Dependency Grammars

- Topological Dependency Trees: A Constraint-based Account of Linear Precedence
  - Extensible Dependency Grammar: A New Methodology

Sibel Ciddi

NPFL106 - Linguistics

2013 Summer

Framework

- Immediate dependency (ID) → syntactic dependency tree →*(initially)* non-projective, non-ordered
  - The edges of the ID tree → syntactic roles
    - {subject, object, vinf, …}
- Linear precedence (LP) → topological dependency tree → projective, partially ordered.
  - The edges of the LP tree → topological fields
    - {df, mf, vc, xf, ...}

    (determiner-field, mittelfeld, canonical-position, extraposition...)

Discontinuous VP constructions in free word order

(1) (dass) Einen Mann Maria zu lieben versucht

(that) a man_{acc} $Maria_{nom}$ to love tries

To handle discontinuous constituents, Reape's Theory:

1. the unordered syntax tree
2. the totally ordered tree of word order domains, which handles the following:

(2) (dass) Maria einen Mann zu lieben versucht→ scrambling

(3) (dass) einen Mann Maria zu lieben versucht→ scrambling

(4) (dass) Maria versucht, einen Mann zu lieben→ full extraposition

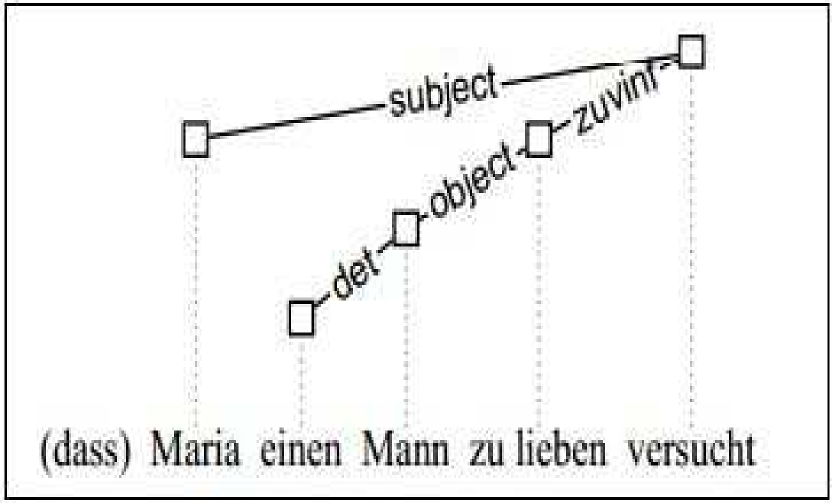
But it does not handle the following:

(5) (dass) Maria einen Mann versucht, zu lieben→ partial extraposition

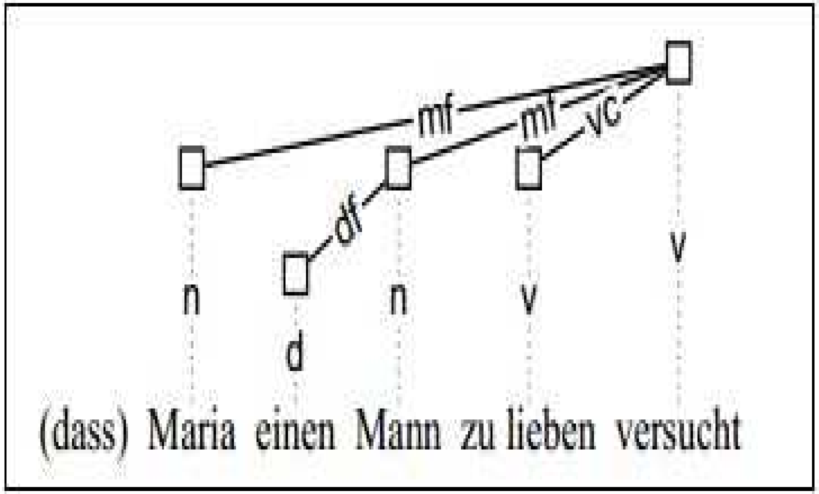
ID / LP Tree Example - free word order

(2) (dass) Maria Einen Mann zu lieben versucht *(scrambling)*

ID Tree



LP Tree



***zu lieben in canonical position {vc}*

Formal Framework & LP Principles

An ID/LP analysis:

- a tuple of (V; E_{ID}; E_{LP}; lex; cat; $valency_{ID}$; $valency_{LP}$; $field_{ext}$; $field_{int}$) s.t. :
  - ID tree: (V; E_{ID}; lex; cat; $valency_{ID}$)
    - $valency_{ID}(w)$ = lex(w).$valency_{ID}$
  - LP tree: (V; E_{LP}; lex; $valency_{LP}$; $field_{ext}$; $field_{int}$)
    - $valency_{LP}(w)$ = lex(w).$valency_{LP}$
  - The following principles are satisfied:

1. A node must land on a transitive head.
2. It may not climb through a barrier.
3. A node must land on, or climb higher than its head.

Valency Satisfaction

A tree (V, E) satisfies the valency assignment, iff:

- The labeled edge, *l-daughter:* $|l(w)| = 1$
- The labeled edge, *l-daughter:* $|l(w)|$ is 0 or 1
- The labeled edge, *l-daughter:* $|l(w)|$ is 0 or more

Example:

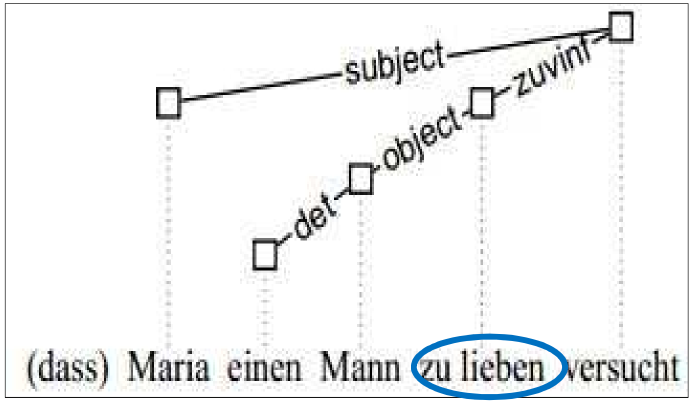
- Valency_{ID}: versucht={subject; zuvinf}
- Valency_{LP}: versucht={mf*; vc?; xf?}

VP- Extraposition (full)

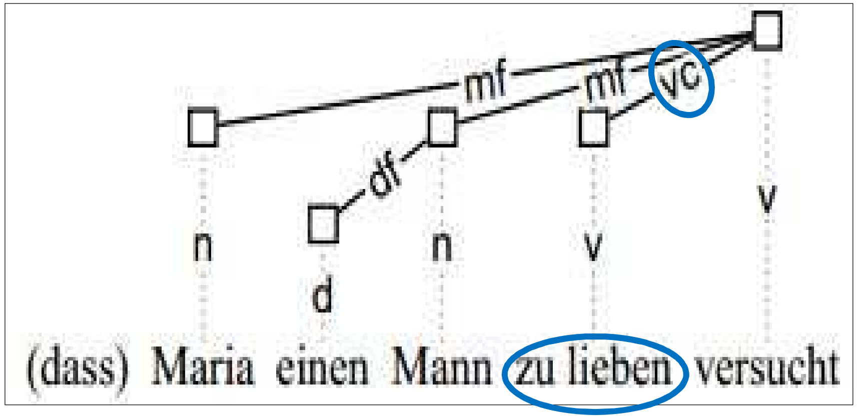
(6) (dass) Maria einen Mann zu lieben versucht

