Checking grammaticality

We want to solve the grammaticality problem for HPSG grammars. Given a set of constraints (a grammar) and a term (a query), is there a model of the constraints that satisfies the query?

Three approaches

1. **"Semantics of HPSG"**:
   Construct models of the grammar. Solution is enumeration of models.
   (King 1989), Algorithms (Carpenter 1992, Chapter 15), ALE 2.0 (Carpenter and Penn 1994)

2. **"Constraint Solving"**:
   Instantiate the query until all constraints in the grammar are satisfied.
   For HPSG e.g., in (Götz and Meurers 1995)

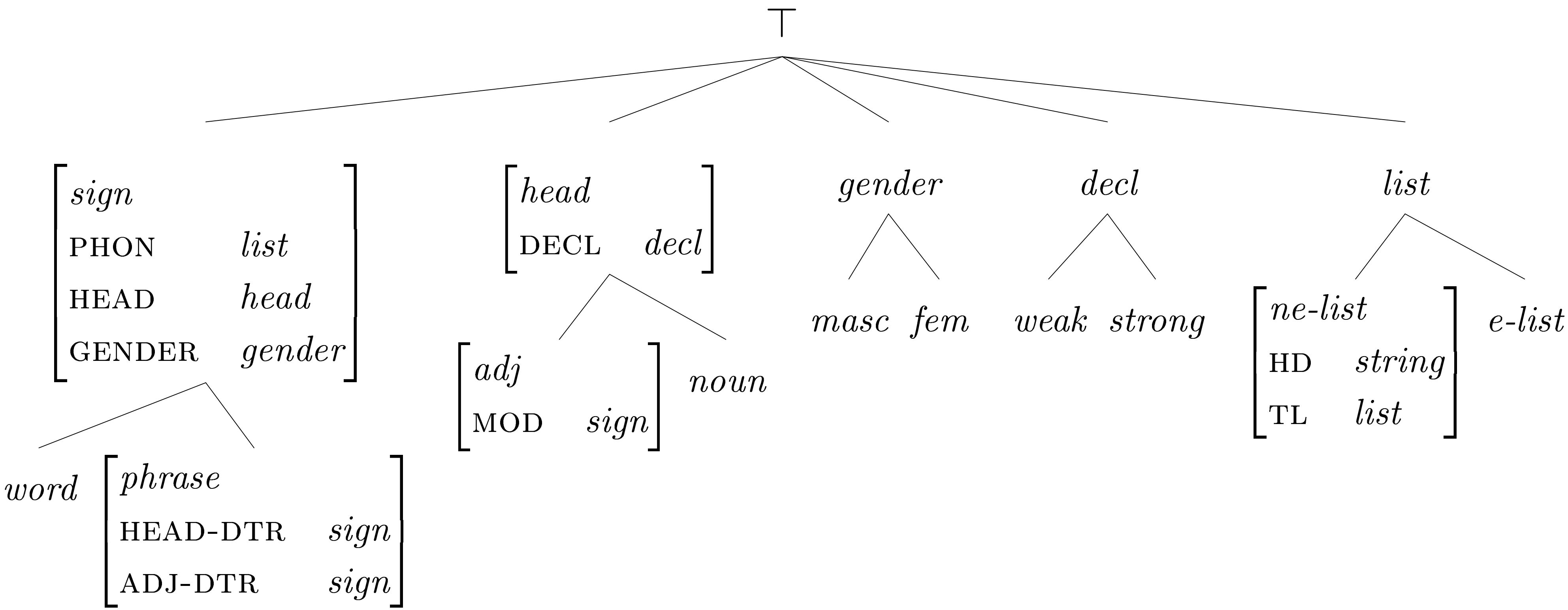
3. **"Lazy Constraint Solving"**:
   Only check those parts of the query, which are specific enough.
   (Aït-Kaci, Podelski, and Goldstein 1993), simplified version for HPSG (Götz and Meurers 1996)

