Topological Dependency Trees: A Constraint-Based Account of Linear Precedence

Denys Duchier
Programming Systems Lab
Universität des Saarlandes, Geb. 45
Postfach 15 11 50
66041 Saarbrücken, Germany
duchier@ps.uni-sb.de

Ralph Debusmann
Computational Linguistics
Universität des Saarlandes, Geb. 17
Postfach 15 11 50
66041 Saarbrücken, Germany
rade@coli.uni-sb.de

Abstract

We describe a new framework for dependency grammar, with a modular decomposition of immediate dependency and linear precedence. Our approach distinguishes two orthogonal yet mutually constraining structures: a *syntactic dependency tree* and a *topological dependency tree*. The syntax tree is non-projective and even non-ordered, while the topological tree is projective and partially ordered.

1 Introduction

Linear precedence in so-called free word order languages remains challenging for modern grammar formalisms. To address this issue, we propose a new framework for dependency grammar which supports the modular decomposition of immediate dependency and linear precedence. Duchier (1999) formulated a constraint-based axiomatization of dependency parsing which characterized well-formed syntax trees but ignored issues of word order. In this article, we develop a complementary approach dedicated to the treatment of linear precedence.

Our framework distinguishes two orthogonal, yet mutually constraining structures: a *syntactic dependency tree* (ID tree) and a *topological dependency tree* (LP tree). While edges of the ID tree are labeled by syntactic roles, those of the LP tree are labeled by topological fields (Bech, 1955). The shape of the LP tree is a flattening of the ID tree's obtained by allowing nodes to 'climb up' to land in an appropriate field at a host node where that field is available. Our theory of ID/LP trees is formulated in terms of (a) lexicalized constraints and (b) principles governing e.g. climbing conditions.

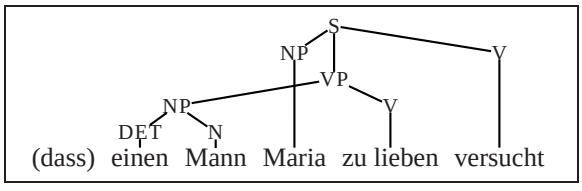
In Section 2 we discuss the difficulties presented by discontinuous constructions in free word order languages, and briefly touch on the limitations of Reape's (1994) popular theory of 'word order domains'. In Section 3 we introduce the concept of topological dependency tree. In Section 4 we outline the formal framework for our theory of ID/LP trees. Finally, in Section 5 we illustrate our approach with an account of the word-order phenomena in the verbal complex of German verb final sentences.

2 Discontinuous Constructions

In free word order languages, discontinuous constructions occur frequently. German, for example, is subject to *scrambling* and *partial extraposition*. In typical phrase structure based analyses, such phenomena lead to e.g. discontinuous VPs:

(1) (dass) einen Mann Maria zu lieben versucht
(that) a man_{acc} $Maria_{nom}$ *to love tries*

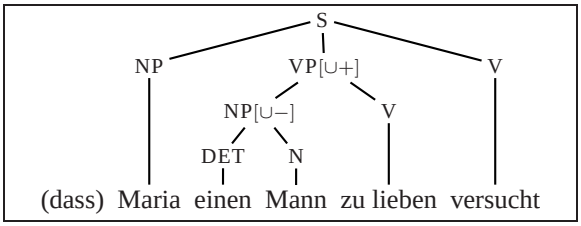
whose natural syntax tree exhibits crossing edges:



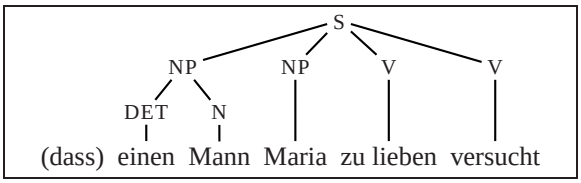
Since this is classically disallowed, discontinuous constituents must often be handled indirectly through grammar extensions such as traces.

Reape (1994) proposed the theory of *word order domains* which became quite popular in the HPSG community and inspired others such as Müller (1999) and Kathol (2000). Reape distinguished two orthogonal tree structures: (a) the unordered syntax tree, (b) the totally ordered tree of

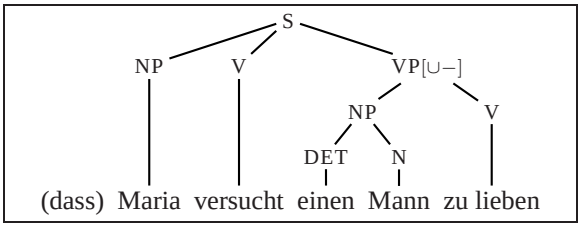
word order domains. The latter is obtained from the syntax tree by *flattening* using the operation of *domain union* to produce arbitrary interleavings. The boolean feature $[\cup\pm]$ of each node controls whether it must be flattened out or not. Infinitives in canonical position are assigned $[\cup+]$:



Thus, the above licenses the following tree of word order domains:



Extraposed infinitives are assigned $[\cup-]$:



As a consequence, Reape's theory correctly predicts scrambling (2,3) and full extraposition (4), but cannot handle the partial extraposition in (5):

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

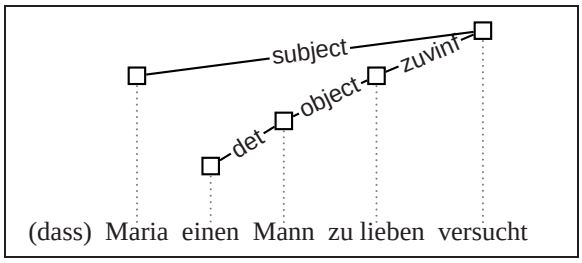
(3) (dass) einen Mann Maria zu lieben versucht

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

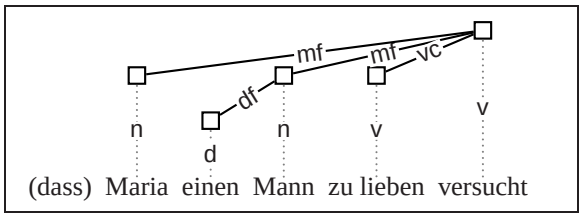
(5) (dass) Maria einen Mann versucht, zu lieben

3 Topological Dependency Trees

Our approach is based on dependency grammar. We also propose to distinguish two structures: (a) a tree of syntactic dependencies, (b) a tree of topological dependencies. The syntax tree (ID tree) is unordered and non-projective (i.e. it admits crossing edges). For display purposes, we pick an arbitrary linear arrangement:



The topological tree (LP tree) is partially ordered and projective:



Its edge labels are called *(external) fields* and are totally ordered: df $\prec$ mf $\prec$ vc. This induces a linear precedence among the daughters of a node in the LP tree. This precedence is partial because daughters with the same label may be freely permuted.

In order to obtain a linearization of a LP tree, it is also necessary to position each node with respect to its daughters. For this reason, each node is also assigned an *internal* field (d, n, or v) shown above on the vertical pseudo-edges. The set of internal and external fields is totally ordered: d $\prec$ df $\prec$ n $\prec$ mf $\prec$ vc $\prec$ v

Like Reape, our LP tree is a flattened version of the ID tree (Reape, 1994; Uszkoreit, 1987), but the flattening doesn't happen by 'unioning up'; rather, we allow each individual daughter to *climb up* to find an appropriate landing place. This idea is reminiscent of GB, but, as we shall see, proceeds rather differently.

4 Formal Framework

The framework underlying both ID and LP trees is the configuration of labeled trees under valency (and other) constraints. Consider a finite set $\mathcal{L}$ of edge labels, a finite set V of nodes, and $E \subseteq V \times V \times \mathcal{L}$ a finite set of directed labeled edges, such that (V, E) forms a tree. We write $w{-}\ell{\rightarrow}w'$ for an edge labeled ℓ from w to w'. We define the ℓ-daughters $\ell(w)$ of $w \in V$ as follows:

$$\ell(w) = \{w' \in V \mid w{-}\ell{\rightarrow}w' \in E\}$$

We write $\widehat{\mathcal{L}}$ for the set of valency specifications $\widehat{\ell}$ defined by the following abstract syntax:

$$\widehat{\ell} \quad ::= \quad \ell \mid \ell? \mid \ell* \qquad (\ell \in \mathcal{L})$$

A valency is a subset of $\widehat{\mathcal{L}}$. The tree (V, E) satisfies the valency assignment $\mathsf{valency} : V \to 2^{\widehat{\mathcal{L}}}$ if for all $w \in V$ and all $\ell \in \mathcal{L}$:

$$\begin{array}{rcl} \ell \in \mathsf{valency}(w) & \Rightarrow & |\ell(w)| = 1 \\ \ell? \in \mathsf{valency}(w) & \Rightarrow & |\ell(w)| \leq 1 \\ \ell* \in \mathsf{valency}(w) & \Rightarrow & |\ell(w)| \geq 0 \\ \text{otherwise} & \Rightarrow & |\ell(w)| = 0 \end{array}$$

4.1 ID Trees

An ID tree $(V, E_{\text{ID}}, \mathsf{lex}, \mathsf{cat}, \mathsf{valency}_{\text{ID}})$ consists of a tree (V, E_{ID}) with $E_{\text{ID}} \subseteq V \times V \times \mathcal{R}$, where the set $\mathcal{R}$ of edge labels (Figure 1) represents syntactic roles such as subject or vinf (bare infinitive argument). $\mathsf{lex} : V \to \textit{Lexicon}$ assigns a lexical entry to each node. An illustrative *Lexicon* is displayed in Figure 1 where the 2 features cats and $\mathsf{valency}_{\text{ID}}$ of concern to ID trees are grouped under table heading "Syntax". Finally, cat and $\mathsf{valency}_{\text{ID}}$ assign a category and an $\widehat{\mathcal{R}}$ valency to each node $w \in V$ and must satisfy:

$$\begin{array}{c} \mathsf{cat}(w) \in \mathsf{lex}(w).\mathsf{cats} \\ \mathsf{valency}_{\text{ID}}(w) = \mathsf{lex}(w).\mathsf{valency}_{\text{ID}} \end{array}$$