(7) (dass) Maria versucht, einen Mann zu lieben

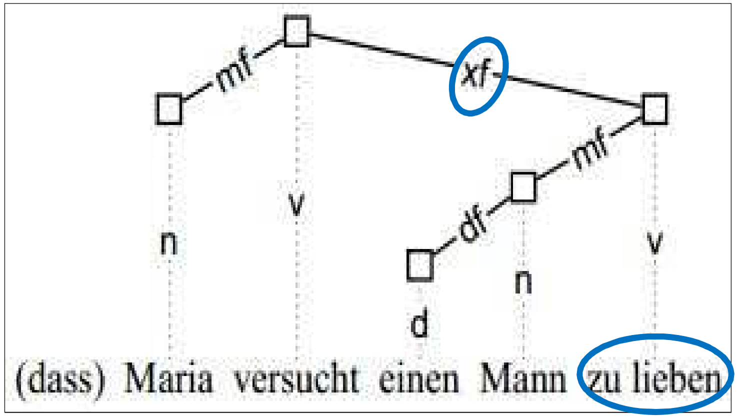
ID Tree



LP Tree: Canonical Position



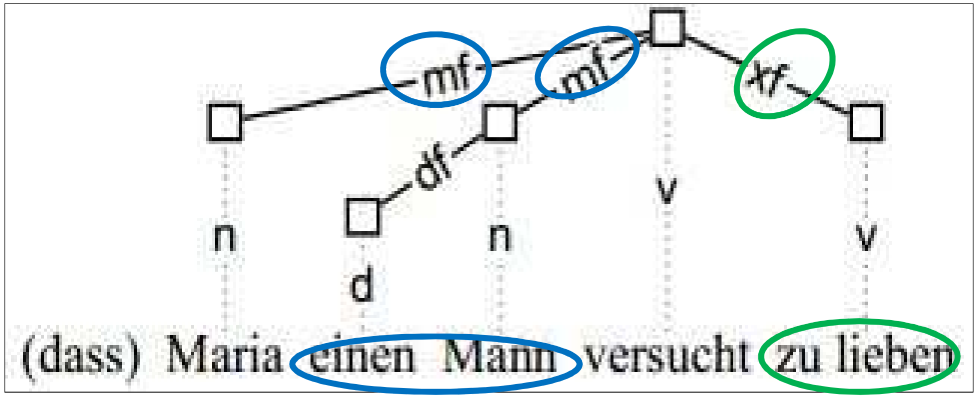
LP Tree: Extraposed (7)



Partial VP- Extraposition

(8) (dass) Maria einen Mann versucht, zu lieben

- *zu lieben extraposed to the right of versucht*
- *its nominal complement einen Mann remains in the Mittelfeld.*

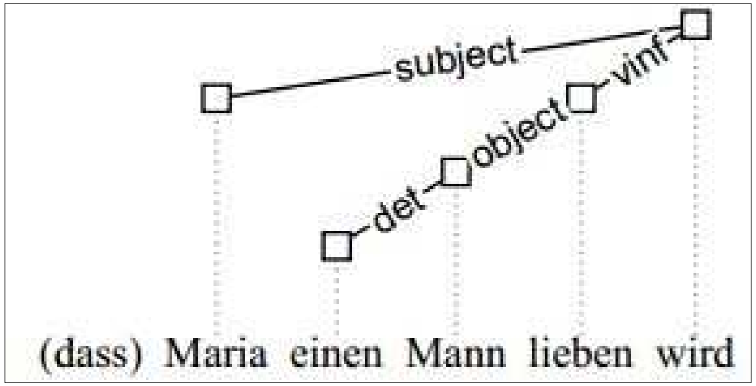


Obligatory Head-Final Placement

(9) (dass) Maria einen Mann lieben wird.

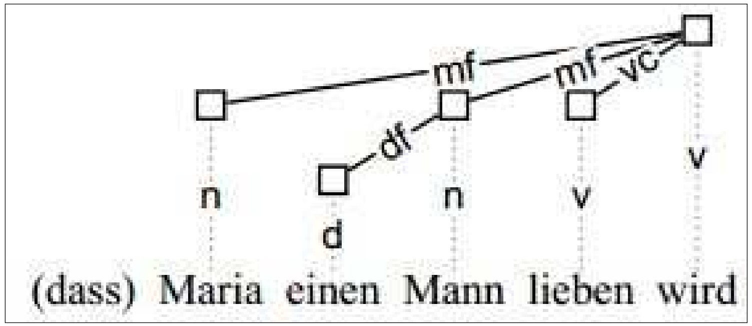
(that) Maria a man_{acc} love will

***In head-final verb-clusters, non-finite verbs precede their verbal heads *(wird).*



ID Tree

$\text{field}_{ext}(\text{lieben}) = \{vc\}$



LP Tree

Extensible Dependency Grammar (XDG)

- **Formalization** *(extended from the LP schema)*

  $$XDG = ((Lab_i; Fea_i; Val_i; Pri_i)^{n}_{i=1}; Pri; Lex)$$

  - n dimensions + multi-dimensional principles + Lex
- **Solver**
  - Infers information about one dimension from another dimension, by using:
    - Either a multi-dimensional principle linking the two dimensions,
    - Or the synchronization induced by the lexical entries.

