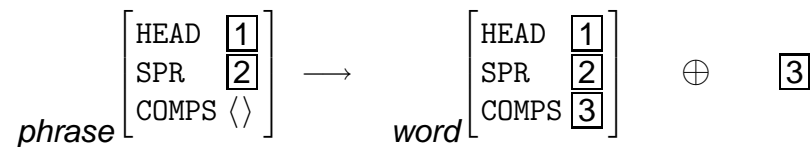


Computational Linguistics (INF2820 — Reflections)



Stephan Oepen

Universitetet i Oslo

oe@ifi.uio.no

Dative Shift: A Productive Process

$$\{hand_1, give_1, send_1, \dots\} \left[\begin{array}{l} \text{HEAD } \mathit{verb} \\ \text{SPR } \langle \dots \rangle \\ \text{COMPS } \left\langle \begin{array}{l} \text{HEAD } \mathit{noun} \\ \text{SPR } \langle \rangle \\ \text{COMPS } \langle \rangle \end{array} \right\rangle, \\ \mathit{phrase} \left[\begin{array}{l} \text{HEAD } \mathit{noun} \\ \text{SPR } \langle \rangle \\ \text{COMPS } \langle \rangle \end{array} \right] \end{array} \right]$$

$$\{hand_2, give_2, send_2, \dots\} \left[\begin{array}{l} \text{HEAD } \mathit{verb} \\ \text{SPR } \langle \dots \rangle \\ \text{COMPS } \left\langle \begin{array}{l} \text{HEAD } \mathit{noun} \\ \text{SPR } \langle \rangle \\ \text{COMPS } \langle \rangle \end{array} \right\rangle, \\ \mathit{phrase} \left[\begin{array}{l} \text{HEAD } \mathit{prep} \left[\text{PFORM } \mathit{to} \right] \\ \text{SPR } \langle \rangle \\ \text{COMPS } \langle \rangle \end{array} \right] \end{array} \right]$$



Dative Shift: A Sketch of a Productive Lexical Rule



Orthographic Variation: Inflectional Rules

```
%(letter-set (!s abcdefghijklmnopqrstuvwxyz))
```

```
noun-non-3sing_irule :=  
%suffix (!s !ss) (!ss !ssses) (ss sses)  
non-3sing-word &  
[ ARGS < noun-lxm > ].
```

```
noun-3sing_irule :=  
3sing-word &  
[ ORTH #1,  
  ARGS < noun-lxm & [ ORTH #1 ] > ].
```

dog

|

dogs

bus

|

busses

pass

|

passes



Exercise: Implementing the Head–Complement Rule

$$\textit{phrase} \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{SPR} & \boxed{2} \\ \text{COMPS} & \langle \rangle \end{bmatrix} \longrightarrow \textit{word} \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{SPR} & \boxed{2} \\ \text{COMPS} & \boxed{3} \end{bmatrix} \oplus \boxed{3}$$

Issues in the Linguistic Knowledge Builder

- The Sag, Wasow, & Bender (2003) rule has a *variable-arity* RHS;
 - the LKB chart parser (conventionally) requires *fixed-arity* rules;
- need to recast head – complement rule as a *recursive* binary rule.



Our Grammars: Table of Contents

Type Description Language (TDL)

- `types.tdl` type definitions: hierarchy of grammatical knowledge;
- `lexicon.tdl` instances of (lexical) types plus orthography;
- `rules.tdl` instances of construction types; used by the parser;
- `lrules.tdl` lexical rules, applied before non-lexical rules;
- `irules.tdl` lexical rules that require orthographemic variation;
- `roots.tdl` grammar start symbol(s): 'selection' of final results.

Auxiliary Files (Grammar Configuration for LKB)

- `labels.tdl` TFS templates abbreviating node labels in trees;
- `globals.lsp`, `user-fns.lsp` parameters and interface functions;
- `mrsglobals.lsp` MRS parameters (path to semantics et al.)

