# *A Comparative Introduction to XDG: The Linear Precedence Dimension*


Ralph Debusmann

and

Denys Duchier

Programming Systems Lab, Saarland University, Saarbrücken, Germany

and

Équipe Calligramme, LORIA, Nancy, France

# *This presentation*

- German has free word order
- consequence: non-projective analyses, discontinuous constituents
- why dependency grammar? can directly and naturally capture the non-projective analyses
- but: so far, we do not restrict word order at all
- question: how can we restrict word order in a declarative way?

# *Approaches to free word order*

- we will introduce the following approaches to handling free word order:
  1. Gazdar et al. (1985), Uszkoreit (1987): GPSG
  2. Reape (1990, 1994), Kathol (1995, 2000): HPSG
  3. Gerdes/Kahane (2001): DG
  4. Duchier/Debusmann (2001): DG

# *Many other approaches*

- we cannot introduce many other approaches for lack of time:
  - Becker/Rambow (1994): TAG
  - Müller (1999): HPSG
  - Bröker (1999): DG
  - Kruijff (2000): CG

# Scrambling example

1. canonical, continuous word order (no extraction):

   | | | | | | |
   |---|---|---|---|---|---|
   | *(dass)* | *Maria* | *einen* | *Roman* | *zu schreiben* | *verpricht.* |
   | (that) | Maria | a | novel | to write | promises. |

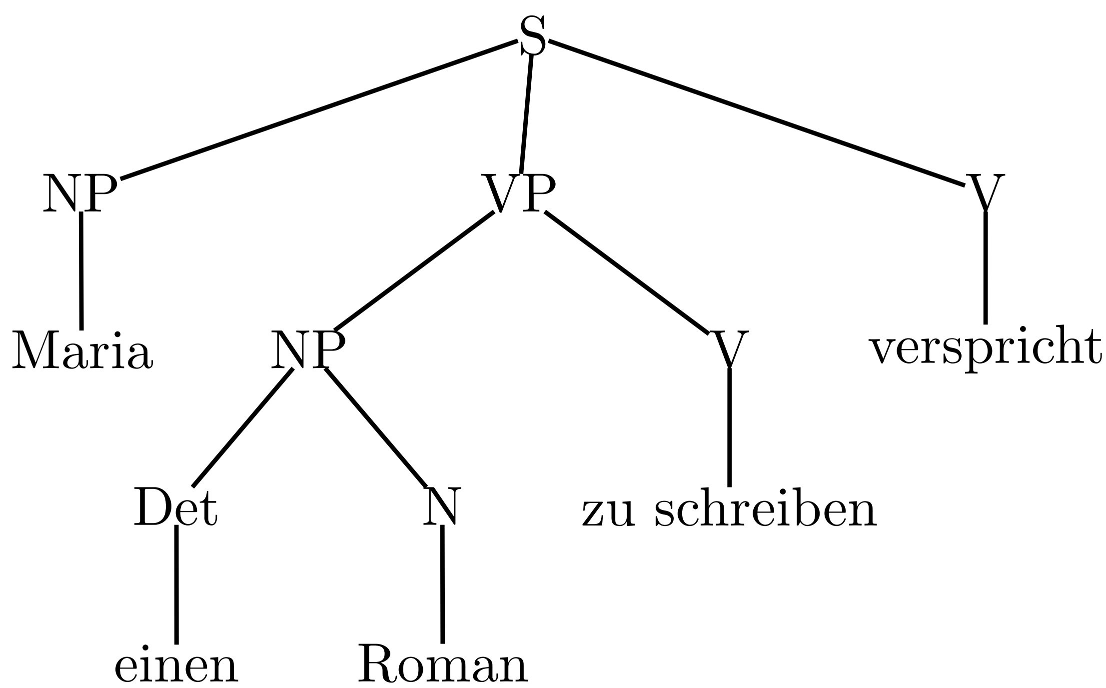
   "(that) Maria promises to write a novel."

2. object NP extracted: scrambling (Ross 1967)

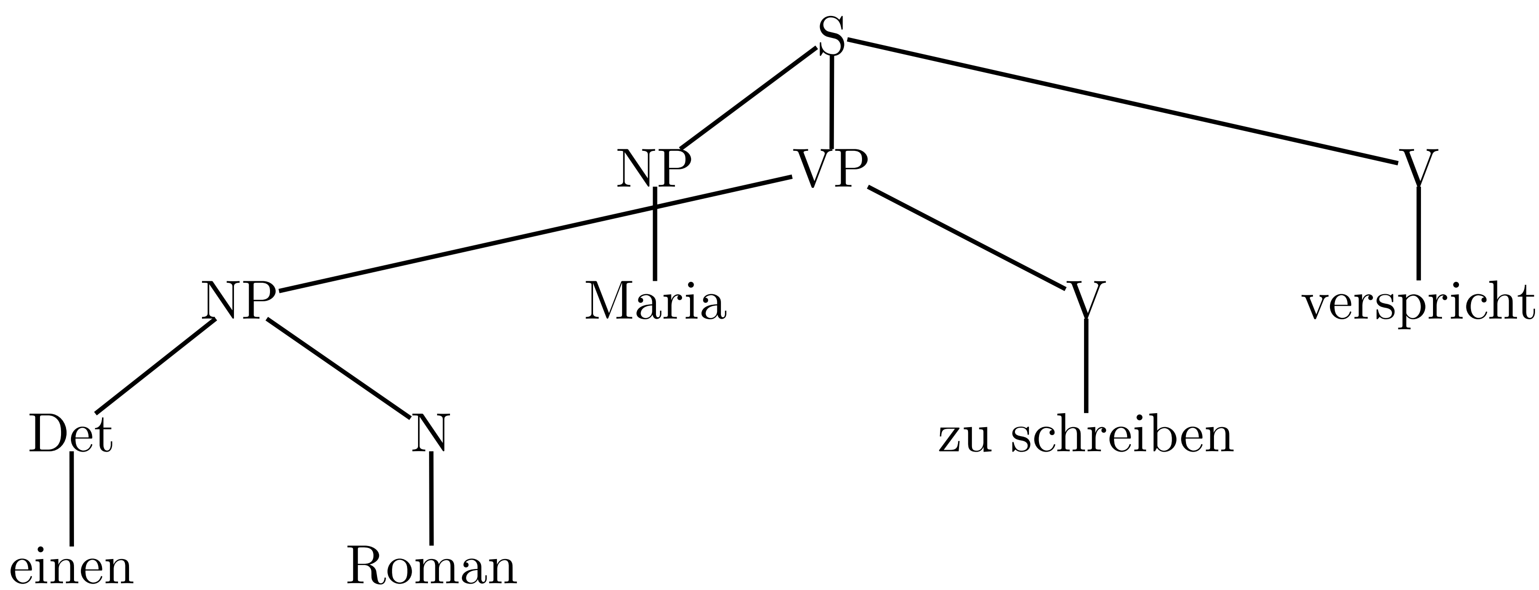
   | | | | | | |
   |---|---|---|---|---|---|
   | *(dass)* | *einen* | *Roman* | *Maria* | *zu schreiben* | *verspricht.* |
   | (that) | a | novel | Maria | to write | promises. |

   "(that) Maria promises to write a novel."

# *Example analysis (no scrambling)*



# Example analysis (scrambling)

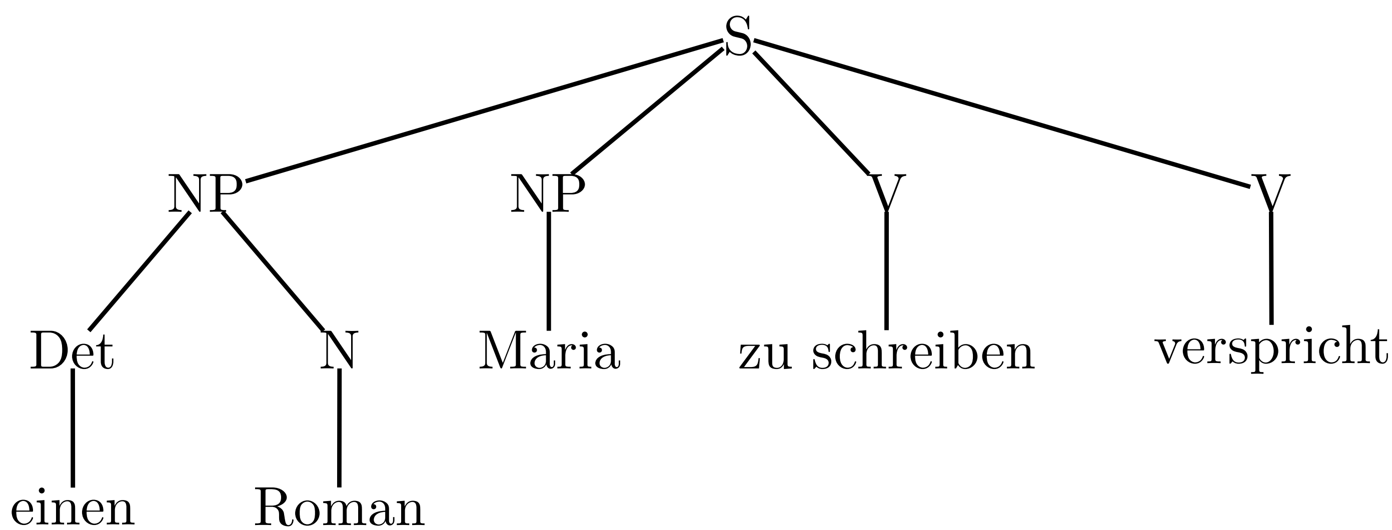



- problem for naive phrase structure-based approaches: VP *einen Roman zu schreiben* is discontinuous

