ \text{SC} & \langle ... \rangle \\ \text{DEEP} & \begin{bmatrix} \text{GOV} & \begin{bmatrix} ... \end{bmatrix}_{\mathit{gov}} \\ \text{DEPS} & \langle ... \rangle \end{bmatrix}_{\mathit{deep}} \end{bmatrix}_{\mathit{local}} \\ \text{NONLOC} & \begin{bmatrix} \text{INHER} & \begin{bmatrix} \text{REL} & \langle ... \rangle \\ \text{SLASH} & \langle ... \rangle \end{bmatrix}_{\mathit{inher}} \\ \text{TO-BIND} & \begin{bmatrix} \text{REL} & \langle ... \rangle \\ \text{SLASH} & \langle ... \rangle \end{bmatrix}_{\mathit{to\text{-}bind}} \end{bmatrix}_{\mathit{nonloc}} \end{bmatrix}_{\mathit{synsem}} \end{bmatrix}_{\mathit{lexical}}$$

## Value of the attribute DEEP for *she*

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

### Value of the attribute DEEP for *expires*

$$
\begin{bmatrix}
\textsc{gov} & \begin{bmatrix}
\textsc{lexeme} & \text{expires} \\
\textsc{wcl} & \text{verb} \\
\textsc{pform} & \text{-nil-} \\
\textsc{sconj} & \text{-nil-} \\
\textsc{voice} & \text{active} \\
\textsc{tense} & \text{present} \\
\textsc{modality} & \text{-nil-} \\
\textsc{progressive} & \text{no} \\
\textsc{perfective} & \text{no} \\
\textsc{polarity} & \text{positive}
\end{bmatrix}_{ngov} \\
\textsc{deps} & \left\langle \begin{bmatrix}
\textsc{fun} & \text{actor} \\
\textsc{dir} & \text{left} \\
\textsc{val} & \mid \textsc{gov} \mid \textsc{index} \begin{bmatrix}
\textsc{pers} & 3 \\
\textsc{num} & \text{sg} \\
\textsc{gend} & \text{-}
\end{bmatrix}_{index}
\end{bmatrix}_{dependent} \right\rangle
\end{bmatrix}_{deep}
$$

**Part of a structure representing the Czech lexical item *pomohly* '(they$_{fem/masc.inanim}$) helped'**

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

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

## Sample definition of some complementation classes

```
# 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)
```

## Sample definition of some morphological paradigms

```
# 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')
```

# 4 Grammar

— a combination of:

- unification-based monostratal syntactic model with a CF backbone (HPSG-inspired)
- dependency-based 'underlying' representation of 'linguistic meaning' (FGD-like)

The grammar describes the relationship between the surface string and its representation via CF rules with constraints.

The CF rules and their dependency interpretation can be viewed as two levels of description: SL and DL; cases where this distinction is used:

- function words not as nodes at DL
- governor–dependent relations explicitly labelled by function names
- SL head need not correspond to DL governor

```
   ____S_____                             S
  /          \                            |
Hamish   ___AuxV___                  ___VC____
        /          \                 /         \
       has    ___AuxV____          nebudu   __VC___
             /           \                  /       \
            been      __VP1__             moci    _VC_
                     /       \                    /    \
                  referring PP_                pást  husy
                            /   \
                            to you
   __refer~to__                      ___pást___
  /            \                     /          \
Hamish         you                 já         husa
```

- long-distance dependencies as DL projective trees

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

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

- A single constituent can be shared (co-indexed) on DL by two or more nodes.
  *Honzovi jsme hnůj kydat nepomohli*
  John$_{dat}$ AUX$_{1pl}$ manure to muck help$_{past,neg}$

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

- SL categories $\neq$ DL categories (deverbatives)
- the number of dependents of a governor at DL $\neq$ the total number of non-head sisters to the head at SL at all projection levels
- SL complementation requirements $\neq$ corresponding DL valency frames;
  at SL: SUBCAT list, SPEC feature
  at DL: DEPS list, no function words, no valency slots for particles, a missing complement of raising verbs

## Types of grammar rules and their DL intepretation:

**head_only_rule:** DL feature due to absence of some lexical item(s)

**head_complement_rule:** governor + its complement(s)

**head_adjunct_rule:** optional free modification

**head_marker_rule:** DL feature due to function word

**aux_head_complement_rule:** DL feature due to auxiliary

**coord_rule:** recursive specification of a coordinated structure

**trace_rule:** detection of a potential non-projective dependency

**head_filler_rule:** non-projective dependent

## Principles of grammar

**HFP:** head feature principle

**HDeep:** deep head feature principle

**HDtrs:** daughters principle

**SCP:** subcategorisation principle

**SpecP:** specifier principle

**NlocFP:** non-local feature principle

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

### Other features:

- surface vs. 'deep' word order distinction via an attribute CD for *communicative dynamism*
  The fence was difficult to break.

```
difficult
      \
     break
      /
   fence
```

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

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

- heuristic constraints: *right association*, ...