An example

Adjectival agreement in German

	fem		masc	
strong	*(manch)*	**kleine** *Sorge*	*(manch)*	**kleiner** *Erfolg*
weak	*(die)*	**kleine** *Sorge*	*(der)*	**kleine** *Erfolg*
	(some/the)	**small** *worry*	*(some/the)*	**small** *success*

Signature
(sample-grammars/lazyness/signature)



Theory
(sample-grammars/lazyness/theory)

$$
\textit{word} \;\rightarrow\;
\begin{bmatrix}
\text{PHON} & \left\langle \textit{kleine} \right\rangle \\
\text{HEAD} & \textit{adj} \\
\text{GENDER} & \textit{fem}
\end{bmatrix} \vee
\begin{bmatrix}
\text{PHON} & \left\langle \textit{kleine} \right\rangle \\
\text{HEAD} & \begin{bmatrix} \textit{adj} & \\ \text{DECL} & \textit{weak} \end{bmatrix} \\
\text{GENDER} & \textit{masc}
\end{bmatrix} \vee
\begin{bmatrix}
\text{PHON} & \left\langle \textit{kleiner} \right\rangle \\
\text{HEAD} & \begin{bmatrix} \textit{adj} & \\ \text{DECL} & \textit{strong} \end{bmatrix} \\
\text{GENDER} & \textit{masc}
\end{bmatrix} \vee
\begin{bmatrix}
\text{PHON} & \left\langle \textit{Erfolg} \right\rangle \\
\text{HEAD} & \textit{noun} \\
\text{GENDER} & \textit{masc}
\end{bmatrix} \vee
\begin{bmatrix}
\text{PHON} & \left\langle \textit{Sorge} \right\rangle \\
\text{HEAD} & \textit{noun} \\
\text{GENDER} & \textit{fem}
\end{bmatrix}
$$

Theory (cont.)

$$
phrase \rightarrow \begin{bmatrix}
\text{PHON} & \left\langle \boxed{1} \mid \boxed{2} \right\rangle \\
\text{HEAD} & \boxed{3} \\
\text{GENDER} & \boxed{4} \\
\text{HEAD-DTR} & \boxed{5} \begin{bmatrix} \text{PHON} & \boxed{2} \\ \text{HEAD} & \boxed{3} \\ \text{GENDER} & \boxed{4} \end{bmatrix} \\
\text{ADJ-DTR} & \begin{bmatrix} \text{PHON} & \left\langle \boxed{1} \right\rangle \\ \text{HEAD} \mid \text{MOD} & \boxed{5} \\ \text{GENDER} & \boxed{4} \end{bmatrix}
\end{bmatrix}
$$

$$
adj \rightarrow \begin{bmatrix}
\text{MOD} \mid \text{HEAD} \mid \text{DECL} & \boxed{1} \\
\text{DECL} & \boxed{1}
\end{bmatrix}
$$

Theory after ordinary compilation

$$
word \rightarrow \begin{bmatrix} \text{PHON} & \left\langle \textit{kleine} \right\rangle \\ \text{HEAD} & \boxed{\boxed{A}}\,\textit{adj} \\ \text{GENDER} & \textit{fem} \end{bmatrix} \vee
\begin{bmatrix} \text{PHON} & \left\langle \textit{kleine} \right\rangle \\ \text{HEAD} & \boxed{\boxed{B}}\begin{bmatrix} \textit{adj} & \\ \text{DECL} & \textit{weak} \end{bmatrix} \\ \text{GENDER} & \textit{masc} \end{bmatrix} \vee
\begin{bmatrix} \text{PHON} & \left\langle \textit{kleiner} \right\rangle \\ \text{HEAD} & \boxed{\boxed{C}}\begin{bmatrix} \textit{adj} & \\ \text{DECL} & \textit{strong} \end{bmatrix} \\ \text{GENDER} & \textit{masc} \end{bmatrix} \vee
\begin{bmatrix} \text{PHON} & \left\langle \textit{Erfolg} \right\rangle \\ \text{HEAD} & \textit{noun} \\ \text{GENDER} & \textit{masc} \end{bmatrix} \vee
\begin{bmatrix} \text{PHON} & \left\langle \textit{Sorge} \right\rangle \\ \text{HEAD} & \textit{noun} \\ \text{GENDER} & \textit{fem} \end{bmatrix}
$$