# GPSG

- Gazdar et al. (1985)
- idea is to separate:
  - Immediate Dominance (ID): $\text{NP} \rightarrow \{\text{DET}, \text{ADJ}, \text{N}\}$
  - Linear Precedence (LP): $\text{DET} < \text{ADJ} < \text{N}$
- but: ID/LP distinction only for local phrase structure rules, cannot handle scrambling (non-local)
- idea: Uszkoreit (1987): flatter phrase structure for German

# *Flatter phrase structure*

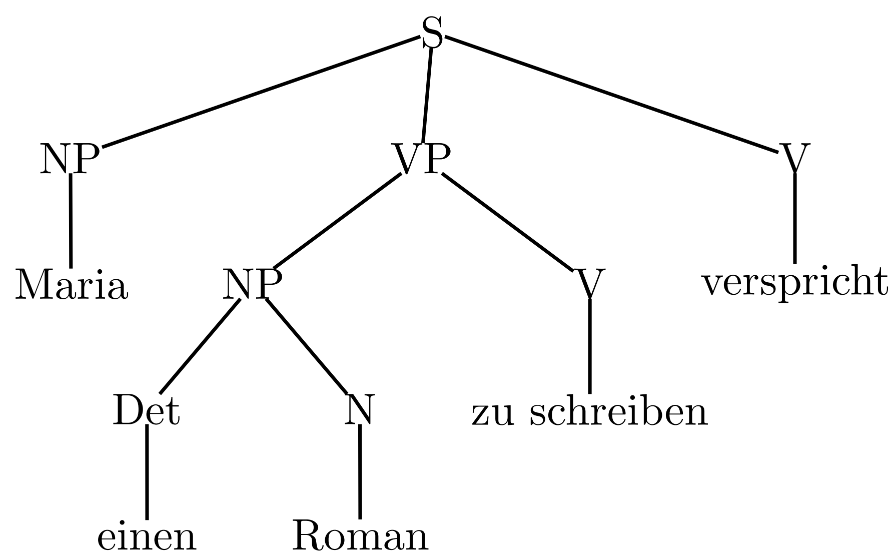


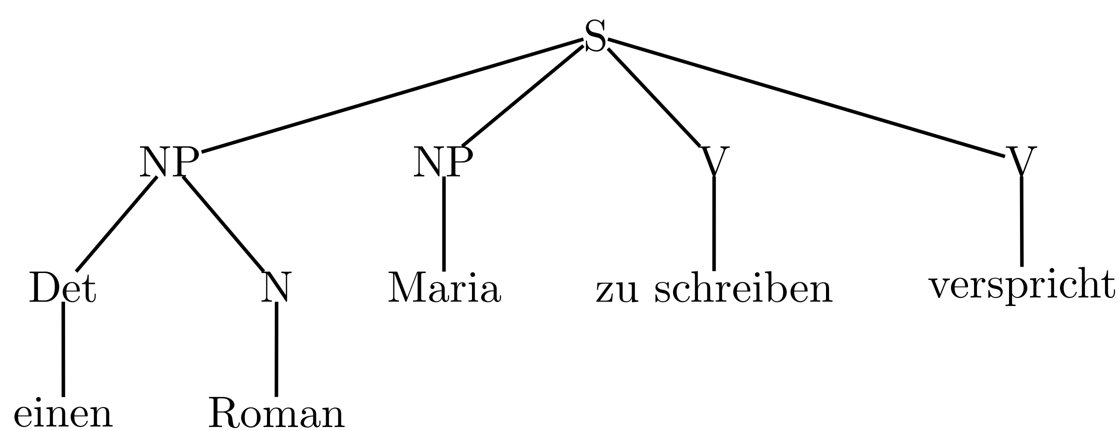

- ID: $\textsf{S} \rightarrow \{\textsf{NP}, \textsf{NP}, \textsf{V}, \textsf{V}\}$
- LP: $\textsf{NP} < \textsf{V}$
- but: we lose the syntactic dependencies (e.g. that *zu schreiben* depends on *verspricht*)

# *Reape: HPSG*

- Reape (1990, 1994)
- two structures:
  1. PS tree
  2. WOD tree
- WOD tree is a flattening of the PS tree
- PS tree: ID, WOD tree: LP

# Example analysis

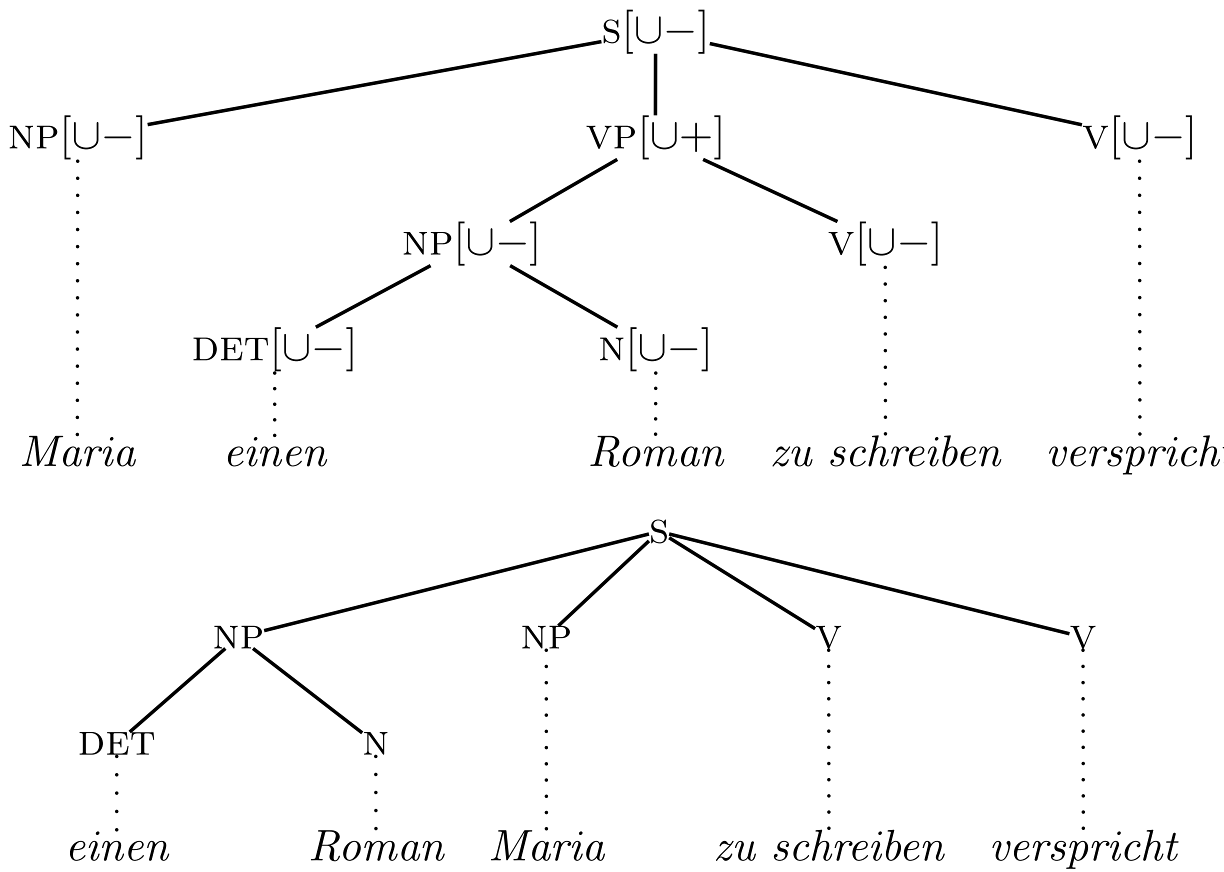


# *Flattening control*

- PS tree and WOD tree not independent
- WOD tree is a flattened PS tree
- each PS tree corresponds to a set of (flattened) WOD tree
- flattening controlled by the binary unioned-attribute
- unioned-values determined by *language-specific principles*
- we will write unioned $= +$ as $[\cup +]$, and unioned $= -$ as $[\cup -]$.
- if $[\cup +]$, the node flattens
- if $[\cup -]$, it does not flatten