XDG Example:

- **Dimensions,** Labels, Principles:

$Lab_{\mathbf{ID}}$ = {det; subj; obj; vinf; part}

1. **Tree** :*tree(i), non-lexicalized, parameterized*
2. **Valency**: *valency(i; in_i; out_i) Lexicalized*
3. **Government**: *government(i; $cases_i$; $govern_i$) Lexicalized.*
4. **Agreement**: *agreement(i; $cases_i$; $agree_i$) Lexicalized.*

XDG Example:

- **Dimensions,** Labels, Principles:

Lab$_{\mathbf{LP}}$ = {detf; nounf; vf; lbf; mf; partf; rbf}

1. **Tree, Valency** *(same as the ID dim. principles)*
2. **Order**: *order(i; on*$_i$*;* $\prec$ *i), lexicalized*
3. ***Projectivity**: **:** *projectivity(i), non-lexicalized*
   - **Climbing**: *climbing(i; j), non-lexicalized, multi-dimensional*
   - **Linking**: *linking(i; j; link*$_{i;j}$*) , lexicalized, multi-dimensional*

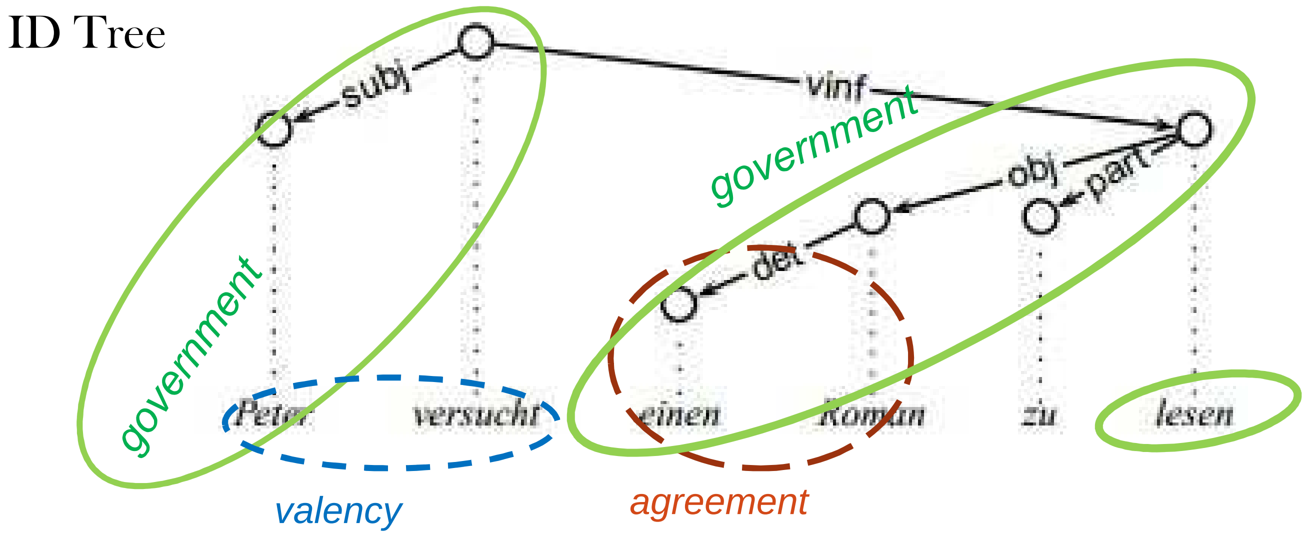
**Projectivity is relevant only for the order principle.*

XDG Example: cont’

- Government and Agreement Principles

Peter versucht einen Roman zu lesen.