(V, E_{ID}) must satisfy the $\mathsf{valency}_{\text{ID}}$ assignment as described earlier. For example the lexical entry for *versucht* specifies (Figure 1):

$$\mathsf{valency}_{\text{ID}}(\textit{versucht}) = \{\mathsf{subject}, \mathsf{zuvinf}\}$$

Furthermore, (V, E_{ID}) must also satisfy the edge constraints stipulated by the grammar (see Figure 1). For example, for an edge w–det→w' to be licensed, w' must be assigned category *det* and both w and w' must be assigned the same agreement.[1]

4.2 LP Trees

An LP tree $(V, E_{\text{LP}}, \mathsf{lex}, \mathsf{valency}_{\text{LP}}, \mathsf{field}_{\text{ext}}, \mathsf{field}_{\text{int}})$ consists of a tree (V, E_{LP}) with $E_{\text{LP}} \subseteq V \times V \times \mathcal{F}_{\text{ext}}$, where the set $\mathcal{F}_{\text{ext}}$ of edge labels represents topological fields (Bech, 1955): df the determiner field, mf the 'Mittelfeld', vc the verbal complement field, xf the extraposition field. Features of lexical entries relevant to LP trees are grouped under table heading "Topology" in Figure 1. $\mathsf{valency}_{\text{LP}}$ assigns a $\widehat{\mathcal{F}_{\text{ext}}}$ valency to each node and is subject to the lexicalized constraint:

$$\mathsf{valency}_{\text{LP}}(w) = \mathsf{lex}(w).\mathsf{valency}_{\text{LP}}$$

(V, E_{LP}) must satisfy the $\mathsf{valency}_{\text{LP}}$ assignment as described earlier. For example, the lexical entry for *zu lieben*$_2$ specifies:

$$\mathsf{valency}_{\text{LP}}(\textit{zu lieben}_2) = \{\mathsf{mf}*, \mathsf{xf}?\}$$

which permits 0 or more mf edges and at most one xf edge; we say that it offers fields mf and xf. Unlike the ID tree, the LP tree must be projective.

The grammar stipulates a total order on $\mathcal{F}_{\text{ext}}$, thus inducing a partial linear precedence on each node's daughters. This order is partial because all daughters in the same field may be freely permuted: our account of *scrambling* rests on free permutations within the mf field. In order to obtain a linearization of the LP tree, it is necessary to specify the position of a node with respect to its daughters. For this reason each node is assigned an internal field in $\mathcal{F}_{\text{int}}$. The set $\mathcal{F}_{\text{ext}} \cup \mathcal{F}_{\text{int}}$ is totally ordered:

$$\mathsf{d} \prec \mathsf{df} \prec \mathsf{n} \prec \mathsf{mf} \prec \mathsf{vc} \prec \mathsf{v} \prec \mathsf{xf}$$

In what (external) field a node may land and what internal field it may be assigned is determined by assignments $\mathsf{field}_{\text{ext}} : V \to \mathcal{F}_{\text{ext}}$ and $\mathsf{field}_{\text{int}} : V \to \mathcal{F}_{\text{int}}$ which are subject to the lexicalized constraints:

$$\begin{array}{c} \mathsf{field}_{\text{ext}}(w) \in \mathsf{lex}(w).\mathsf{field}_{\text{ext}} \\ \mathsf{field}_{\text{int}}(w) \in \mathsf{lex}(w).\mathsf{field}_{\text{int}} \end{array}$$

For example, *zu lieben*$_1$ may only land in field vc (canonical position), and *zu lieben*$_2$ only in xf (extraposed position). The LP tree must satisfy:

$$w\text{–}\ell\text{→}w' \in E_{\text{LP}} \;\Rightarrow\; \ell = \mathsf{field}_{\text{ext}}(w')$$

Thus, whether an edge w–ℓ→w' is licensed depends both on $\mathsf{valency}_{\text{LP}}(w)$ and on $\mathsf{field}_{\text{ext}}(w')$. In other words: w must offer field ℓ and w' must accept it.

For an edge w–ℓ→w' in the ID tree, we say that w is the head of w'. For a similar edge in the LP

[1] Issues of agreement will not be further considered in this paper.