# Flattening control example



# ***Defining the dom-function***

- mapping PS tree to a set of corresponding WOD trees:

$$\mathsf{dom}(v) \in \cup^* \{\mathsf{contrib}(v') \mid v' \in \mathsf{dtrs}_{\mathsf{PS}}(v)\}$$

- i.e. the word order domain $\mathsf{dom}(v)$ for PS node $v$ is one obtained by combining the contributions of $v$'s PS daughters by the sequence union operation $\cup^*$

# Sequence union example

- we write $\langle a_1, \ldots, a_n \rangle$ for a sequence of elements $a_1, \ldots, a_n$
- given $A_1 = \langle a, b \rangle$ and $A_2 = \langle c, d \rangle$, $A_1 \cup^* A_2$ is:

$$\begin{array}{rcll} \langle a, b \rangle \cup^* \langle c, d \rangle & = & \{ & \langle a, b, c, d \rangle \\ & & & \langle a, c, b, d \rangle \\ & & & \langle a, c, d, b \rangle \\ & & & \langle c, a, b, d \rangle \\ & & & \langle c, a, d, b \rangle \\ & & & \langle c, d, a, b \rangle \quad \} \end{array}$$

# Contributions

$$\mathsf{contrib}(v) = \begin{cases} \mathsf{dom}(v) & \text{if} \quad v[\cup+] \quad (merged) \\ \langle v \rangle & \text{if} \quad v[\cup-] \quad (inserted) \end{cases}$$

# *Language-specific principles*

- Reape controls the value of $\mathsf{unioned}$ by so-called *language-specific principles*
- for German, they prevent e.g. illegal extraction out of NPs:

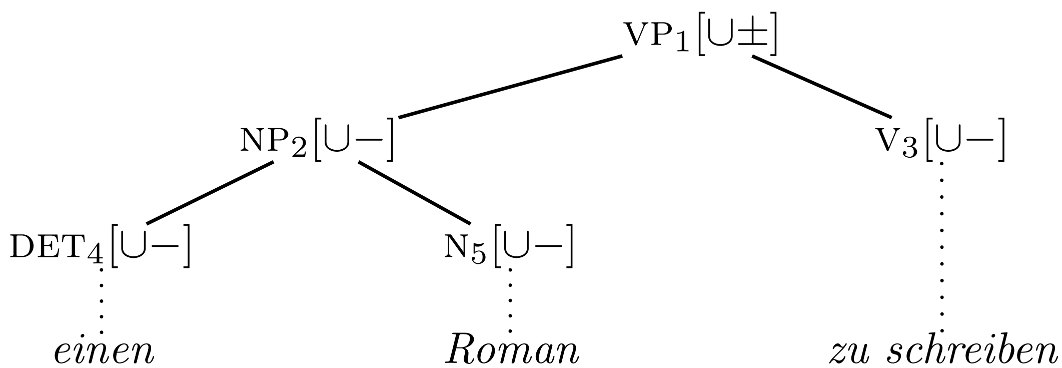
**einen Maria Roman zu lesen verspricht*

- language-specific principle: only verbal projections can be $[\cup+]$:

$$v[\cup+] \quad \Rightarrow \quad \mathsf{cat}(v) \in \{\mathsf{VP}, \mathsf{S}\}$$

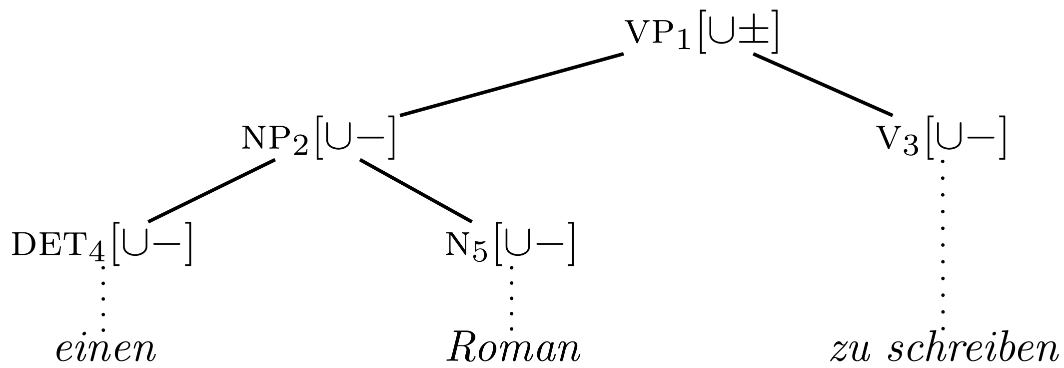
# Examples: VPs (1)

- consider the following PS tree:




- what are the licensed WOD trees?

# Examples: VPs (2)




- we proceed bottom-up, starting with $\textsf{NP}_2$:

$$
\begin{aligned}
\textsf{dom}(\textsf{NP}_2) \quad \in \quad & \textsf{contrib}(\textsf{DET}_4) \cup^* \textsf{contrib}(\textsf{N}_5) \\
& = \langle \textsf{DET}_4 \rangle \cup^* \langle \textsf{N}_5 \rangle \\
& = \{ \ \langle \textsf{DET}_4, \textsf{N}_5 \rangle, \langle \textsf{N}_5, \textsf{DET}_4 \rangle \ \}
\end{aligned}
$$

- linearizations: *einen Roman*, *Roman einen*
- how to rule out the latter?

# *Linear precedence rules*

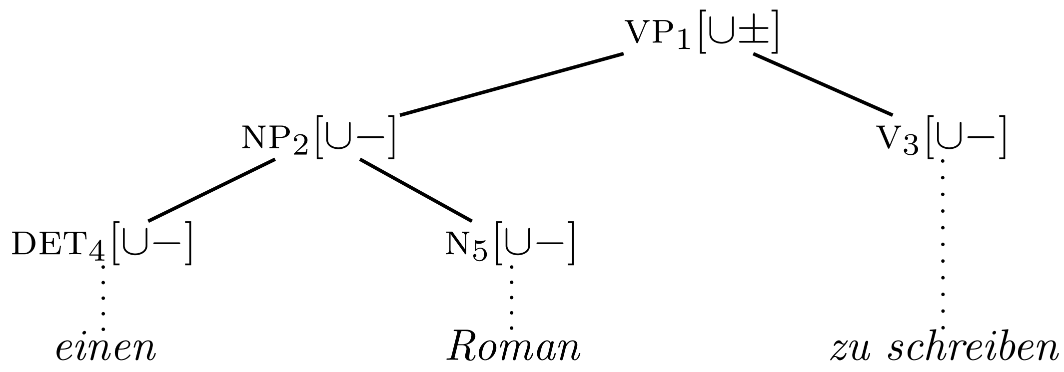
- Reape uses LP rules to restrict the number of licensed word order domains
- LP rules must hold for all $v \in V_{\mathsf{PS}}$ and for all $v_1, v_2 \in \mathsf{dom}(v)$, i.e. they apply locally within a word order domain
- NP rule: determiners must precede nouns:

$$\mathsf{cat}(v_1) = \mathsf{DET} \ \wedge \ \mathsf{cat}(v_2) = \mathsf{N} \quad \Rightarrow \quad v_1 \prec v_2$$

- NP/V rule: NPs must precede Vs:

$$\mathsf{cat}(v_1) = \mathsf{NP} \ \wedge \ \mathsf{cat}(v_2) = \mathsf{V} \quad \Rightarrow \quad v_1 \prec v_2$$

