Typed Feature Structures: Notational Variants

$$
{}_{\textit{word}}\begin{bmatrix}
\mathrm{ORTH} & \textit{chased} \\
\mathrm{HEAD} & {}_{\textit{verb}}[\;] \\
\mathrm{SPR} & \left\langle {}_{\textit{phrase}}\begin{bmatrix} \mathrm{HEAD} & {}_{\textit{noun}}[\;] \\ \mathrm{SPR} & \langle\cdots\rangle \\ \mathrm{COMPS} & \langle\cdots\rangle \\ \mathrm{ARGS} & \langle\cdots\rangle \end{bmatrix} \right\rangle \\
\mathrm{COMPS} & \left\langle {}_{\textit{phrase}}\begin{bmatrix} \mathrm{HEAD} & {}_{\textit{noun}}[\;] \\ \mathrm{SPR} & \langle\cdots\rangle \\ \mathrm{COMPS} & \langle\cdots\rangle \\ \mathrm{ARGS} & \langle\cdots\rangle \end{bmatrix} \right\rangle
\end{bmatrix}
$$

Types, Attributes, and Values — very large number of equivalent notations

Typed Feature Structures: More Notational Variants

```
syn-struc := feat-struc &
[ HEAD pos,
  SPR *list*,
  COMPS *list* ].


phrase := syn-struc &
[ ARGS *list* ].


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

Feature Structure Unification: The Logics

TFS_1: $\begin{bmatrix} \text{FOO } x \\ \text{BAR } x \end{bmatrix}_{a}$ TFS_2: $\begin{bmatrix} \text{FOO } x \\ \text{BAR } y \end{bmatrix}_{a}$

TFS_3: $\begin{bmatrix} \text{FOO } y \\ \text{BAR } x \\ \text{BAZ } x \end{bmatrix}_{b}$ TFS_4: $\begin{bmatrix} \text{FOO } \boxed{1}\, x \\ \text{BAR } \boxed{1} \end{bmatrix}_{a}$

Signature

$$\begin{tikzcd} a \;\; \begin{matrix}\text{FOO}\\ \text{BAR}\end{matrix} \arrow[d, dash] & x \arrow[d, dash] \\ b \;\; \text{BAZ} & y \end{tikzcd}$$

$\text{TFS}_1 \sqcap \text{TFS}_2 \equiv \text{TFS}_2$ $\text{TFS}_1 \sqcap \text{TFS}_3 \equiv \text{TFS}_3$ $\text{TFS}_3 \sqcap \text{TFS}_4 \equiv \begin{bmatrix} \text{FOO } \boxed{1}\, y \\ \text{BAR } \boxed{1} \\ \text{BAZ } x \end{bmatrix}_{b}$

UNIFICATION (‘$\sqcap$’) ensures compatibility and combines all information

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.

$$
{}_{\textit{phrase}}\begin{bmatrix}\texttt{HEAD} & \textit{noun} \\ \texttt{SPR} & \langle\rangle \\ \texttt{COMPS} & \langle\rangle\end{bmatrix}
\qquad
{}_{\textit{phrase}}\begin{bmatrix}\texttt{HEAD} & \textit{verb} \\ \texttt{SPR} & \langle\rangle \\ \texttt{COMPS} & \langle\rangle\end{bmatrix}
\qquad
{}_{\textit{phrase}}\begin{bmatrix}\texttt{HEAD} & \textit{verb} \\ \texttt{SPR} & \left\langle {}_{\textit{phrase}}\begin{bmatrix}\,\end{bmatrix} \right\rangle \\ \texttt{COMPS} & \langle\rangle\end{bmatrix}
$$

'NP' **'S'** **'VP'**

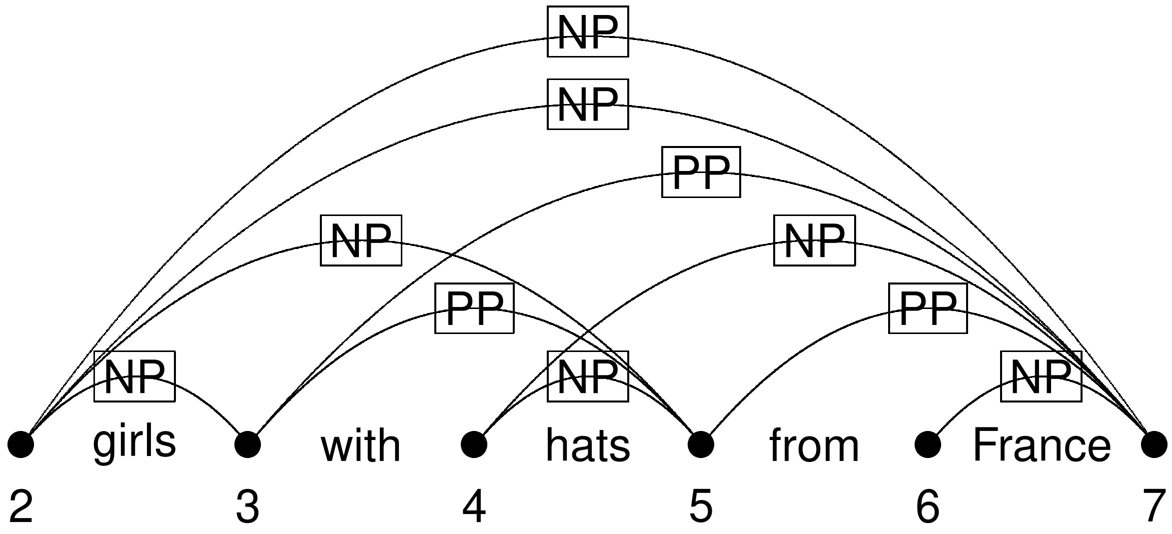
The Format of Grammar Rules in the LKB

$$
\textit{mother}\begin{bmatrix} \mathtt{HEAD} & \boxed{1} \\ \mathtt{SPR} & \boxed{2} \\ \mathtt{COMPS} & \langle\rangle \\ \multicolumn{2}{c}{\ldots} \end{bmatrix} \longrightarrow \textit{daughter}_1\begin{bmatrix} \mathtt{HEAD} & \boxed{1} \\ \mathtt{SPR} & \boxed{2} \\ \mathtt{COMPS} & \langle\boxed{3}\rangle \end{bmatrix}, \quad \boxed{3}\,\textit{daughter}_2\begin{bmatrix} \quad\ldots\quad \end{bmatrix}
$$

$$
\textit{mother}\begin{bmatrix} \mathtt{HEAD} & \boxed{1} \\ \mathtt{SPR} & \boxed{2} \\ \mathtt{COMPS} & \langle\rangle \\ \multicolumn{2}{c}{\ldots} \\ \mathtt{ARGS} & \left\langle \textit{daughter}_1\begin{bmatrix} \mathtt{HEAD} & \boxed{1} \\ \mathtt{SPR} & \boxed{2} \\ \mathtt{COMPS} & \langle\boxed{3}\rangle \end{bmatrix}, \quad \boxed{3}\,\textit{daughter}_2\begin{bmatrix} \quad\ldots\quad \end{bmatrix} \right\rangle \end{bmatrix}
$$

Bottom-Up Chart Parsing in the LKB

- Initialize chart: retrieve all lexical entries for all words in the input string;
- Parsing: apply all rules to all adjacent tuples of edges in the chart;
- Add new edge for each successful instantiation of a grammar rule.



Literature and Further Pointers

On-line User Manual for the LKB

- at NTNU, use the command '`lkbdoc`' from the `xterm` command line;
- anywhere, go to '`http://www-csli.stanford.edu/~aac/lkb.html`'.

Course Materials: Exercises *and* Solutions

'`http://www.coli.uni-sb.de/trondheim01/`'

Background Reading

- Sag, Ivan A. and Thomas Wasow (1999). *Syntactic Theory. A Formal Introduction*. Stanford, CA: CSLI Publications.
- Ann A. Copestake (forthcoming). *An Introduction to Grammar Engineering in the LKB.* Stanford, CA: CSLI Publications.
- LinGO project at CSLI Stanford: '`http://lingo.stanford.edu/`'.