



# ***Towards a competitive dependency grammar formalism***

Ralph Debusmann

`rade@ps.uni-sb.de`

Programming Systems Lab

Universität des Saarlandes

# *Dependency grammar*

- ⑥ idea: words in a sentence depend on each other
- ⑥ dates back to the middle ages
- ⑥ modern dependency grammar: Tesnière (1959) (also: Melcuk, Sgall)
- ⑥ dependency analysis = dependency tree (\*1)

# Comparison with context-free grammar

(\*2)

	CFG	DG
nodes	phrasal and terminal	only terminal
gr. function	no, only categories	first class citizen
word order	rules	no, unconstrained
lexicalization	no, rule-based	lexicalized (valency)

# Valency

- ⑥ in the lexicon, each word specifies its daughters, similar to subcategorization in HPSG
- ⑥ e.g. *liebt* requires a subject and an object:

$$\textit{liebt} = [ \text{valency\_syn} : \{ \text{subj!}, \text{obj!} \} ]$$

- ⑥ *Frau* requires a determiner and may be modified by arbitrary many adjectives:

$$\textit{Frau} = [ \text{valency\_syn} : \{ \text{det!}, \text{adj*} \} ]$$

# Word Order

- ⑥ so far: no assumptions about word order
- ⑥ but: need to state word order constraints to prevent overgeneration
- ⑥ existing formalisms often invent non-declarative extensions to constrain word order

# *The failure of dependency grammar*

- ⑥ some of the most well-known formalisms:
  - △ Meaning Text Theory, Melcuk (1988)
  - △ Functional Generative Description, Sgall et al (1986)
  - △ Word Grammar, Hudson (1990)
  - △ Slot Grammar, McCord (1990)
- ⑥ none of these as successful as e.g. HPSG in the linguistic mainstream
- ⑥ why? lack of declarativity (esp. word order)

# *Topological Dependency Grammar*

- ⑥ new dependency grammar formalism, described in Duchier/Debusmann 2001 (ACL) and Debusmann 2001 (Diplomarbeit)
- ⑥ includes a new, declarative way of adding word order constraints to dependency grammar
- ⑥ dependency relations clearly separated from word order
- ⑥ makes use of topological fields theory

# Topological fields theory

- ⑥ traditional descriptive theory (Herling 1821, Erdmann 1886) for German word order
- ⑥ divides a sentence into contiguous substrings and assigns to them so-called *topological fields*
- ⑥ topological fields: Vorfeld, left sentence bracket, Mittelfeld, right sentence bracket, Nachfeld
- ⑥ e.g.:

Vorfeld	(	Mittelfeld	)
<i>Peter</i>	<i>liebt</i>	<i>eine blonde Frau.</i>	
<i>Peter</i>	<i>versucht</i>	<i>eine Frau</i>	<i>zu lieben.</i>



# *Topological trees*

- ⑥ idea: add to the dependency tree a second level of analysis: the topology tree (\*3)
- ⑥ two differences
  1. edge labels = topological fields
  2. daughters of each node ordered by the label of their incoming edge
- ⑥ the topology tree is a flattening of the dependency tree
- ⑥ same notion of valency

# Topological valency

- ⑥ lexical entry now specifies *syntactic valency* and *topological valency*
- ⑥ e.g. *liebt* may have at most one daughter in the Vorfeld and an arbitrary number of daughters in the Mittelfeld:

*liebt* :

$$\left[ \begin{array}{l} \text{valency\_syn} : \{\text{subj!}, \text{obj!}\} \\ \text{valency\_top} : \{\text{vf?}, \text{mf*}, \text{nf?}\} \end{array} \right]$$

# ***TDG overview***

- ⑥ copes well with languages with a high degree of word order freedom (e.g. German, Dutch)
- ⑥ highly lexicalized, lexical inheritance to state linguistic generalizations
- ⑥ efficient parser, uses constraint technology developed at Programming Systems Lab
- ⑥ but: so far only covers syntax