# Examples: VPs (3)

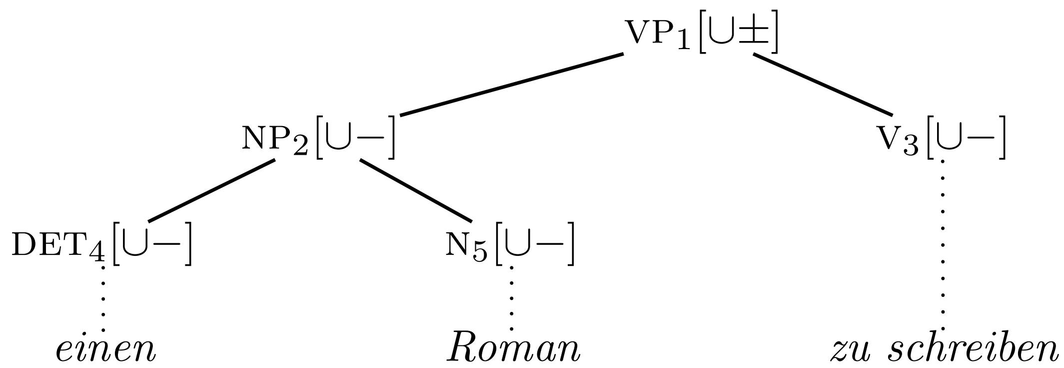



- we continue with $\text{VP}_1$:

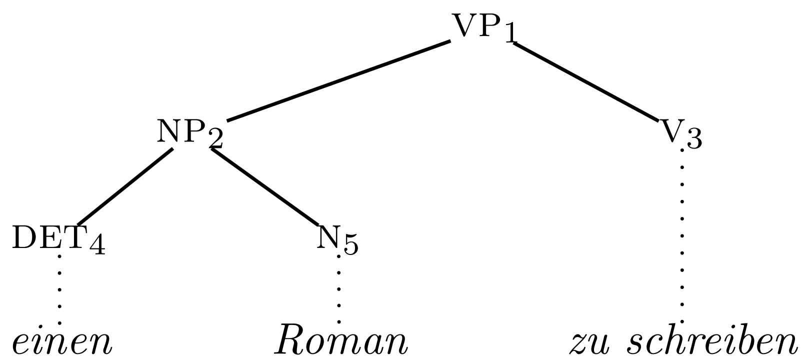
$$\begin{aligned} \mathsf{dom}(\mathsf{VP}_1) &\in \mathsf{contrib}(\mathsf{NP}_2) \cup^* \mathsf{contrib}(\mathsf{V}_3) \\ &= \langle \mathsf{NP}_2 \rangle \cup^* \langle \mathsf{V}_3 \rangle \\ &= \{ \langle \mathsf{NP}_2, \mathsf{V}_3 \rangle, \langle \mathsf{V}_3, \mathsf{NP}_2 \rangle \} \end{aligned}$$

- linearizations: *einen Roman zu schreiben*, *zu schreiben einen Roman*
- by the NP/V rule, we rule out the latter

# Examples: NPs (4)




- licensed WOD at VP$_1$: $\langle$NP$_2$, V$_3\rangle$
- corresponds to the following WOD tree:

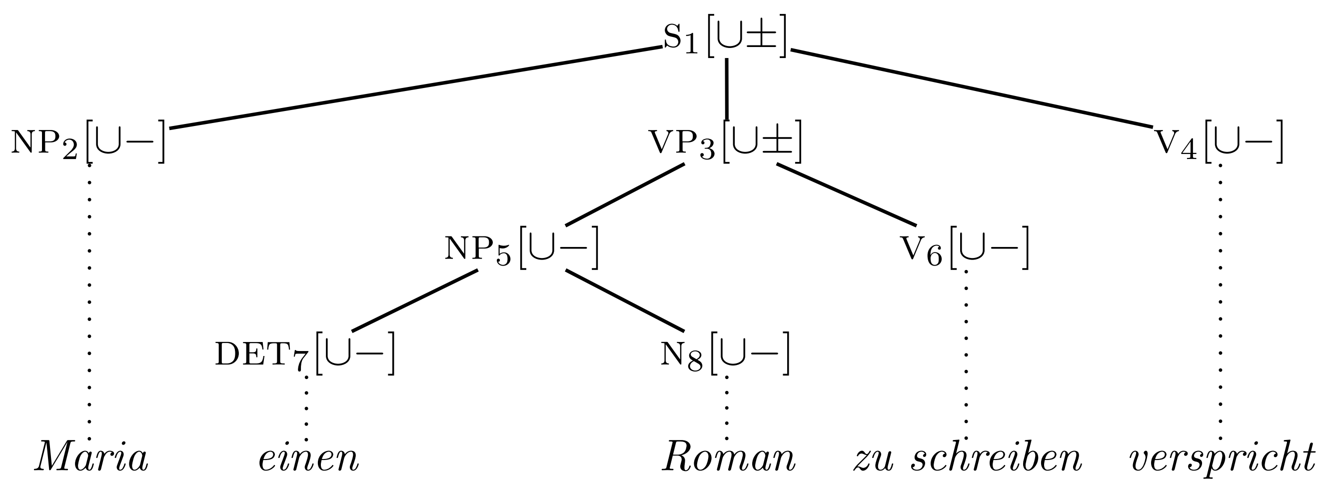


# Examples: Scrambling (1)

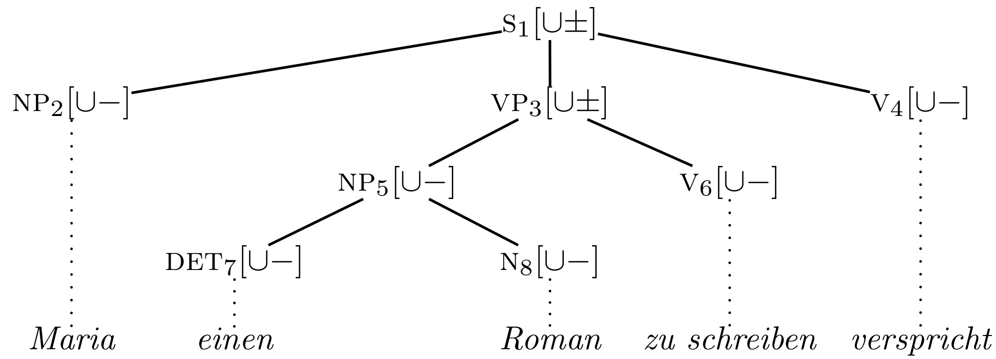
- the scrambling example again:

| | | | | | |
|---|---|---|---|---|---|
| *(dass)* | *einen* | *Roman* | *Maria* | *zu schreiben* | *verpricht.* |
| (that) | a | novel | Maria | to write | promises. |

"(that) Maria promises to write a novel."



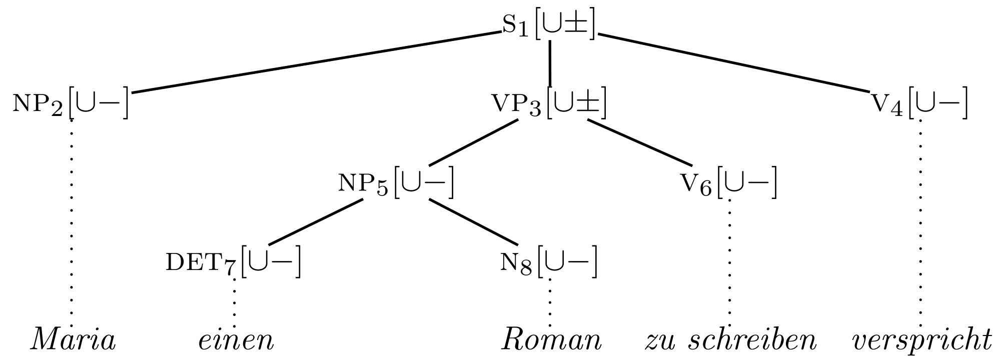
# *Examples: Scrambling (2)*




- see previous example:

$$\mathsf{dom}(\mathsf{VP}_3) \quad = \quad \langle \mathsf{NP}_5, \mathsf{V}_6 \rangle$$

# *Examples: Scrambling (3)*

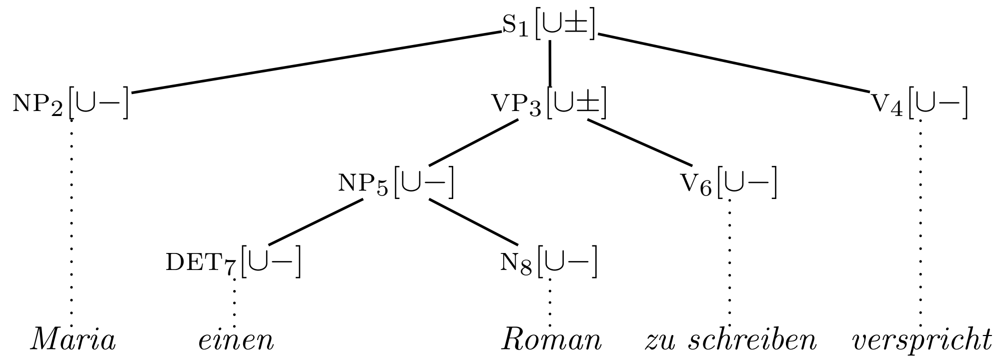



- $\text{VP}_3$: inserted or merged into $\text{S}_1$? first: inserted:
- inserted:

$$\begin{aligned} \text{dom}(\text{S}_1) \quad \in \quad & \langle \text{NP}_2 \rangle \cup^* \langle \text{VP}_3 \rangle \cup^* \langle \text{V}_4 \rangle \\ & = \{ \langle \text{NP}_2, \text{VP}_3, \text{V}_4 \rangle, \langle \text{VP}_3, \text{NP}_2, \text{V}_4 \rangle, \langle \text{NP}_2, \text{V}_4, \text{VP}_3 \rangle \} \end{aligned}$$