Peter tries a_{acc} novel to read



**subject of versucht- nom → gov‘t princ.*

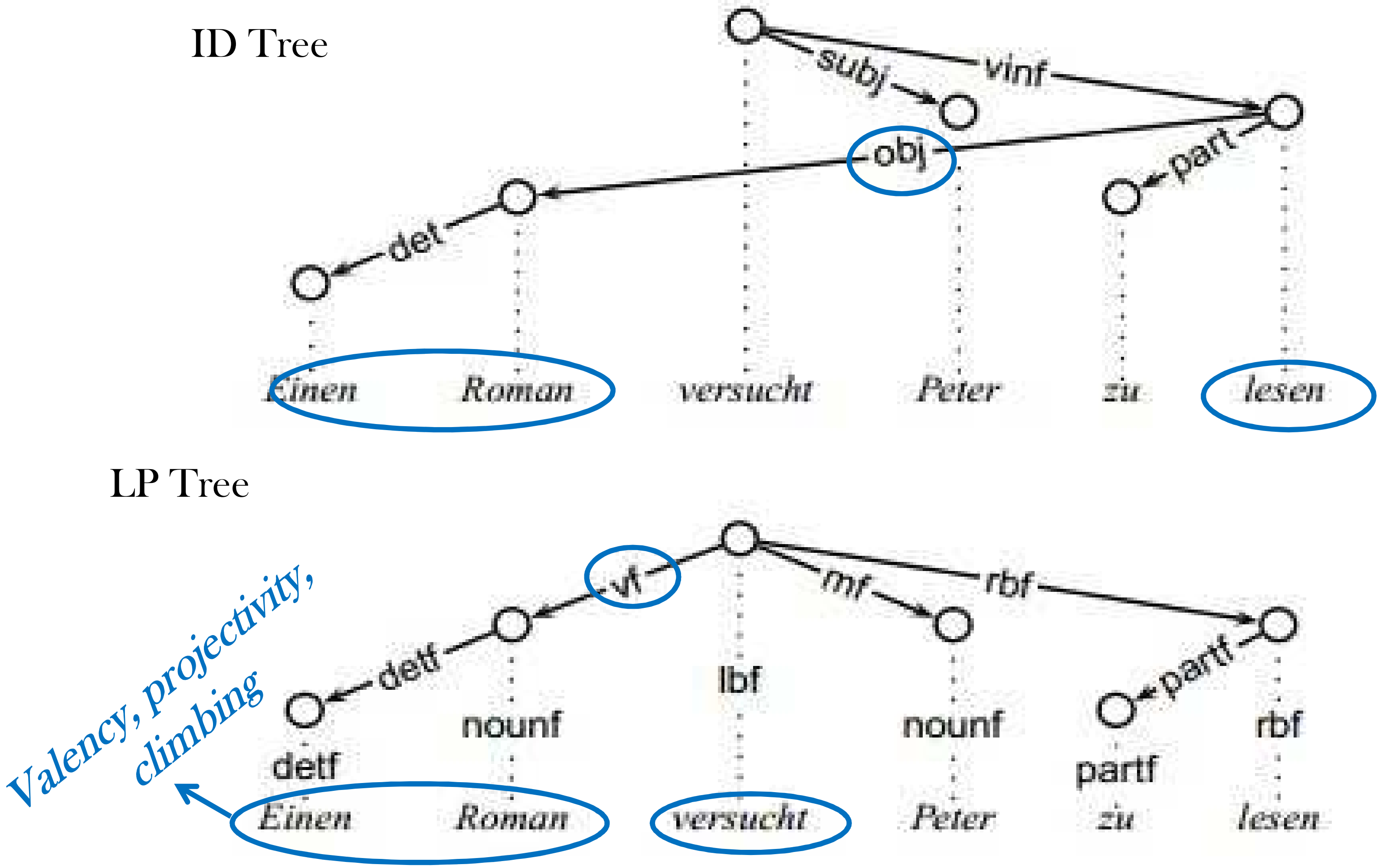
**object of lesen is acc. → gov‘t princ.*

**Roman is acc. due to its acc. det → agr. princ.*

** Versucht must have a subj. ‘Peter‘→ valency princ.*

XDG: Topicalization (Peter versucht einen Roman zu lesen)

Einen Roman versucht Peter zu lesen.



XDG Example: ungrammatical sentence

**Peter einen Roman versucht zu lesen.*

From the lexicon, we have:

Versucht-LP: in{ }, out{ vf?; mf*; rbf?}, on{lbf}, link{ }

- *The finite verb versucht → 1 dependent in its Vorfeld (to left)*
- *This sentence has 2 dependents (? ?)*
- *The sentence gets ruled out before further analysis is made.*

XDG Example: Dutch

Peter probeert een roman te lezen

Peter tries a novel to read.