# *Advancing to semantics*

- ⑥ plan of my doctoral thesis: extend TDG grammar formalism with a concurrent syntax-semantics interface
- ⑥ concurrency = syntax inferences can rule out semantic readings and vice versa
- ⑥ falls out rather naturally from the constraint-based approach to parsing
- ⑥ facilitates incorporation of preferences (CHORUS)

# Recovering predicate-argument structure

- ⑥ problem: dependency trees reflect only syntactic argument structure
- ⑥ does not always match semantic predicate-argument structure, e.g. passive, control/raising
- ⑥ need a more semantically oriented structure to recover predicate-argument structure

# *Thematic graphs*

- ⑥ idea: add a new level of representation to the dependency and topology tree levels: thematic graphs (\*4)
- ⑥ represent semantic predicate-argument structure
- ⑥ edge labels = thematic roles (agent, patient...)
- ⑥ again: notion of valency re-used

# Thematic valency

- ⑥ each lexical entry now specifies syntactic, topological and thematic valencies
- ⑥ e.g. *liebt* must have an agent and a patient:

*liebt* :

$$\left[ \begin{array}{l} \text{valency\_syn} : \{\text{subj!}, \text{obj!}\} \\ \text{valency\_top} : \{\text{vf?}, \text{mf*}\} \\ \text{valency\_them} : \{\text{ag!}, \text{pt!}\} \end{array} \right]$$

# Linking

- ⑥ problem: need to link the thematic graph and the dependency tree
- ⑥ idea: thematic roles realized as dependency relations
- ⑥ e.g. for *liebt*: agent realized as the subject, and patient realized as the object:

*liebt* :

valency_syn	:	{subj!, obj!}
valency_top	:	{vf?, mf*}
valency_them	:	{ag!, pt!}
link	:	{ag : subj, pt : obj}



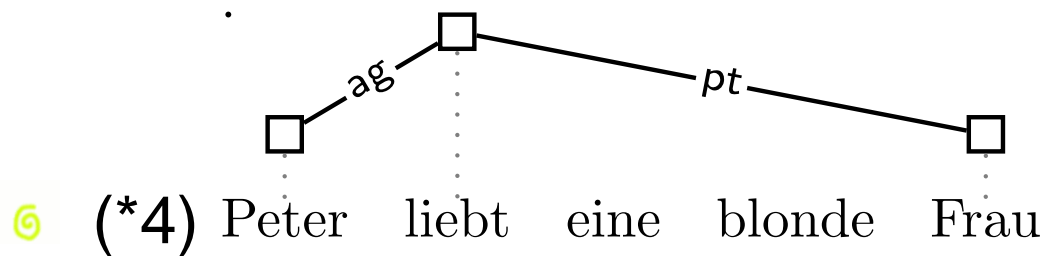
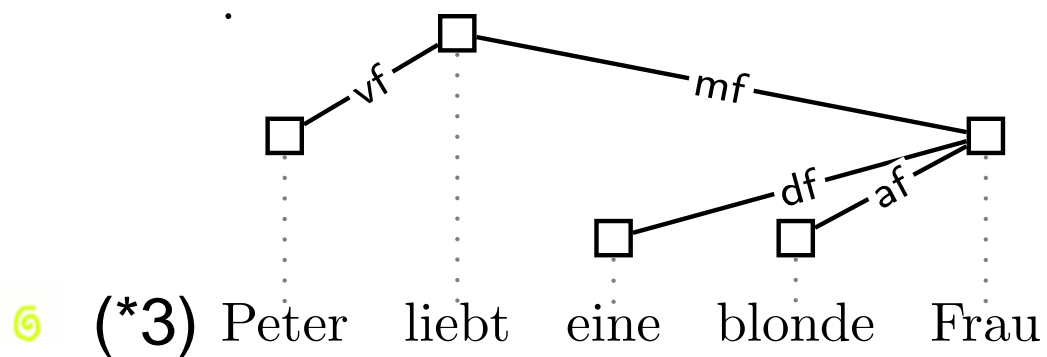
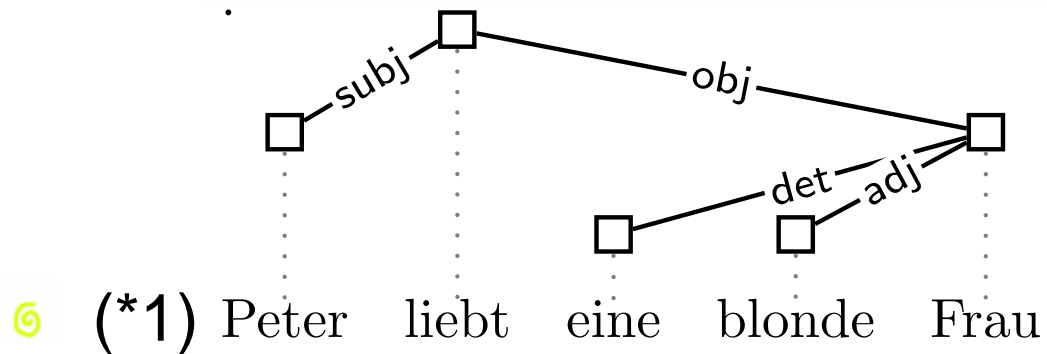
# ***State of the art***