1. *Maria einen Roman zu schreiben verspricht* (canonical)
2. *einen Roman zu schreiben Maria verspricht* (intraposition)
3. *Maria verspricht einen Roman zu schreiben* (extraposition)

# Examples: Scrambling (4)

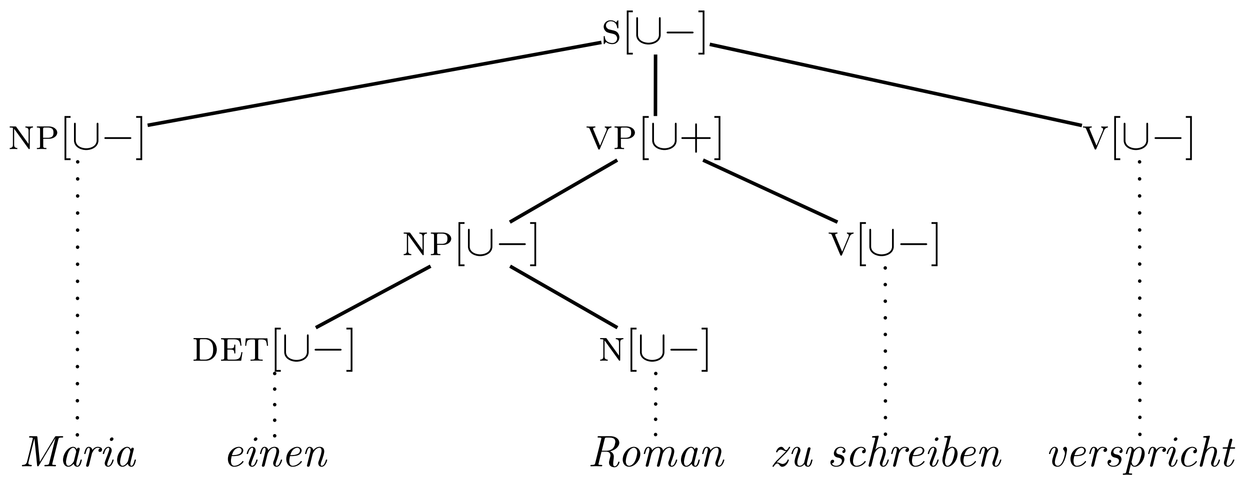



- now, merged:

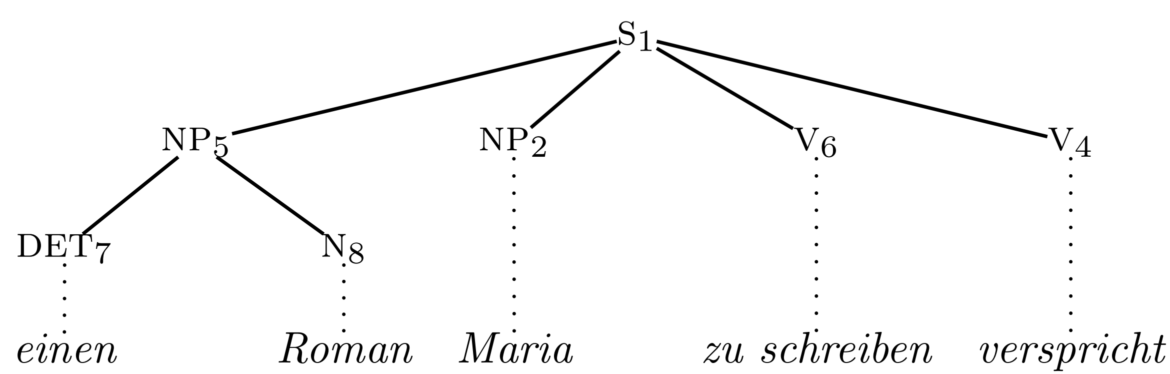
$$\begin{aligned} \mathsf{dom}(\mathsf{S}_1) \quad \in \quad & \langle \mathsf{NP}_2 \rangle \cup^* \mathsf{dom}(\mathsf{VP}_3) \cup^* \langle \mathsf{V}_4 \rangle \\ &= \langle \mathsf{NP}_2 \rangle \cup^* \langle \mathsf{NP}_5, \mathsf{V}_6 \rangle \cup^* \langle \mathsf{V}_4 \rangle \\ &= \{ \langle \mathsf{NP}_2, \mathsf{NP}_5, \mathsf{V}_6, \mathsf{V}_4 \rangle, \langle \mathsf{NP}_5, \mathsf{NP}_2, \mathsf{V}_6, \mathsf{V}_4 \rangle \} \end{aligned}$$

1. *Maria einen Roman zu schreiben verspricht* (canonical)
2. *einen Roman Maria zu schreiben verspricht* (scrambling)

# Examples: Scrambling (5)




- the resulting WOD tree is:



# *Examples: Partial VP extraposition*

| | | | | | |
|---|---|---|---|---|---|
| *(dass)* | *Maria* | *einen* | *Roman* | *verspricht* | *zu schreiben.* |
| (that) | Maria | a | novel | promises | to write. |

“(that) Maria promises to write a novel.”

- cannot be derived
- why? can only *insert*, or *merge*, but nothing in between

# *Commentary of Reape's approach*

- groundbreaking work (for HPSG and beyond) making for a better treatment of free word order
- ideas adopted e.g. in (Müller 1999), (Kathol 2000) for HSPG, (Bröker 1999) and (Gerdes/Kahane 2001) for DG, and also, well, TDG (Duchier/Debusmann 2001)
- use of unioned-feature to control the flattening is not fine-grained enough

# *Topological fields theory*

- the following approaches will borrow from topological fields theory
- traditional descriptive theory of German syntax (Herling 1821, Höhle 1986)
- sentences separated into topological fields:

| *Vorfeld* | *(* | *Mittelfeld* | *)* | *Nachfeld* |
|---|---|---|---|---|
| | *dass* | *Maria einen Roman* | *zu schreiben verspricht* | |
| | *dass* | *einen Roman Maria* | *zu schreiben verspricht* | |
| | *dass* | *Maria* | *verspricht* | *einen Roman zu schreiben* |
| *Maria* | *verspricht* | *einen Roman* | *zu schreiben* | |
| *einen Roman* | *verspricht* | *Maria* | *zu schreiben* | |

# *Kathol: HPSG*

- Kathol (1995, 2000)
- based on Reape's notion of word order domains
- dispenses with Reape's binary `unioned`- and `extra`-features
- instead, he associates domain objects directly with topological fields