The Vorfeld of the finite verb probeert cannot be occupied by an object (but only by an object).

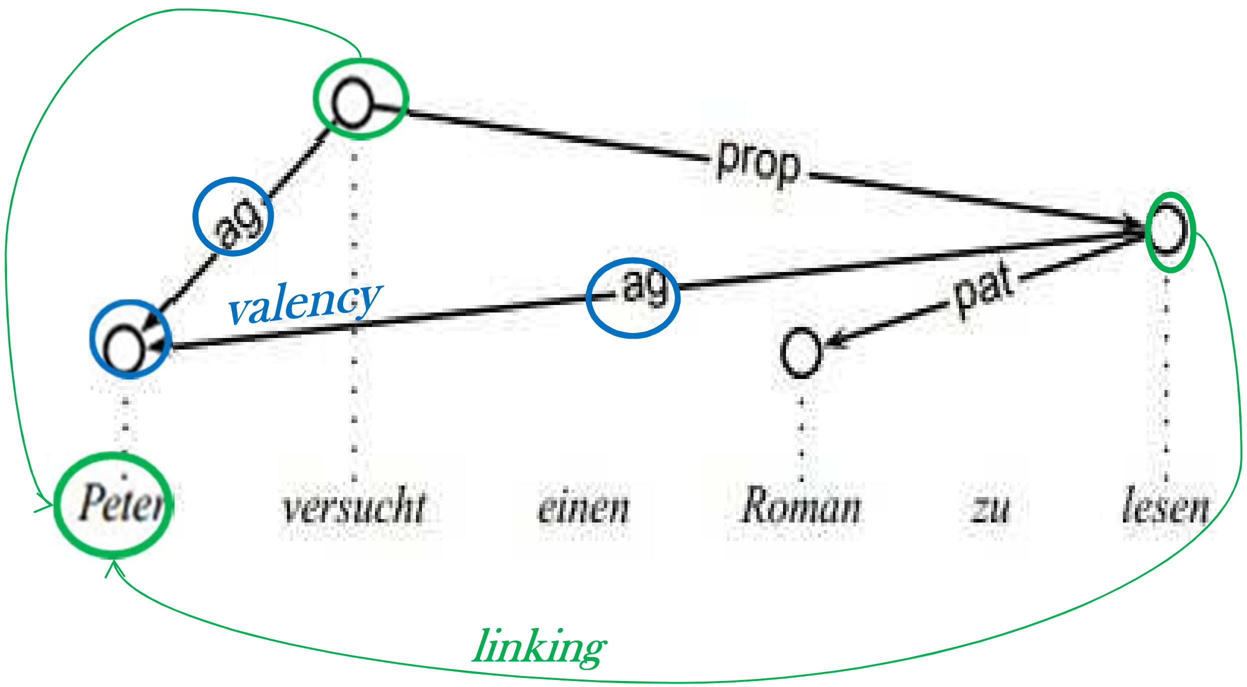
- $\text{link}_{LP;ID} = \{vf \rightarrow \{subj\}\}$.
- **The linking principle**: The Vorfeld of probeert must be filled by a subject, and not by an object.
- Peter in the Vorfeld must be a subject.

XDG Example: Predicate-Argument Structure

Labels: Lab_{PA} = {ag; pat; prop} *(agent, patient, proposition)*

1-Dimensional principles: dag, valency

Multi-Dimensional principles: climbing, linking



XDG Comparisons & Conclusions

1. LFG: Ruling out ambiguity involves several steps:

   - the ambiguity on the f-structure is duplicated

   - the ill-formed structure on the semantic σ-structure is filtered out later.

   + In XDG, the semantic principles can rule out the ill-formed analysis much earlier, typically on the basis of a partial syntactic analysis.

   + Ill-formed analyses are never duplicated, so processing is faster.

2. HPSG: Adaptation of semantics and syntax is not independent.

   - Whenever the syntax part of the grammar changes, the semantics part needs to be adapted.

   + In XDG, semantic phenomena can be described much more independently from syntax.

   + Facilitates grammar engineering, and the statement of cross-linguistic generalizations