## 5 Specifics of English

### Lexicon:

20000 entries compiled, 100000 entries available;

### Grammar:

26 rules

### Phenomena covered:

**nominal group:**

- (pre)determination, prepositional marking
- modified by adj's, p'ples, PPs, rel. clauses

**verbal group:**

- tenses, passive, modality, negation, inversion
- complements, adjuncts within the verb's head domain
- *raising* and *equi* embedded predicates
- *unbounded dependencies*: topicalisations, (embedded) relative expressions (NPs, PPs, NPs extracted from PPs), *tough/easy*

**coordination of phrasal and lexical categories**

**other phenomena related to unbounded dependencies:**

- Complex NP Constraint (a relative clause cannot include a trace) (but: *Moby Dick*$_1$*, I was the only person that hadn't read* $\_\_{1}$.)

- Trace Principle (every trace must be subcategorized, but not as a subject) (*$How_1$ do you think the building is $\_\_1$ tall?*)
- Subject Condition (a trace may occur inside a subject only when it is subcategorized by another item in the SC list. (*$Who_1$ did my talking to $\_\_1$ bother $\_\_1$?*)
- 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.*)

# 6 Specifics of Czech

rich morphology, 'free' word order
the question: can the same processing tool, formalism and theory be used as for English?

## Word-order-related issues:

- traditional or merely **formal** constituents?
- flat or **binary** branching structure?
- left or right branching or **both**?
- how to impose constraints on word order? DIFFICULT!
- a special mapping between surface string and leaves? NO

## Morphology-related issues:

description of inflectional paradigms via default inheritance invites a problem of multiplicity of structures, an example:

1. in a paradigm, an ending stands for more cases
2. across more paradigms, the same ending stands for a case
3. in a paradigm, more than one ending is possible for a case
4. the present formalism handles only atomic disjunction

```
Class (Noun)     ...
                 | Form = GenForm  case = gen
                 | Form = AccForm  case = acc
                 | ...
Class AB (Noun) ... GenForm = Base && a  ...
Class A  (AB)   ... AccForm = Base && a  ...
Class B  (AB)   ... AccForm = Base  ...

Class (Noun)     ...
Class A  (Noun) | ...
                 | Form = Base && a  case = gen/acc
                 | ...
Class B  (Noun) | ...
                 | Form = Base && a  case = gen
                 | Form = Base       case = acc
                 | ...
```

## Lexicon:

- the strategy: verification of non-lexeme-specific parts (definition of inflectional paradigms, lexical classes)
- hand-coded
- 100+ lemmas
- integrated morphological component

## Grammar (the core):

- strategy: methods used for English preserved where useful; stress on differences in E/CZ syntax to verify viability
- 1 unary rule (S)
- 3 binary fixed-precedence rules
  (Prep N, Adj N, N PP)
- 3 pairs of head-initial/final rules
  (V C, V A, V Aux)