# *Kathol's approach*

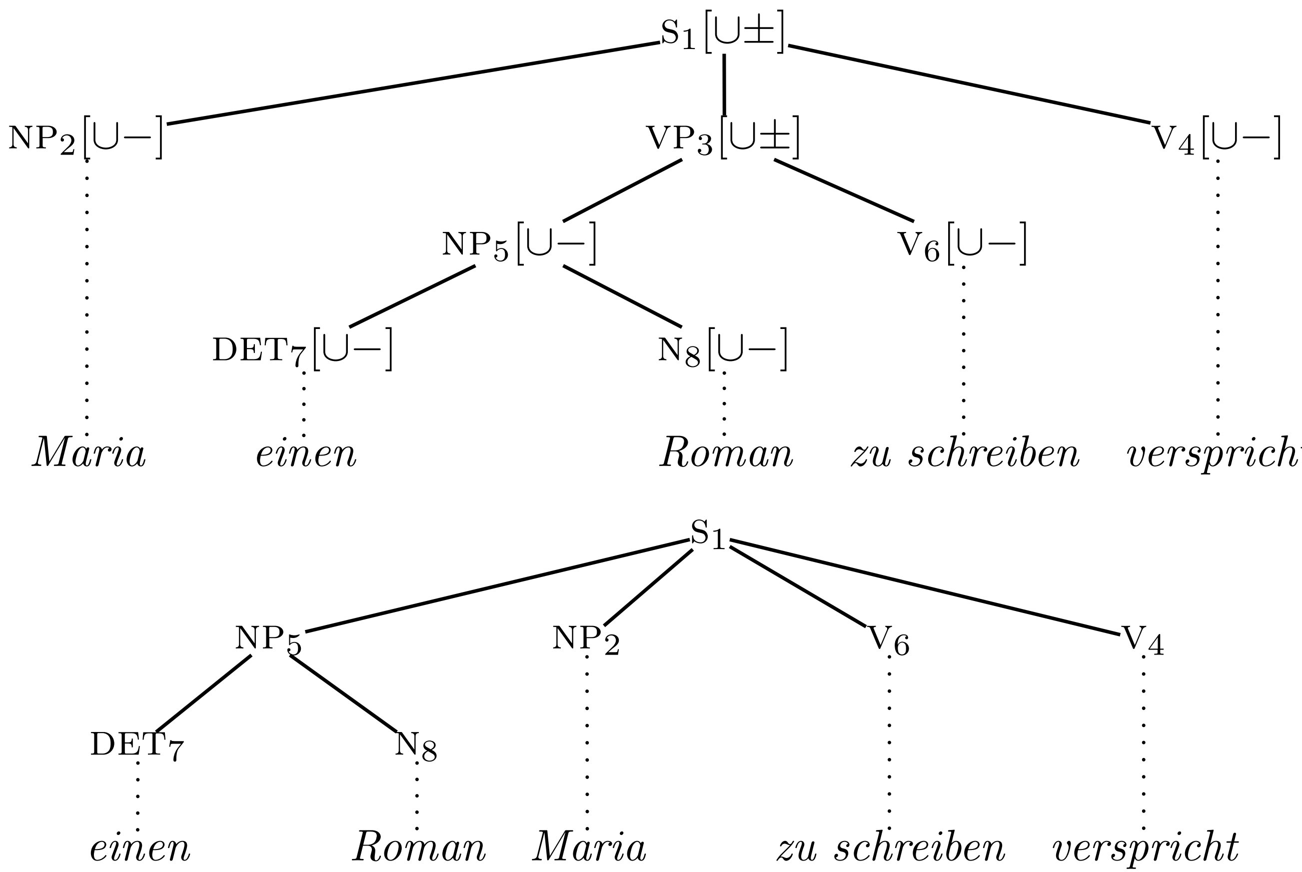
- make use of the following set of topological fields:

| field name | explanation |
|---|---|
| vf | Vorfeld |
| cf | complementizer field |
| mf | Mittelfeld |
| vc | verb cluster |
| nf | Nachfeld |

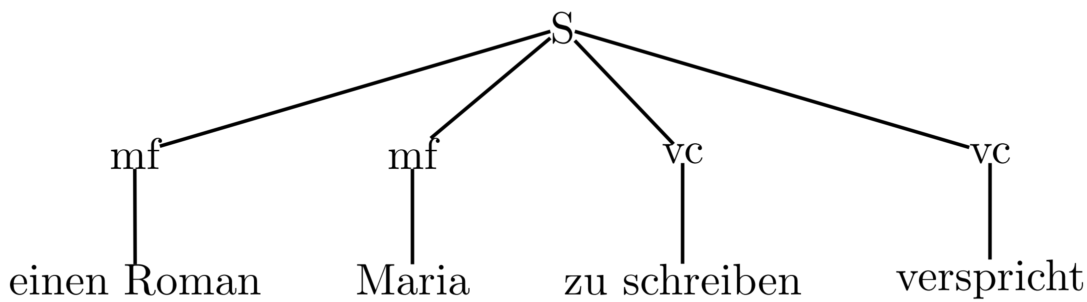
- LP rules replaced by 1) increased lexicalization, and 2) the Topological Linear Precedence Statement:

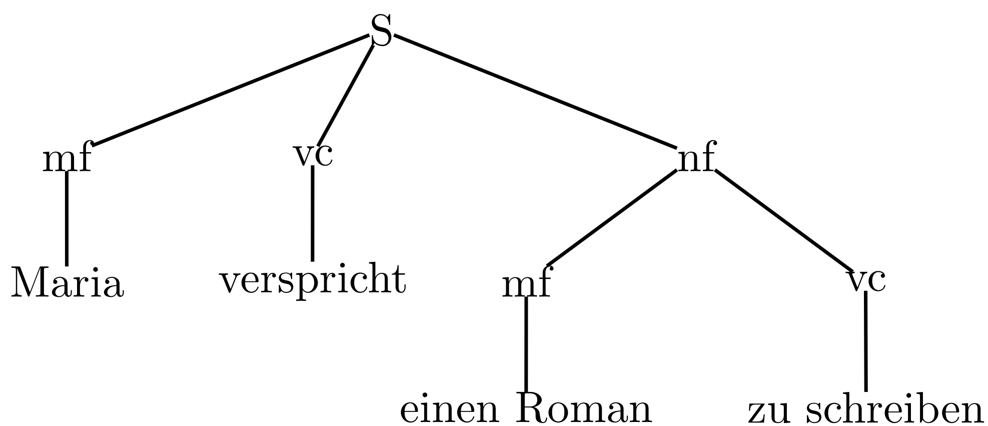
$$\mathsf{vf} \prec \mathsf{cf} \prec \mathsf{mf} \prec \mathsf{vc} \prec \mathsf{nf}$$

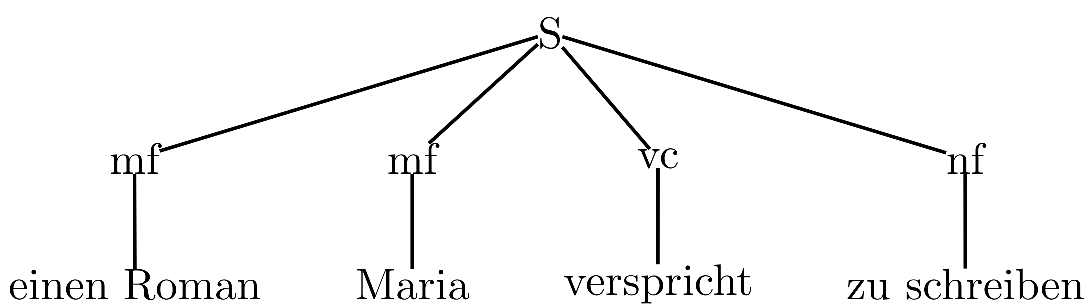
# Reape scrambling



# Kathol examples



# *Commentary of Kathol's approach*

- overcomes Reape's defects
- new: primitive notion of topological fields
- LP constraints order topological fields, not stated on categorial grounds

# Gerdes and Kahane: DG

- Gerdes/Kahane (2001)
- dependency-based
- called Topological Dependency Grammar (TDG)
- emerged at the same time as Duchier/Debusmann (2001) (also TDG)
- places itself in the context of Meaning Text Theory (MTT) (Melcuk 1988)
- syntactic module of MTT (correspondence between syntactical dependency trees and morphological strings)