Grammar Symbols

$\mathcal{C} = \{\mathit{det}, \mathit{n}, \mathit{vfin}, \mathit{vinf}, \mathit{vpast}, \mathit{zuvinf}\}$	(Categories)
$\mathcal{R} = \{\mathsf{det}, \mathsf{subject}, \mathsf{object}, \mathsf{vinf}, \mathsf{vpast}, \mathsf{zuvinf}\}$	(Syntactic Roles)
$\mathcal{F}_{\text{ext}} = \{\mathsf{df}, \mathsf{mf}, \mathsf{vc}, \mathsf{xf}\}$	(External Topological Fields)
$\mathcal{F}_{\text{int}} = \{\mathsf{d}, \mathsf{n}, \mathsf{v}\}$	(Internal Topological Fields)
$\mathsf{d} \prec \mathsf{df} \prec \mathsf{n} \prec \mathsf{mf} \prec \mathsf{vc} \prec \mathsf{v} \prec \mathsf{xf}$	(Topological Ordering)

Edge Constraints

w——det⟶w'	$\Rightarrow$	$\mathsf{cat}(w') = \mathit{det}$	$\wedge\ \mathsf{agr}(w) = \mathsf{agr}(w')$
w—subject⟶w'	$\Rightarrow$	$\mathsf{cat}(w') = \mathit{n}$	$\wedge\ \mathsf{agr}(w) = \mathsf{agr}(w') \in \text{NOM}$
w—object⟶w'	$\Rightarrow$	$\mathsf{cat}(w') = \mathit{n}$	$\wedge\ \mathsf{agr}(w') \in \text{ACC}$
w——vinf⟶w'	$\Rightarrow$	$\mathsf{cat}(w') = \mathit{vinf}$	
w——vpast⟶w'	$\Rightarrow$	$\mathsf{cat}(w') = \mathit{vpast}$	
w——zuvinf⟶w'	$\Rightarrow$	$\mathsf{cat}(w') = \mathit{zuvinf}$	

Lexicon

Word	Syntax		Topology		
	cats	$\mathsf{valency}_{\text{ID}}$	$\mathsf{field}_{\text{int}}$	$\mathsf{field}_{\text{ext}}$	$\mathsf{valency}_{\text{LP}}$
einen	$\{\mathit{det}\}$	$\{\}$	$\{\mathsf{d}\}$	$\{\mathsf{df}\}$	$\{\}$
Mann	$\{\mathit{n}\}$	$\{\mathsf{det}\}$	$\{\mathsf{n}\}$	$\{\mathsf{mf}\}$	$\{\mathsf{df}?\}$
Maria	$\{\mathit{n}\}$	$\{\}$	$\{\mathsf{n}\}$	$\{\mathsf{mf}\}$	$\{\}$
lieben	$\{\mathit{vinf}\}$	$\{\mathsf{object}?\}$	$\{\mathsf{v}\}$	$\{\mathsf{vc}\}$	$\{\}$
geliebt	$\{\mathit{vpast}\}$	$\{\mathsf{object}?\}$	$\{\mathsf{v}\}$	$\{\mathsf{vc}\}$	$\{\}$
können$_1$	$\{\mathit{vinf}\}$	$\{\mathsf{vinf}\}$	$\{\mathsf{v}\}$	$\{\mathsf{vc}\}$	$\{\mathsf{vc}?\}$
können$_2$	$\{\mathit{vinf}, \mathit{vpast}\}$	$\{\mathsf{vinf}\}$	$\{\mathsf{v}\}$	$\{\mathsf{xf}\}$	$\{\mathsf{mf}*, \mathsf{vc}?, \mathsf{xf}?\}$
wird	$\{\mathit{vfin}\}$	$\{\mathsf{subject}, \mathsf{vinf}\}$	$\{\mathsf{v}\}$	$\{\mathsf{vc}\}$	$\{\mathsf{mf}*, \mathsf{vc}?, \mathsf{xf}?\}$
haben	$\{\mathit{vinf}\}$	$\{\mathsf{vpast}\}$	$\{\mathsf{v}\}$	$\{\mathsf{xf}\}$	$\{\mathsf{mf}*, \mathsf{vc}?, \mathsf{xf}?\}$
hat	$\{\mathit{vinf}\}$	$\{\mathsf{subject}, \mathsf{vpast}\}$	$\{\mathsf{v}\}$	$\{\mathsf{vc}\}$	$\{\mathsf{mf}*, \mathsf{vc}?, \mathsf{xf}?\}$
zu lieben$_1$	$\{\mathit{zuvinf}\}$	$\{\mathsf{object}?\}$	$\{\mathsf{v}\}$	$\{\mathsf{vc}\}$	$\{\}$
zu lieben$_2$	$\{\mathit{zuvinf}\}$	$\{\mathsf{object}?\}$	$\{\mathsf{v}\}$	$\{\mathsf{xf}\}$	$\{\mathsf{mf}*, \mathsf{xf}?\}$
versucht	$\{\mathit{vfin}\}$	$\{\mathsf{subject}, \mathsf{zuvinf}\}$	$\{\mathsf{v}\}$	$\{\mathsf{vc}\}$	$\{\mathsf{mf}*, \mathsf{vc}?, \mathsf{xf}?\}$

Figure 1: Grammar Fragment

tree, we say that w is the host of w' or that w' lands on w. The shape of the LP tree is a flattened version of the ID tree which is obtained by allowing nodes to *climb up* subject to the following principles:

Principle 1 *a node must land on a transitive head*[2]

Principle 2 *it may not climb through a barrier*

We will not elaborate the notion of barrier which is beyond the scope of this article, but, for example, a noun will prevent a determiner from climbing through it, and finite verbs are typically general barriers.

[2]This is Bröcker's terminology and means a node in the transitive closure of the head relation.

Principle 3 *a node must land on, or climb higher than, its head*

Subject to these principles, a node w' may climb up to any host w which offers a field licensed by $\mathsf{field}_{\text{ext}}(w')$.

Definition. *An* ID/LP *analysis is a tuple* $(V, E_{\text{ID}}, E_{\text{LP}}, \mathsf{lex}, \mathsf{cat}, \mathsf{valency}_{\text{ID}}, \mathsf{valency}_{\text{LP}}, \mathsf{field}_{\mathit{ext}}, \mathsf{field}_{\mathit{int}})$ *such that* $(V, E_{\text{ID}}, \mathsf{lex}, \mathsf{cat}, \mathsf{valency}_{\text{ID}})$ *is an* ID *tree and* $(V, E_{\text{LP}}, \mathsf{lex}, \mathsf{valency}_{\text{LP}}, \mathsf{field}_{\mathit{ext}}, \mathsf{field}_{\mathit{int}})$ *is an* LP *tree and all principles are satisfied.*

Our approach has points of similarity with (Bröker, 1999) but eschews modal logic in favor of a simpler and arguably more perspicuous constraint-based formulation. It is also related

to the lifting rules of (Kahane et al., 1998), but where they choose to stipulate rules that license liftings, we opt instead for placing constraints on otherwise unrestricted climbing.

5 German Verbal Phenomena

We now illustrate our theory by applying it to the treatment of word order phenomena in the verbal complex of German verb final sentences. We assume the grammar and lexicon shown in Figure 1. These are intended purely for didactic purposes and we extend for them no claim of linguistic adequacy.

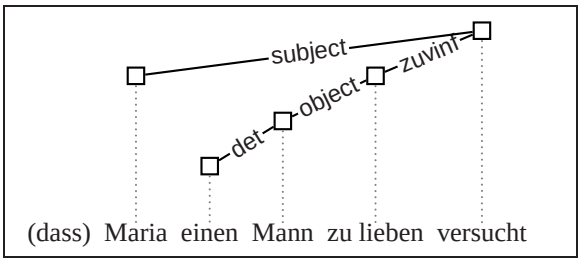
5.1 VP Extraposition

Control verbs like *versuchen* or *versprechen* allow their *zu*-infinitival complement to be optionally extraposed. This phenomenon is also known as optional coherence.

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

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

Both examples share the following ID tree:



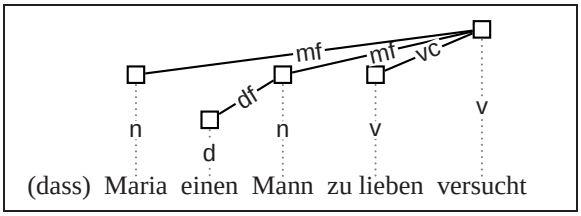
Optional extraposition is handled by having two lexical entries for *zu lieben*. One requires it to land in canonical position:

$$\mathsf{field}_{\text{ext}}(\textit{zu lieben}_1) = \{\mathsf{vc}\}$$

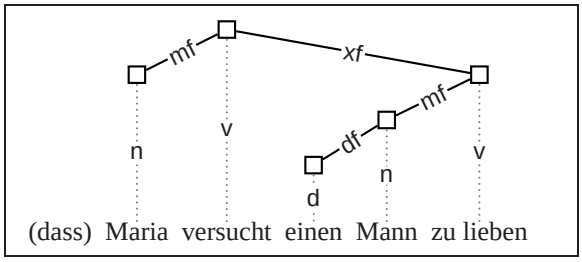
the other requires it to be extraposed:

$$\mathsf{field}_{\text{ext}}(\textit{zu lieben}_2) = \{\mathsf{xf}\}$$

In the canonical case, *zu lieben*$_1$ does not offer field mf and *einen Mann* must climb to the finite verb:



In the extraposed case, *zu lieben*$_2$ itself offers field mf:

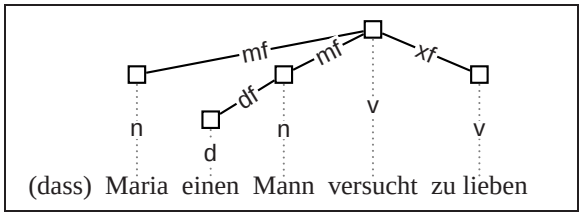


5.2 Partial VP Extraposition

In example (8), the *zu*-infinitive *zu lieben* is extraposed to the right of its governing verb *versucht*, but its nominal complement *einen Mann* remains in the Mittelfeld:

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

In our account, *Mann* is restricted to land in an mf field which both extraposed *zu lieben*$_2$ and finite verb *versucht* offer. In example (8) the nominal complement simply climbed up to the finite verb:



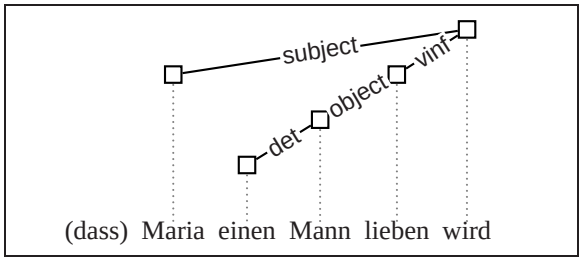
5.3 Obligatory Head-final Placement

Verb clusters are typically head-final in German: non-finite verbs precede their verbal heads.

(9) (dass) Maria einen Mann lieben wird
(that) Maria$_{nom}$ *a man*$_{acc}$ *love will*

(10)$*$(dass) Maria einen Mann wird lieben

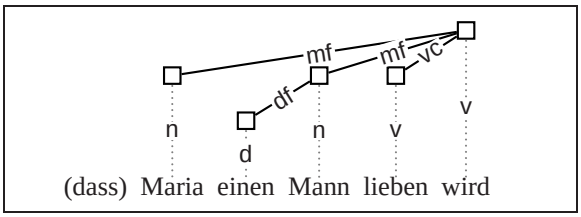
The ID tree for (9) is:



The lexical entry for the bare infinitive *lieben* requires it to land in a vc field:

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

therefore only the following LP tree is licensed:[3]



where mf $\prec$ vc $\prec$ v, and subject and object, both in field mf, remain mutually unordered. Thus we correctly license (9) and reject (10).

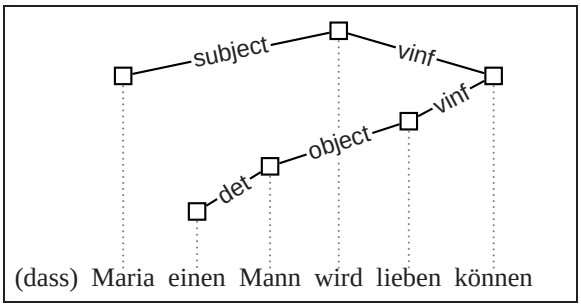
5.4 Optional Auxiliary Flip

In an auxiliary flip construction (Hinrichs and Nakazawa, 1994), the verbal complement of an auxiliary verb, such as *haben* or *werden*, follows rather than precedes its head. Only a certain class of bare infinitive verbs can land in extraposed position. As we illustrated above, main verbs do not belong to this class; however, modals such as *können* do, and may land in either canonical (11) or in extraposed (12) position. This behavior is called 'optional auxiliary flip'.

(11) (dass) Maria einen Mann lieben können wird
(that) Maria a man love can will
(that) Maria will be able to love a man

(12) (dass) Maria einen Mann wird lieben können

Both examples share the following ID tree:



Our grammar fragment describes optional auxiliary flip constructions in two steps:

- *wird* offers both vc and xf fields:

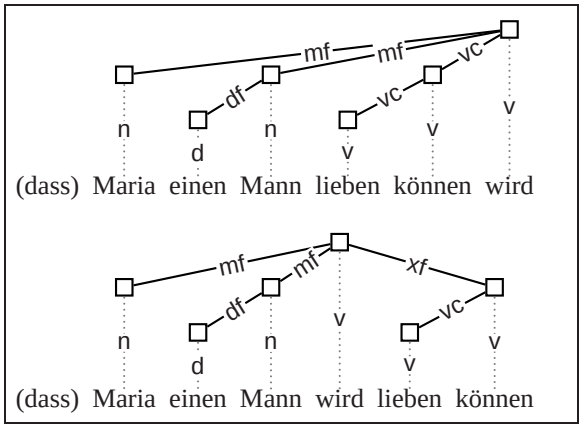
$$\mathsf{valency}_{\text{ID}}(\textit{wird}) = \{\mathsf{mf}*, \mathsf{vc}?, \mathsf{xf}?\}$$

- *können* has two lexical entries, one canonical and one extraposed:

$$\mathsf{field}_{\text{ext}}(\textit{können}_1) = \{\mathsf{vc}\}$$
$$\mathsf{field}_{\text{ext}}(\textit{können}_2) = \{\mathsf{xf}\}$$

[3]It is important to notice that there is no spurious ambiguity concerning the topological placement of *Mann*: *lieben* in canonical position does not offer field mf; therefore *Mann* must climb to the finite verb.

Thus we correctly account for examples (11) and (12) with the following LP trees:



The astute reader will have noticed that other LP trees are licensed for the earlier ID tree: they are considered in the section below.

