Formal Syntax and Grammar Engineering

Stephan Oepen

Universitetet i Oslo & CSLI Stanford

oe@csli.stanford.edu

Lilja Øvrelid

Göteborgs Universitet

lilja.ovrelid@svenska.gu.se

http://www.delph-in.net/courses/04/fs/

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.).

Lexical Variation: Lexical Rules

- Dative shift, passivization, et al. are systematic processes in the lexicon;
- *lexical rules* are unary grammar rules operating 'within the lexicon';
- take as input a lexical sign (*expression*) and output a derived lexical sign.

Rough Approximation of Passive Lexical Rule

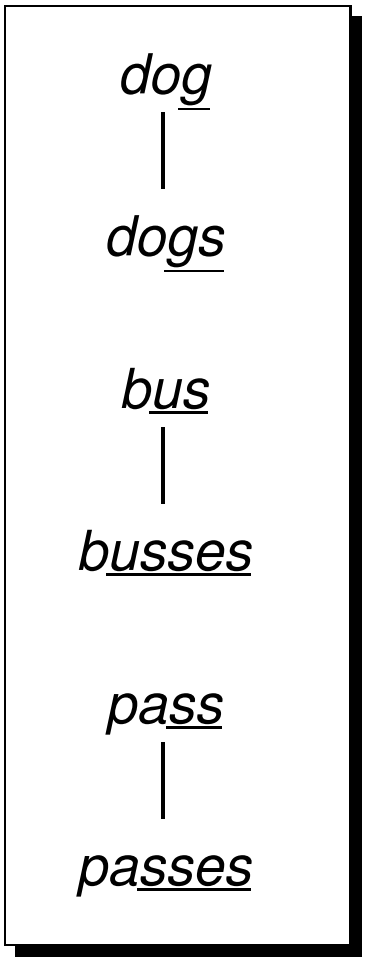
$$\begin{bmatrix} \textsc{head} & \boxed{1} \\ \textsc{spr} & \langle \boxed{2} \rangle \\ \textsc{comps} & \boxed{3} \\ \textsc{args} & \left\langle \begin{bmatrix} \textsc{head} & \boxed{1}\, \textit{verb} \\ \textsc{spr} & \left\langle \textit{phrase} \begin{bmatrix} \textsc{head} & \textit{noun} \end{bmatrix} \right\rangle \\ \textsc{comps} & \begin{bmatrix} \textsc{first} & \boxed{2}\, \textit{phrase} \begin{bmatrix} \textsc{head} & \textit{noun} \end{bmatrix} \\ \textsc{rest} & \boxed{3} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Orthographemic Variation: 'Inflectional' Rules

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

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

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



Structured Categories in a Unification Grammar

- All (constituent) categories in the grammar are typed feature structures;
- specific TFS configurations may correspond to ‘traditional’ categories;
- labels like ‘S’ or ‘NP’ are mere abbreviations, not elements of the theory.

$$\textit{word}\begin{bmatrix}\texttt{HEAD} & \textit{verb}\\ \texttt{SPR} & \langle [\,] \rangle \\ \texttt{COMPS} & \langle\rangle\end{bmatrix}$$

$$\textit{phrase}\begin{bmatrix}\texttt{HEAD} & \textit{verb}\\ \texttt{SPR} & \langle\rangle \\ \texttt{COMPS} & \langle\rangle\end{bmatrix}$$

$$\textit{phrase}\begin{bmatrix}\texttt{HEAD} & \textit{verb}\\ \texttt{SPR} & \langle [\,] \rangle \\ \texttt{COMPS} & \langle\rangle\end{bmatrix}$$

‘N’

‘lexical’

‘S’