**head_only_rule:** S

**S:** s $\rightarrow$ **v/vbar**
- [$_s$ [$_{vbar}$ Honza je na poli] ]
- [$_s$ [$_v$ pomáhal jsem Honzovi pást husy] ]

**head_complement_rule:** VC, CV

**VC:** v $\rightarrow$ **v** $X$
- Honza [$_v$ [$_v$ dal Marii] pusu ] ]
- [$_v$ [$_v$ nemohl jsem] přijít ] ]

**CV:** vbar $\rightarrow$ $X$ **v/vbar**
- [$_{vbar}$ Honza [$_v$ dal Marii pusu ] ]

**head_adjunct_rule:** VA, AV, NA, AdjN

**VA:** v $\rightarrow$ **v** pp/adv
- [$_v$ [$_v$ potkal jsem ji ] [$_{pp}$ v divadle ] ]

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

**NA:** n $\rightarrow$ **n** pp
- [$_n$ [$_n$ slepice ] [$_{pp}$ s kuřaty ] ]

**AdjN:** nbar $\rightarrow$ adj **n/nbar**
- [$_{nbar}$ velká [$_{nbar}$ rybí [$_n$ kost od Marie ] ] ]

**head_marker_rule:** VAux, AuxV, PP

**VAux:** v $\rightarrow$ **v** vaux
- [$_v$ [$_v$ nemohla ] [$_{vaux}$ jsem ] ] mu pomoci

**AuxV:** vbar $\rightarrow$ vaux **v/vbar**
- v lese [$_{vbar}$ [$_{vaux}$ jsem ] [$_{vbar}$ se třásl strachy ] ]

**PP:** pp $\rightarrow$ prep **n/nbar**
- [$_{pp}$ [$_{prep}$ na ] [$_{nbar}$ vysokém stavení ] ]

### A deep level representation of a sentence ***Honza dal Marii balíček*** — 'John gave Mary a packet.'

(and of all other 5 possible orderings in the sentence — note that the attribute CD for *communicative dynamism* does not appear within the structure)

```
<129> (Deep)
gov : (VGov)
      lexeme = dát
      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 =  balíček
                           wcl = noun
                           index : (Index)
                                   pers = 3
                                   num = sg
                                   gend = mi
                           pform = -NIL-
                     deps = [] ]
```

**Derivation trees for the 6 word order variants of the sentence *Honza dal Marii balíček* — 'John gave Mary a packet'.**

```
        S                             S
        |                             |
   ____CV_____                   _____CV_____
  /           \                 /            \
Honza      ____VC____        balíček      ___VC___
          /          \                   /        \
        _VC_     balíček               _VC_      Marii
       /    \                         /    \
      dal Marii                      dal Honza


        S                             S
        |                             |
   _____CV_____                   ____CV_____
  /            \                 /           \
Honza       ____CV____        balíček     ___CV___
           /          \                  /        \
       balíček      _VC__              Marii     _VC__
                   /     \                      /     \
                  dal   Marii                  dal   Honza


          S                                   S
          |                                   |
     ____CV_____                         ____VC___
    /           \                       /         \
 balíček      ___CV___             ___VC___      balíček
             /        \           /        \
          Honza      __CV_      _VC_      Marii
                    /     \    /    \
                  Marii    dal dal Honza
```

## Clitics

- past tense auxiliary *jsem, ...*
- reflexive particles *se, si*
- weak personal pronouns in dative, accusative *mu, ho, ...*
- Wackernagel position, fixed order in clitic cluster, optional climbing

The following sample derivation trees exhibit the structures used for clauses including clitical elements. Cases where a non-clitic intervenes in a clitic cluster or the order of elements within the cluster is incorrect are correctly refused.

```
       S                                   S
       |                                   |
   ___AV____                           ___VA____
  /         \                         /         \
 PP_      AuxV_                     __VC__    _PP__
/   \    /     \                   /      \  /     \
na poli jsem  _CV_               __VC__   ho v  divadle
             /    \             /      \
             mu  _CV_          VAux    mu
                /    \        /    \
                ho nedal    nedal jsem

* nedal jsem jemu ho v divadle
     S                        S
     |                        |
  __CV___                 ___AV____
 /       \               /         \
jemu   AuxV_            PP_      _AuxV_
      /     \          /   \    /      \
     jsem  _CV_       na  poli jsem   _CV__
          /    \                     /     \
          ho nedal                  se    třásl

* na poli se jsem třásl
```

# 7 Transfer

- declarative (bidirectional)
- lexical and structural rules
- extension of unification preserving locality
- control of recursion via
  - subsumption between source object and rules
  - specificity ordering among transfer rules

```
:TA: Toma2s3 Tom
:TA: Zuzana Sue

:T:  general_gov_deps
:L1: <gov> = GovCz    <deps> = DepsCz
:L2: <gov> = GovEng   <deps> = DepsEng
:X:  GovCz = GovEng   DepsCz = DepsEng

:T:  li2bit_se_like
:L1: <gov lexeme> = li2bit_se
     <deps> = [Arg1Cz Arg2Cz RCz]
:L2: <gov lexeme> = like
     <deps> = [Arg1Eng Arg2Eng RCz]
:X:  Arg1Cz = Arg2Eng
     Arg2Cz = Arg1Eng
     RCz = REng
```

# 8 Conclusion

## Results:

- an English lexical data base
- a common approach, combining unification-based CFG with dependency representation, verified for two typologically different languages in parsing and generation
- a partial model of Czech morphology within a hierarchically structured lexical component
- solutions to various phenomena implemented and tested
- the model is open to further development as a kind of linguistic testbed

## Plans:

- lexical and structural transfer
- extension of Czech lexicon and morphology
- reimplementation in a more sophisticated framework
- augmentation by a stochastic model?
- other languages?