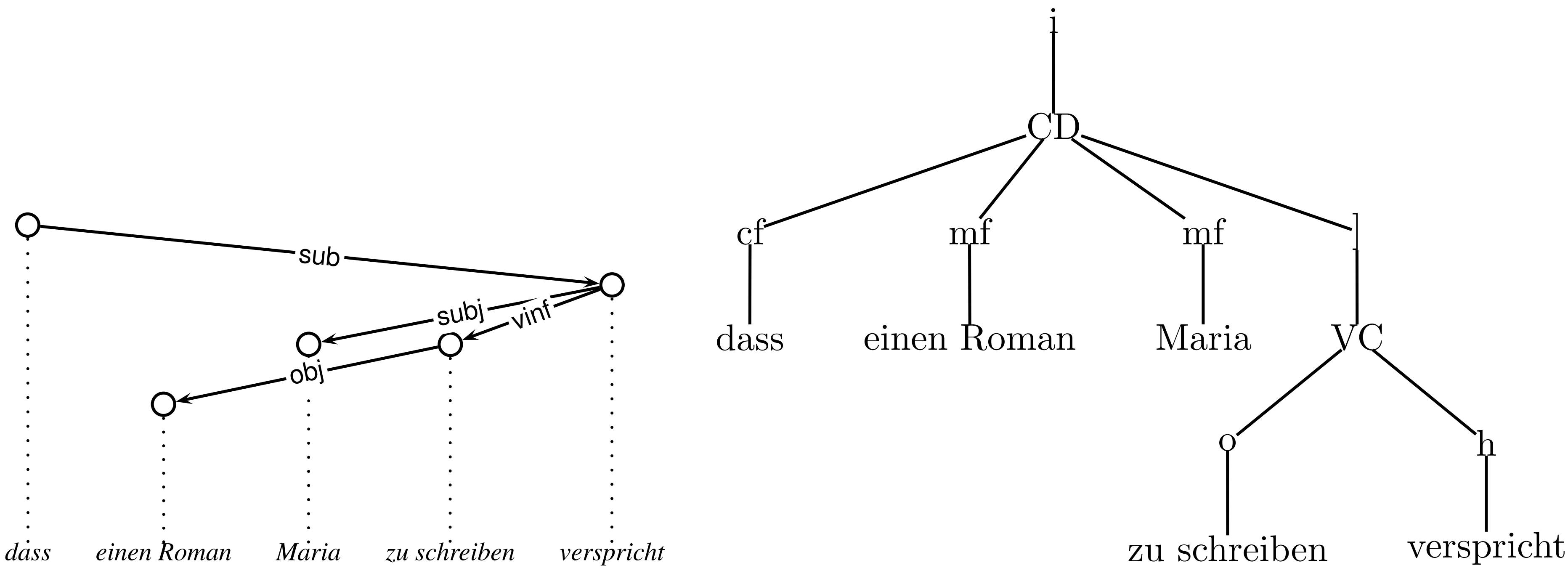
# *Gerdes and Kahane: structures*

- again two structures:
  1. unordered dependency tree
  2. topological phrase structure tree
- similar to Kathol, but dependency tree instead of phrase structure tree

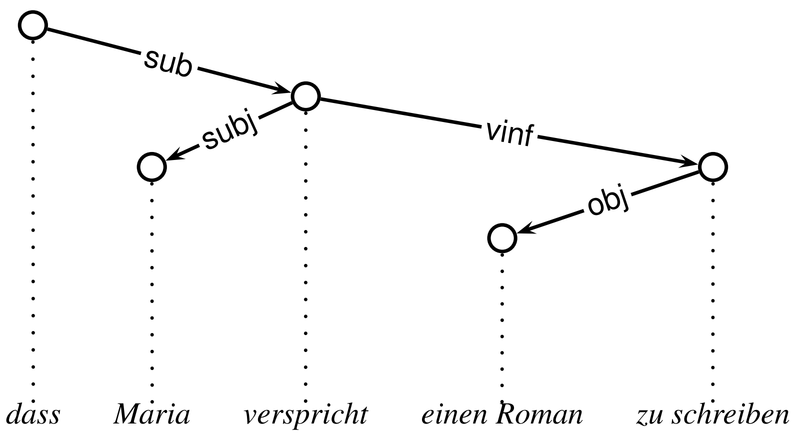
# *Gerdes and Kahane: Topological structure*

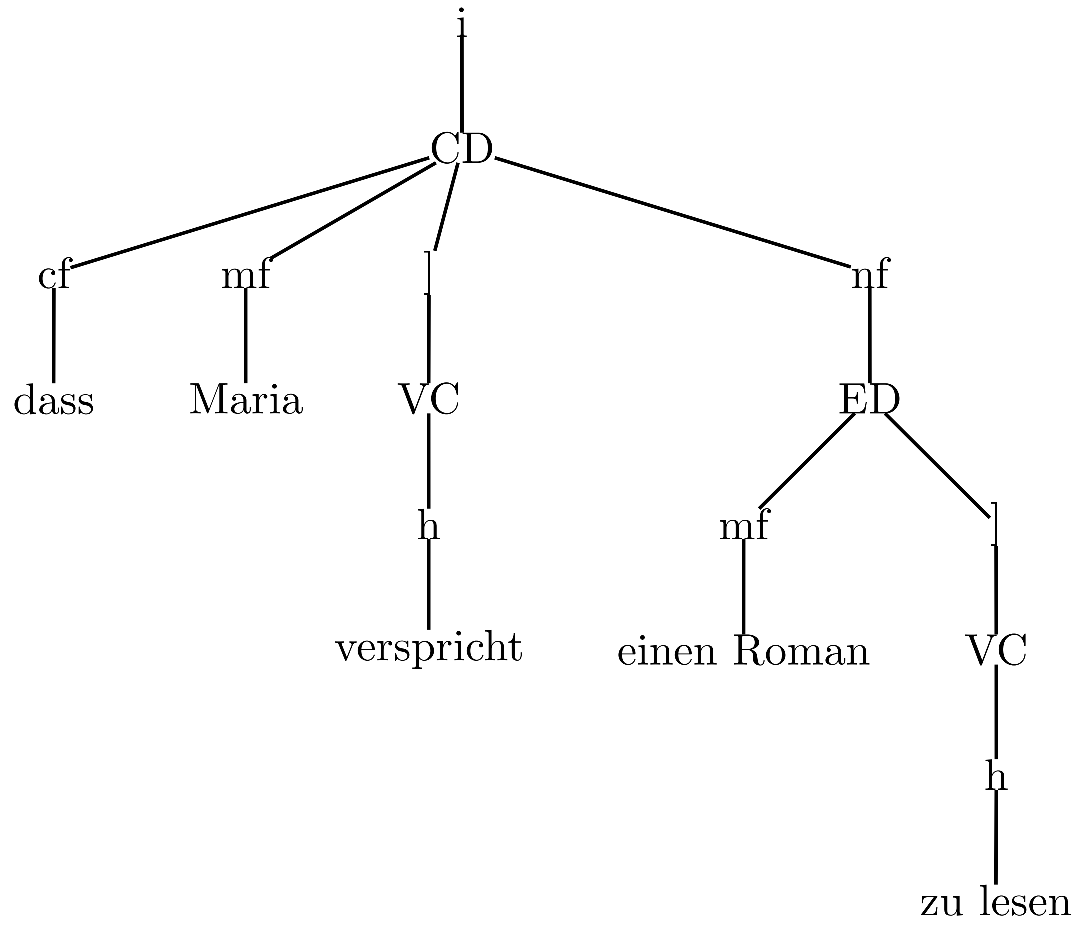
- the topological structure is made up of *domains*, *fields* and *words*
- domain: sequence of fields
- field: words and/or domains