- ⑥ new implementation of the dependency parser including the thematic graph level
- ⑥ also includes a scope tree-level to represent quantifier scope (not talked about here)
- ⑥ no breakdown in parsing performance

## ***Still to do***

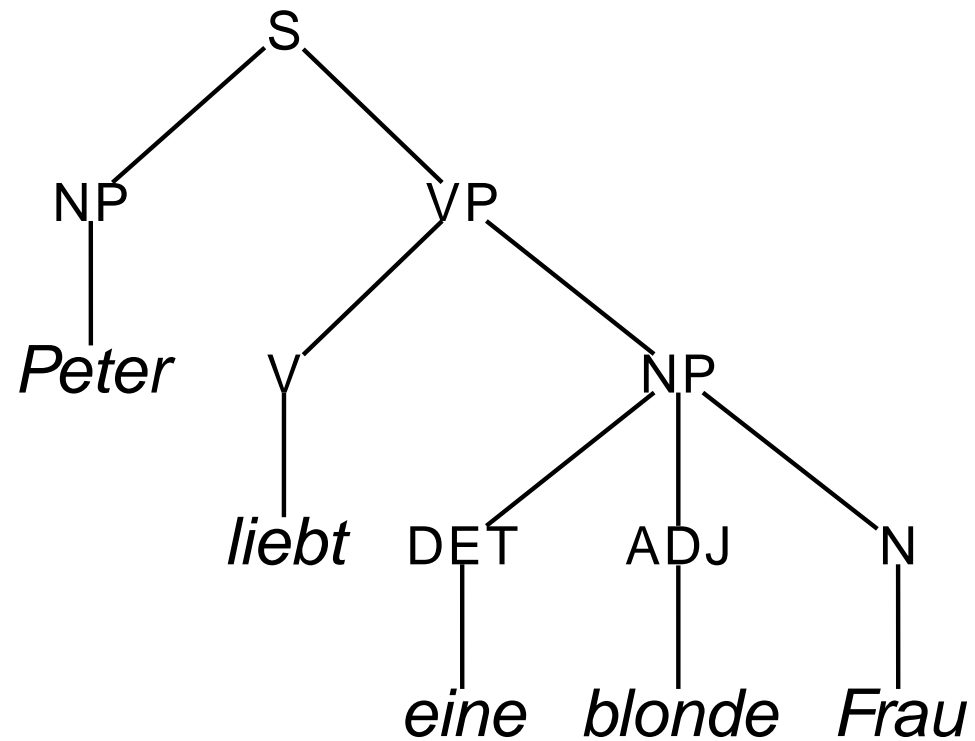
- ⑥ construct semantics (e.g. CLLS) from the thematic graph and the scope tree
- ⑥ incorporate preferences (CHORUS)
- ⑥ improve the grammar formalism (e.g. lexical rules)
- ⑥ improve grammar coverage
- ⑥ improve implementation (parser, GUI)

# X1: Levels of analysis



## X2: Phrase structure analysis

6 (\*2)



# X3: Control and raising

- ⑥ subjects need not be realized locally but also higher in the dependency tree:

