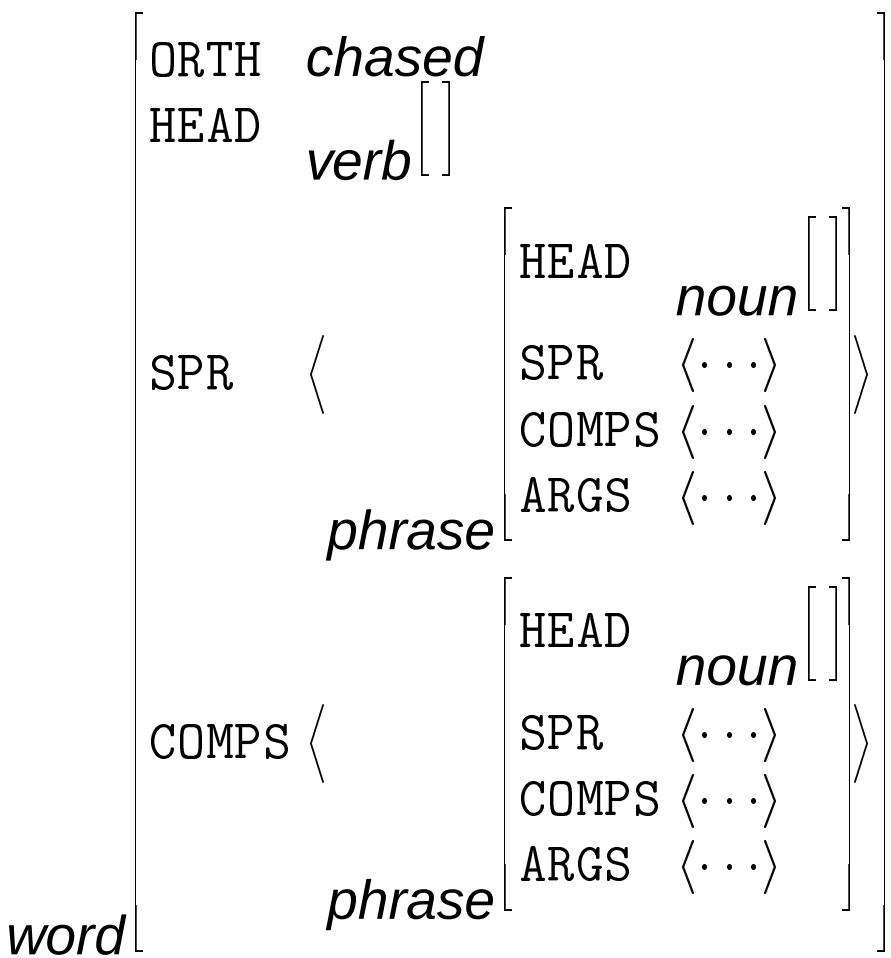
Typed Feature Structures: Notational Variants



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

Typed Feature Structures: More Notational Variants

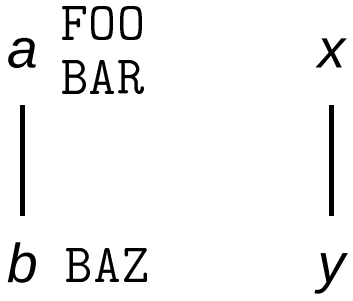
```
syn-struc := feat-struc &
[ HEAD pos,
  SPR *list*,
  COMPS *list* ].


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: $a\begin{bmatrix}\mathtt{FOO}\ x\\ \mathtt{BAR}\ x\end{bmatrix}$ TFS_2: $a\begin{bmatrix}\mathtt{FOO}\ x\\ \mathtt{BAR}\ y\end{bmatrix}$

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

Signature

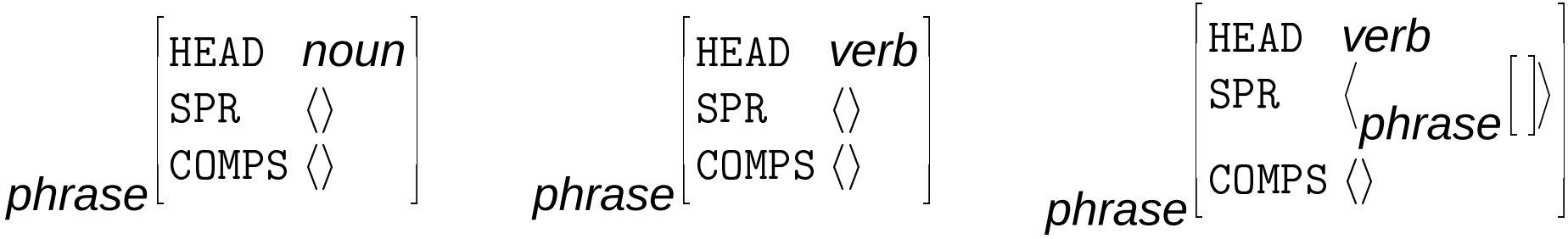


$\mathsf{TFS}_1 \sqcap \mathsf{TFS}_2 \equiv \mathsf{TFS}_2$ $\mathsf{TFS}_1 \sqcap \mathsf{TFS}_3 \equiv \mathsf{TFS}_3$ $\mathsf{TFS}_3 \sqcap \mathsf{TFS}_4 \equiv b\begin{bmatrix}\mathtt{FOO}\ \boxed{1}y\\ \mathtt{BAR}\ \boxed{1}\\ \mathtt{BAZ}\ x\end{bmatrix}$

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.



‘NP’ **‘S’** **‘VP’**

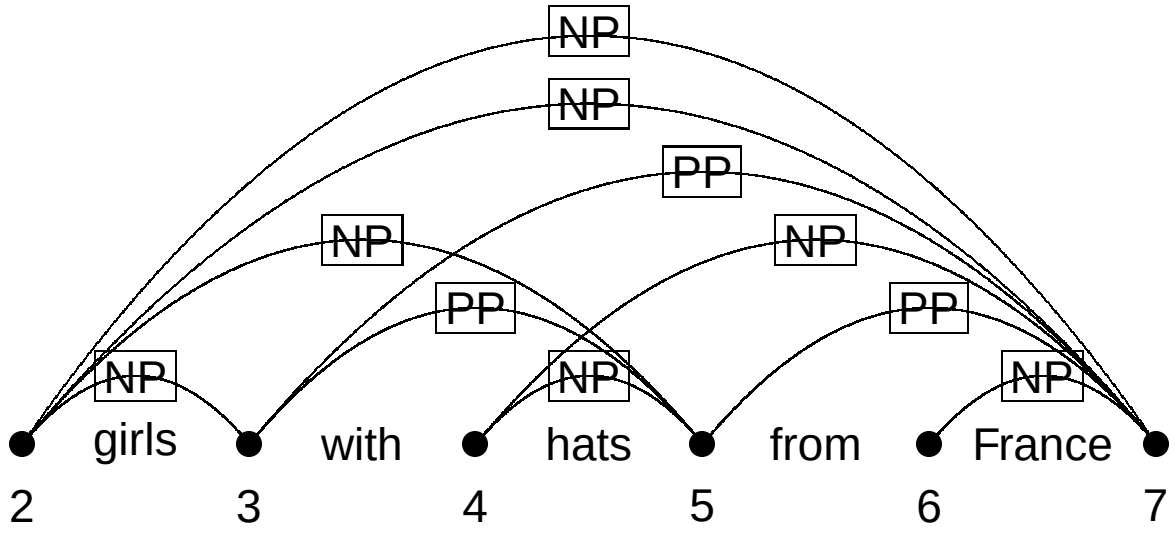
The Format of Grammar Rules in the LKB

$$\textit{mother}\begin{bmatrix}\texttt{HEAD} & \boxed{1}\\ \texttt{SPR} & \boxed{2}\\ \texttt{COMPS} & \langle\rangle\\ & \cdots\end{bmatrix} \longrightarrow \textit{daughter}_1\begin{bmatrix}\texttt{HEAD} & \boxed{1}\\ \texttt{SPR} & \boxed{2}\\ \texttt{COMPS} & \langle\boxed{3}\rangle\end{bmatrix}, \quad \boxed{3}\begin{bmatrix}\cdots\end{bmatrix} \; \textit{daughter}_2\begin{bmatrix}\end{bmatrix}$$

$$\textit{mother}\left[\begin{array}{ll}\texttt{HEAD} & \boxed{1}\\ \texttt{SPR} & \boxed{2}\\ \texttt{COMPS} & \langle\rangle\\ & \cdots\\ \texttt{ARGS} & \left\langle \textit{daughter}_1\begin{bmatrix}\texttt{HEAD} & \boxed{1}\\ \texttt{SPR} & \boxed{2}\\ \texttt{COMPS} & \langle\boxed{3}\rangle\end{bmatrix}, \quad \boxed{3}\begin{bmatrix}\cdots\end{bmatrix} \; \textit{daughter}_2\begin{bmatrix}\end{bmatrix} \right\rangle\end{array}\right]$$

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/`'.