Theory after ordinary compilation (cont.)

$$
phrase \;\rightarrow\; \begin{bmatrix}
\text{PHON} & \left\langle \boxed{1} \mid \boxed{2} \right\rangle \\
\text{HEAD} & \boxed{\boxed{D}}\boxed{3} \\
\text{GENDER} & \boxed{4} \\
\text{HEAD-DTR} & \boxed{\boxed{E}}\boxed{5}\begin{bmatrix} \text{PHON} & \boxed{2} \\ \text{HEAD} & \boxed{3} \\ \text{GENDER} & \boxed{4} \end{bmatrix} \\
\text{ADJ-DTR} & \boxed{\boxed{F}}\begin{bmatrix} \text{PHON} & \left\langle \boxed{1} \right\rangle \\ \text{HEAD} & \boxed{\boxed{G}}\begin{bmatrix} \text{MOD} & \boxed{5} \end{bmatrix} \\ \text{GENDER} & \boxed{4} \end{bmatrix}
\end{bmatrix}
$$

$$
adj \;\rightarrow\; \begin{bmatrix}
\text{MOD} & \boxed{\boxed{H}}\begin{bmatrix} \text{HEAD} & \boxed{\boxed{I}}\begin{bmatrix} \text{DECL} & \boxed{1} \end{bmatrix} \end{bmatrix} \\
\text{DECL} & \boxed{1}
\end{bmatrix}
$$

Answering a query too eagerly

`?- query(word).` $\underset{1}{\rightsquigarrow}$ $\boxed{Q}\textit{word}$

$$\underset{2}{\rightsquigarrow} \begin{bmatrix} \textit{word} & \\ \text{PHON} & \left\langle \textit{kleine} \right\rangle \\ \text{HEAD} & \boxed{A}\,\textit{adj} \\ \text{GENDER} & \textit{fem} \end{bmatrix}$$

$$\underset{3}{\rightsquigarrow} \begin{bmatrix} \textit{word} & \\ \text{PHON} & \left\langle \textit{kleine} \right\rangle \\ \text{HEAD} & \begin{bmatrix} \textit{adj} & \\ \text{MOD} & \boxed{H}\begin{bmatrix} \text{HEAD} & \boxed{I}\begin{bmatrix} \text{DECL} & \boxed{1} \end{bmatrix} \end{bmatrix} \\ \text{DECL} & \boxed{1} \end{bmatrix} \\ \text{GENDER} & \textit{fem} \end{bmatrix}$$

Answering a query too eagerly (cont.)

$$\underset{4}{\leadsto} \quad \begin{bmatrix} \textit{word} & \\ \text{PHON} & \left\langle \textit{kleine} \right\rangle \\ \text{HEAD} & \begin{bmatrix} \textit{adj} & \\ \text{MOD} & \begin{bmatrix} \text{PHON} & \left\langle \textit{kleine} \right\rangle \\ \text{HEAD} & \boxed{\boxed{I\&A}} \begin{bmatrix} \textit{adj} & \\ \text{DECL} & \boxed{1} \end{bmatrix} \\ \text{GENDER} & \textit{fem} \end{bmatrix} \\ \text{DECL} & \boxed{1} \end{bmatrix} \\ \text{GENDER} & \textit{fem} \end{bmatrix}$$