# Gerdes and Kahane: Scrambling

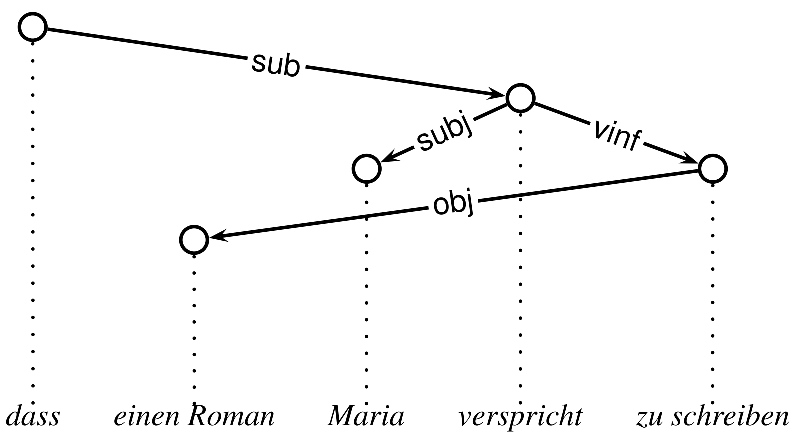


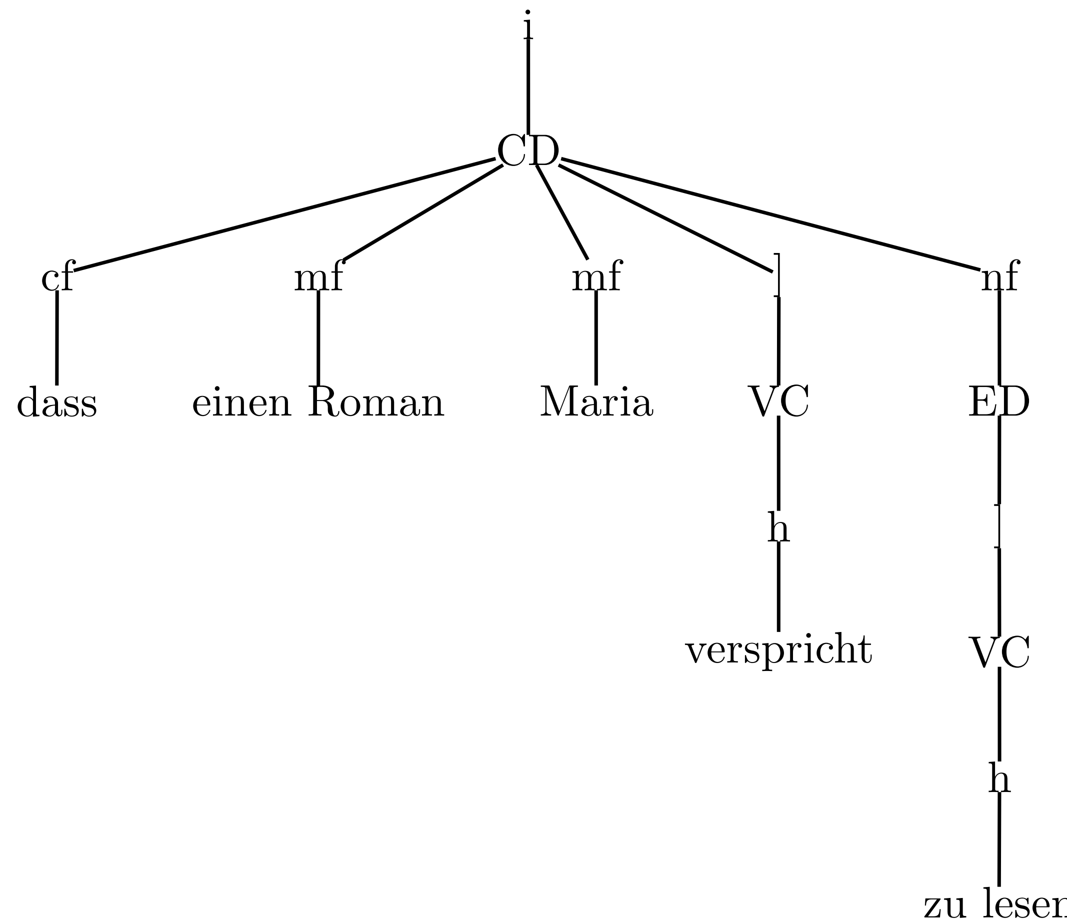
# Gerdes and Kahane: Full VP Extraposition



# Gerdes and Kahane: Partial VP Extraposition



# Gerdes and Kahane: Grammar definition

- grammar defined in a rule-based fashion using four sets of rules:
  1. domain creation rules
  2. domain description rules
  3. field description rules
  4. correspondence rules
- additionally, extraction is restricted using a *permeability order* on the domains

# Gerdes and Kahane: Grammar definition contd.

1. domain creation rules: $(\texttt{comp}, \mathsf{i}, \mathsf{CD}, \mathsf{cf})$
2. domain description: $\mathsf{CD} \rightarrow \mathsf{cf}, \mathsf{mf}, ], \mathsf{nf}$
3. field description: $(\mathsf{cf}, !), \ (\mathsf{mf}, *), \ \ldots$

$$\mathsf{CD} \quad \rightarrow \quad \mathsf{cf}!, \mathsf{mf}*, ]!, \mathsf{nf}?$$

4. correspondence: $(\mathsf{obj}, \texttt{v}, \texttt{n}, \mathsf{mf}, \mathsf{ED})$

# *Gerdes and Kahane: Permeability order*

- control extraction by permeability order:

$$\mathsf{VC} < \mathsf{ED} < \mathsf{CD} < \mathsf{MD}$$

- correspondence rule:

$$(\mathrm{obj}, \mathtt{v}, \mathtt{n}, \mathsf{mf}, \mathsf{ED})$$

indirect objects can be extracted out of everything $\leq$ ED in the permeability order, i.e. both out of verb clusters and embedded domains

# *Gerdes and Kahane: Commentary*

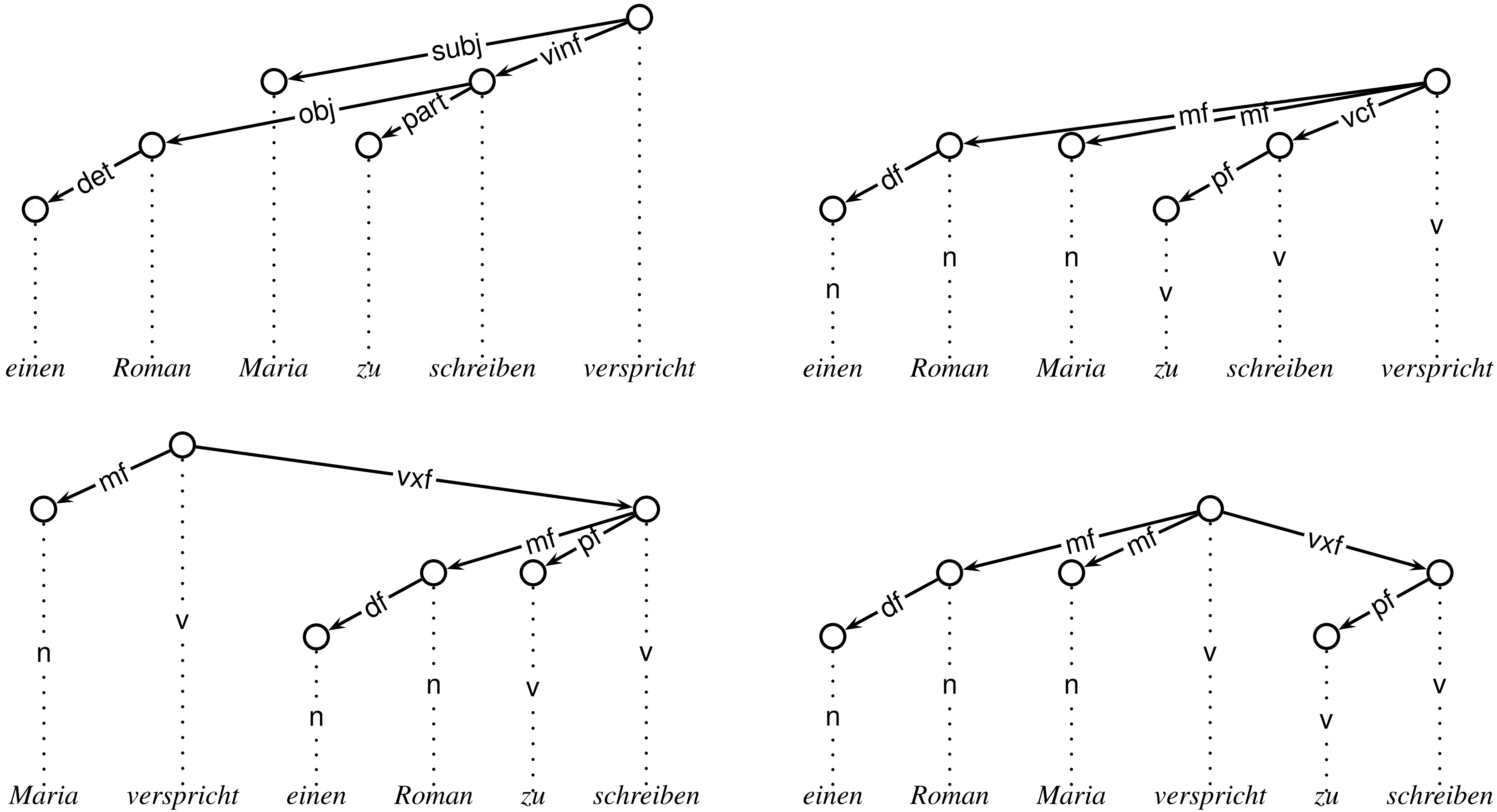
- very similar to TDG (Duchier/Debusmann 2001), as we will see shortly
- difference: topological phrase structure as opposed to topological dependency structure
- primitive notion of topological fields allows for concise statements of constraints
- claim: topological phrase structure realizes prosodic structure, allows to give an account of prosody
- parser available

# The TDG/XDG approach

- Duchier/Debusmann (2001)
- distinguishes two dimensions: ID tree (Immediate Dominance), and LP tree (Linear Precedence)
- dependency trees on both dimensions, sharing the same set of nodes, but having different edges
- valency used on both ID and LP dimensions
- ID tree unordered, LP tree ordered and projective
- LP tree is a flattening of the ID tree
- like Kathol: global order on the set of fields, e.g.:

$$\mathsf{d} < \mathsf{df} < \mathsf{n} < \mathsf{mf} < \mathsf{vcf} < \mathsf{pf} < \mathsf{v} < \mathsf{vxf}$$

# XDG: Examples



# *XDG, commentary*

- very similar to (Gerdes/Kahane 2001)
- differences: LP dimension modeled also by a dependency tree rather than a phrase structure tree
- handles scrambling, full VP extraposition and partial VP extraposition
- and more, as we will see next...