5.5 V-Projection Raising

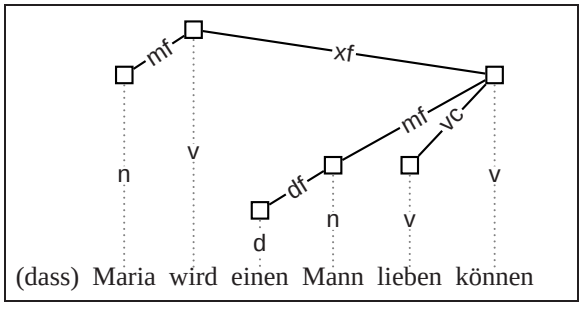
This phenomenon related to auxiliary flip describes the case where non-verbal material is interspersed in the verb cluster:

(13) (dass) Maria wird einen Mann lieben können

(14)$*$(dass) Maria lieben einen Mann können wird

(15)$*$(dass) Maria lieben können einen Mann wird

The ID tree remains as before. The NP *einen Mann* must land in a mf field. *lieben* is in canonical position and thus does not offer mf, but both extraposed *können*$_2$ and finite verb *wird* do. Whereas in (12), the NP climbed up to *wird*, in (13) it climbs only up to *können*.



(14) is ruled out because *können* must be in the vc of *wird*, therefore *lieben* must be in the vc of *können*, and *einen Mann* must be in the mf of *wird*. Therefore, *einen Mann* must precede both *lieben* and *können*. Similarly for (15).

5.6 Intermediate Placement

The *Zwischenstellung* construction describes cases where the auxiliary has been flipped but its verbal argument remains in the Mittelfeld. These are the remaining linearizations predicted by our theory for the running example started above:

(16) (dass) Maria einen Mann lieben wird können

(17) (dass) einen Mann Maria lieben wird können

where *lieben* has climbed up to the finite verb.

5.7 Obligatory Auxiliary Flip

Substitute infinitives (Ersatzinfinitiv) are further examples of extraposed verbal forms. A substitute infinitive exhibits bare infinitival inflection, yet acts as a complement of the perfectizer *haben*, which syntactically requires a past participle. Only modals, AcI-verbs such as *sehen* and *lassen*, and the verb *helfen* can appear in substitute infinitival inflection.

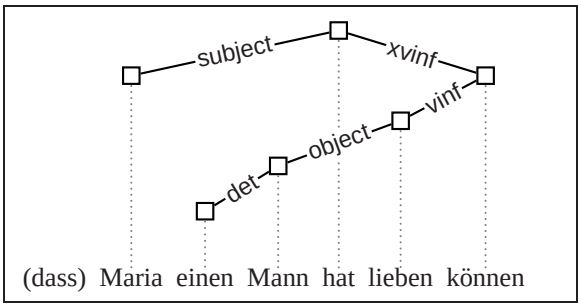
A substitute infinitive cannot land in canonical position; it must be extraposed: an auxiliary flip involving a substitute infinitive is called an 'obligatory auxiliary flip'.

(18) (dass) Maria einen Mann hat lieben können
(that) Maria a man has love can
(that) Maria was able to love a man

(19) (dass) Maria hat einen Mann lieben können

(20)$*$(dass) Maria einen Mann lieben können hat

These examples share the ID tree:



hat subcategorizes for a verb in past participle inflection because:

$$\mathsf{valency}_{\mathrm{ID}}(\textit{hat}) = \{\mathsf{subject}, \mathsf{vpast}\}$$

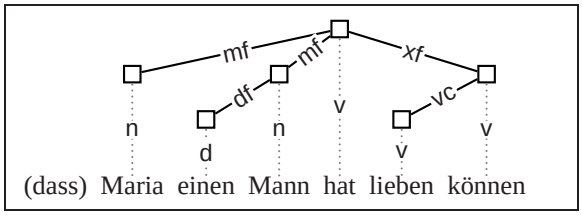
and the edge constraint for w–vpast→w' requires:

$$\mathsf{cat}(w') = \textit{vpast}$$

This is satisfied by *können*$_2$ which insists on being extraposed, thus ruling (20) out:

$$\mathsf{field}_{\mathrm{ext}}(\textit{können}_2) = \{\mathsf{xf}\}$$

Example (18) has LP tree:



In (18) *einen Mann* climbs up to *hat*, while in (19) it only climbs up to *können*.

5.8 Double Auxiliary Flip

Double auxiliary flip constructions occur when an auxiliary is an argument of another auxiliary. Each extraposed verb form offers both vc and mf: thus there are more opportunities for verbal and nominal arguments to climb to.