$$\underset{5}{\leadsto} \quad \ldots$$

Theory after lazy compilation

$$
word \;\rightarrow\;
\begin{bmatrix} \text{PHON} & \left\langle \textit{kleine} \right\rangle \\ \text{HEAD} & \cancel{\boxed{A}}\, adj \\ \text{GENDER} & fem \end{bmatrix} \vee
\begin{bmatrix} \text{PHON} & \left\langle \textit{kleine} \right\rangle \\ \text{HEAD} & \boxed{B} \begin{bmatrix} adj & \\ \text{DECL} & weak \end{bmatrix} \\ \text{GENDER} & masc \end{bmatrix} \vee
\begin{bmatrix} \text{PHON} & \left\langle \textit{kleiner} \right\rangle \\ \text{HEAD} & \boxed{C} \begin{bmatrix} adj & \\ \text{DECL} & strong \end{bmatrix} \\ \text{GENDER} & masc \end{bmatrix} \vee
\begin{bmatrix} \text{PHON} & \left\langle \textit{Erfolg} \right\rangle \\ \text{HEAD} & noun \\ \text{GENDER} & masc \end{bmatrix} \vee
\begin{bmatrix} \text{PHON} & \left\langle \textit{Sorge} \right\rangle \\ \text{HEAD} & noun \\ \text{GENDER} & fem \end{bmatrix}
$$

Theory after lazy compilation (cont.)

$$
\textit{phrase} \;\rightarrow\; \begin{bmatrix}
\text{PHON} & \left\langle \boxed{1} \mid \boxed{2} \right\rangle \\
\text{HEAD} & \cancel{\boxed{\boxed{D}}}\,\boxed{3} \\
\text{GENDER} & \boxed{4} \\
\text{HEAD-DTR} & \boxed{\boxed{E}}\,\boxed{5}\begin{bmatrix} \text{PHON} & \boxed{2} \\ \text{HEAD} & \boxed{3} \\ \text{GENDER} & \boxed{4} \end{bmatrix} \\
\text{ADJ-DTR} & \boxed{\boxed{F}}\begin{bmatrix} \text{PHON} & \left\langle \boxed{1} \right\rangle \\ \text{HEAD} & \boxed{\boxed{G}}\begin{bmatrix} \text{MOD} & \boxed{5} \end{bmatrix} \\ \text{GENDER} & \boxed{4} \end{bmatrix}
\end{bmatrix}
$$

$$
\textit{adj} \;\rightarrow\; \begin{bmatrix}
\text{MOD} & \boxed{\boxed{H}}\begin{bmatrix} \text{HEAD} & \boxed{\boxed{I}}\begin{bmatrix} \text{DECL} & \boxed{1} \end{bmatrix} \end{bmatrix} \\
\text{DECL} & \boxed{1}
\end{bmatrix}
$$

Answering a query the lazy way

`?- query(word).` $\underset{1}{\leadsto}$ $word$

$$\texttt{?- query(}\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \langle \textit{kleine} \rangle \end{bmatrix}\texttt{).}$$

$$\underset{1}{\leadsto} \quad \boxed{Q}\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \langle \textit{kleine} \rangle \end{bmatrix}$$

$$\underset{2}{\leadsto} \quad \begin{bmatrix} \textit{word} & \\ \textsc{phon} & \langle \textit{kleine} \rangle \\ \textsc{head} & \textit{adj} \\ \textsc{gender} & \textit{fem} \end{bmatrix}$$

Which grammars can be dealt with in a lazy fashion?

Lazy evaluation is not sound in general!

<u>Example</u>

$$word \;\rightarrow\; \begin{bmatrix} \text{PHON} & \left\langle kleine \right\rangle \\ \text{HEAD} & adj \\ \text{GENDER} & fem \end{bmatrix} \wedge$$

$$\begin{bmatrix} \text{PHON} & \left\langle kleine \right\rangle \\ \text{HEAD} & \begin{bmatrix} adj & \\ \text{DECL} & weak \end{bmatrix} \\ \text{GENDER} & masc \end{bmatrix} \vee$$

. . .

Type *word* is inconsistent, but the query

```
?- query(word).
```

succeeds nevertheless.

Restricting permissible grammars: type consistency

Aït-Kaci et al. (1993) gave a (syntactic) condition on theories called *well-formedness* that guarantees soundness for query evaluation. This condition is too strong for HPSG grammars of the type we consider.