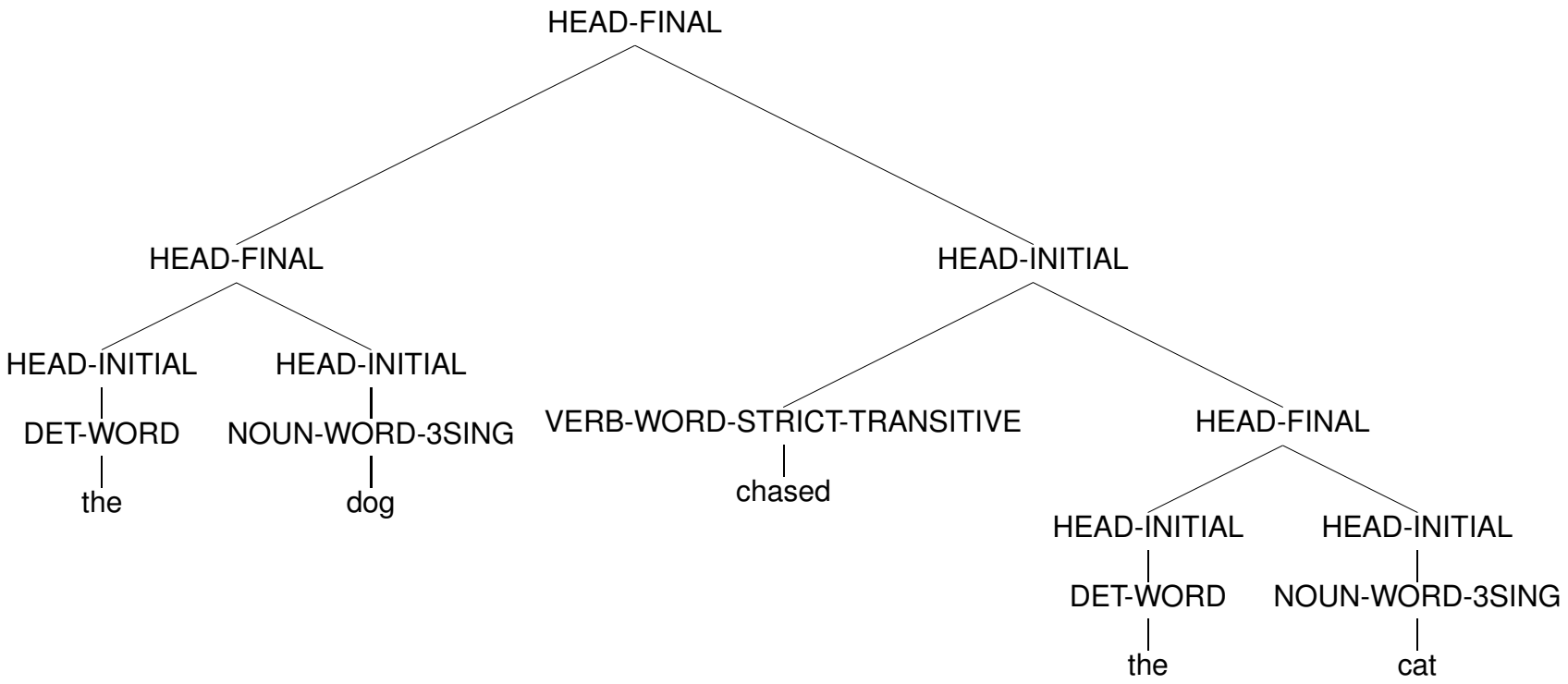
‘intermediate’

‘VP’

‘maximal’

Parse Trees and Node Labeling

- Derivation trees are constructed from lexical items and grammar rules;
- node labels (so far) are instance names from ‘lexicon.tdl’ and ‘rules.tdl’.



Decorating our Trees: Abbreviatory Node Labels

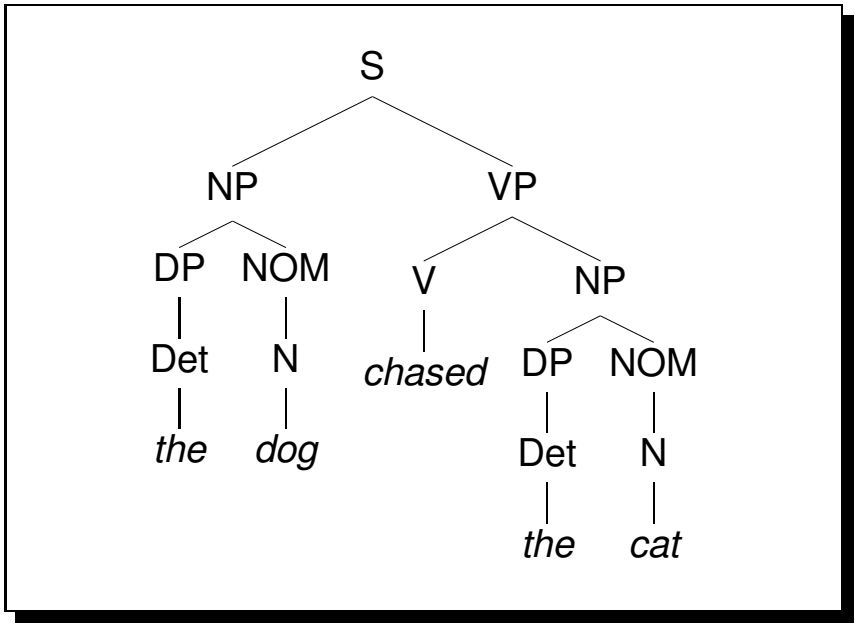
‘labels.tdl’

```
s-label := label &
[ HEAD verb, SPR <>, COMPS <>,
  LABEL-NAME "S" ].

vp-label := label &
[ HEAD verb, SPR < [] >, COMPS <>,
  LABEL-NAME "VP" ].

v-label := label &
[ HEAD verb,
  LABEL-NAME "V" ].

np-label := label &
[ HEAD noun, SPR <>, COMPS <>,
  LABEL-NAME "NP" ].
```



Types vs. Named Feature Structures ('Instances')

'types.tdl'

```
verb-word := word &
[ HEAD verb,
  SPR < phrase & [ HEAD noun,
                   SPR < >,
                   COMPS < > ] > ].

verb-word-3sing := verb-word &
[ SPR < [ HEAD [ AGR 3sing ] ] > ].
```

'lexicon.tdl'

```
barks := verb-word-3sing &
[ ORTH "barks" ].
```