(21) (dass) Maria wird haben einen Mann lieben können
(that) Maria will have been able to love a man

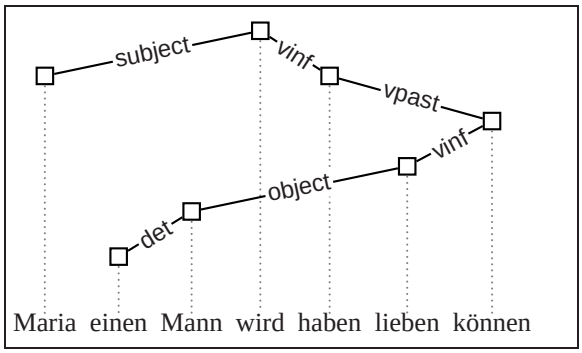
(22) (dass) Maria einen Mann wird haben lieben können

(23) (dass) Maria wird einen Mann lieben haben können

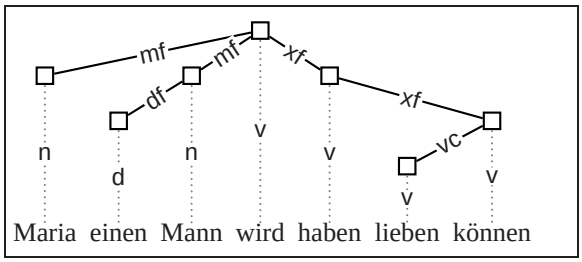
(24) (dass) Maria einen Mann wird lieben haben können

(25) (dass) Maria einen Mann lieben wird haben können

These examples have ID tree:



and (22) obtains LP tree:



5.9 Obligatory Coherence

Certain verbs like *scheint* require their argument to appear in canonical (or coherent) position.

(26) (dass) Maria einen Mann zu lieben scheint
(that) Maria a man to love seems
(that) Maria seems to love a man

(27)$*$(dass) Maria einen Mann scheint, zu lieben

Obligatory coherence may be enforced with the following constraint principle: if w is an obligatory coherence verb and w' is its verbal argument, then w' must land in w's vc field. Like barriers, the expression of this principle in our grammatical formalism falls outside the scope of the present article and remains the subject of active research.[4]

6 Conclusions

In this article, we described a treatment of linear precedence that extends the constraint-based framework for dependency grammar proposed by Duchier (1999). We distinguished two orthogonal, yet mutually constraining tree structures: unordered, non-projective ID trees which capture purely syntactic dependencies, and ordered, projective LP trees which capture topological dependencies. Our theory is formulated in terms of (a) lexicalized constraints and (b) principles which govern 'climbing' conditions.

We illustrated this theory with an application to the treatment of word order phenomena in the verbal complex of German verb final sentences, and demonstrated that these traditionally challenging phenomena emerge naturally from our simple and elegant account.

Although we provided here an account specific to German, our framework intentionally permits the definition of arbitrary language-specific topologies. Whether this proves linguistically adequate in practice needs to be substantiated in future research.

Characteristic of our approach is that the formal presentation defines valid analyses as the solutions of a constraint satisfaction problem which is amenable to efficient processing through constraint propagation. A prototype was implemented in Mozart/Oz and supports a parsing mode as well as a mode generating all licensed linearizations for a given input. It was used to prepare all examples in this article.

While the preliminary results presented here are encouraging and demonstrate the potential of our approach to linear precedence, much work remains to be done to extend its coverage and to arrive at a cohesive and comprehensive grammar formalism.

[4] we also thank an anonymous reviewer for pointing out that our grammar fragment does not permit intraposition

References

Gunnar Bech. 1955. *Studien über das deutsche Verbum infinitum*. 2nd unrevised edition published 1983 by Max Niemeyer Verlag, Tübingen (Linguistische Arbeiten 139).

Norbert Bröker. 1999. *Eine Dependenzgrammatik zur Kopplung heterogener Wissensquellen*. Linguistische Arbeiten 405. Max Niemeyer Verlag, Tübingen/FRG.

Denys Duchier. 1999. Axiomatizing dependency parsing using set constraints. In *Sixth Meeting on the Mathematics of Language*, Orlando/FL, July.

Erhard Hinrichs and Tsuneko Nakazawa. 1994. Linearizing AUXs in German verbal complexes. In Nerbonne et al. (Nerbonne et al., 1994), pages 11–37.

Sylvain Kahane, Alexis Nasr, and Owen Rambow. 1998. Pseudo-projectivity: a polynomially parsable non-projective dependency grammar. In *Proc. ACL/COLING'98*, pages 646–52, Montréal.

Andreas Kathol. 2000. *Linear Syntax*. Oxford University Press.

Igor Melćuk. 1988. *Dependency Syntax: Theory and Practice*. The SUNY Press, Albany, N.Y.

Stefan Müller. 1999. *Deutsche Syntax deklarativ. Head-Driven Phrase Structure Grammar für das Deutsche*. Linguistische Arbeiten 394. Max Niemeyer Verlag, Tübingen/FRG.

John Nerbonne, Klaus Netter, and Carl Pollard, editors. 1994. *German in Head-Driven Phrase Structure Grammar*. CSLI, Stanford/CA.

Mike Reape. 1994. Domain union and word order variation in German. In Nerbonne et al. (Nerbonne et al., 1994), pages 151–197.

Hans Uszkoreit. 1987. *Word Order and Constituent Structure in German*. CSLI, Stanford/CA.