Instead we propose a semantic condition on permissible grammars: *type consistency*. A theory is type consistent iff it has model where every type has a non-empty denotation. This is the weakest condition that guarantees soundness of lazy evaluation, i.e., a necessary condition.

What laziness can't solve

$$\texttt{?- query(}\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \left\langle \textit{kleine} \right\rangle \\ \textsc{head} & \begin{bmatrix} \textit{adj} & \\ \textsc{decl} & \textit{weak} \end{bmatrix} \end{bmatrix}\texttt{).}$$

$$\underset{1}{\leadsto} \quad \boxed{Q}\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \left\langle \textit{kleine} \right\rangle \\ \textsc{head} & \boxed{R}\begin{bmatrix} \textit{adj} & \\ \textsc{decl} & \textit{weak} \end{bmatrix} \end{bmatrix} \quad \underset{2}{\leadsto} \quad \ldots$$

The constraints on *adj* will have to be tested at one point, so we cannot drop more goals to achieve termination.

Alternative idea: the constraint on *adj* is only relevant, when an adjective is combined with another constituent.

$\Rightarrow$ statement to postpone execution needed

Delays

<u>Syntax</u>: `delay(`*<Goalname>*`,` *<DelayDesc>*`).`

- *<Goalname>* is the name of a goal.
- *<DelayDes>* is a special kind of description, where
  - the special list syntax ([], . . .) is disallowed,
  - the additional unary operator `@` can be used.

<u>Interpretation</u>

- A delay statement postpones the execution of the call to *<Goalname>* until the call is at least as specific as *<DelayDesc>*.
- If a type in *<DelayDesc>* is prefixed by the new operator `@`, the corresponding type in the call has to be more specific than the one provided.
- If several delay statements exist for one goal, all of them have to be satisfied before a call to that goal can be executed.

More on delays

Example

To ensure that our constraint on adj is only applied when the modified constituent is known, we can include the following delay statement.

```
delay(adj, mod: (word;phrase)).
```

Using the `@` operator one can express this as

```
delay(adj, mod: @sign).
```

Delaying forever

If during a calculation only delayed goals remain, then ConTroll fails this computation and reports `no more undelayed goals`.

By chosing one of the options of the menu item “remaining delays action” in the “options” menu, it is possible to start the debugger at this point.

Summary: theory vs. computation

In principle, the computational system should give answers to queries according to the denotational semantics of the logic behind HPSG (e.g., King's SRL).

Restrictions:

- Processing is lazy, to improve efficiency and termination behaviour. As a result, not all objects described by the solution have to be grammatical. Rather, if the grammar is type consistent, it is guaranteed that some object described by the solution is grammatical.

- The constraints are processed one after the other. Delays can be specified to ensure termination and efficienct processing.

- Processing leads to multiple solutions in case the same solution is described by two disjuncts of a constraint.

References

Aït-Kaci, H., A. Podelski, and S. C. Goldstein (1993). Order-sorted theory unification. Technical Report 32, Digital Equipment Corporation.

Carpenter, B. (1992). *The Logic of Typed Feature Structures - With Applications to Unification Grammars, Logic Programs and Constraint Resolution*, Volume 32 of *Cambridge Tracts in Theoretical Computer Science*. New York: Cambridge University Press.

Carpenter, B. and G. Penn (1994). ALE – the attribute logic engine, user's guide, version 2.0.1, December 1994. Technical report, Computational Linguistics Program, Philosopy Department, Carnegie Mellon University.

Götz, T. and W. D. Meurers (1995). Compiling HPSG type constraints into definite clause programs. In *Proceedings of the Thirty-Third Annual Meeting of the ACL*, Boston. Association for Computational Linguistics.

Götz, T. and W. D. Meurers (1996). The importance of being lazy - using lazy evaluation to process queries to HPSG grammars. In *Proceedings of TALN 96 (joint session with the Third International Conference on HPSG)*, Marseille, France.

King, P. (1989). *A Logical Formalism for Head-Driven Phrase Structure Grammar*. Ph. D. thesis, University of Manchester.