Developing a TT-MCTAG for German with an RCG-based Parser

Laura Kallmeyer, Timm Lichte, Wolfgang Maier,
Yannick Parmentier*, Johannes Dellert

University of Tübingen, Germany
*CNRS-LORIA, France

LREC 2008, 28.05.2008

Aims and scope

Presentation of an implementation framework for a German TAG-based grammar

- How to design and maintain a grammatical resource ? (i.e., a German TT-MCTAG)
- How to connect this with a (2-layered) lexical resource?
- How to parse German using these resources?

Outline:

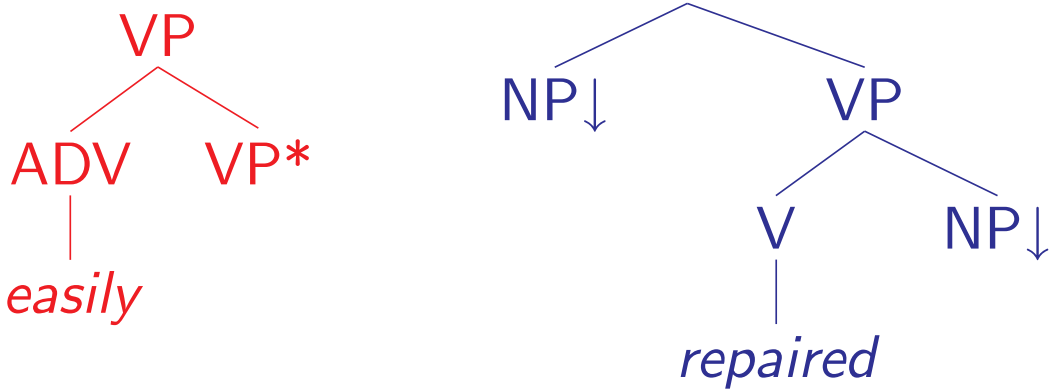
1. The formalism: TAG and TT-MCTAG
2. The implementation framework: XMG and TuLiPA
3. The grammar: GerTT

Tree-Adjoining Grammar - Basics

A **Tree Adjoining Grammar (TAG)** is a set of elementary trees:

- a finite set of initial trees
- a finite set of auxiliary trees

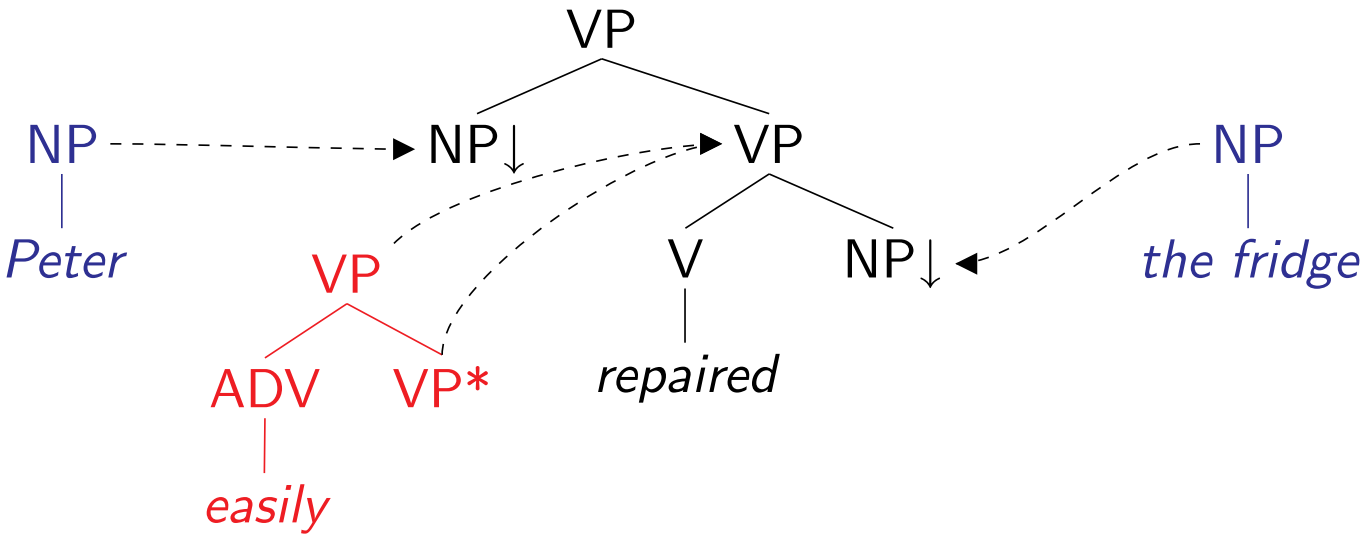
E.g.:



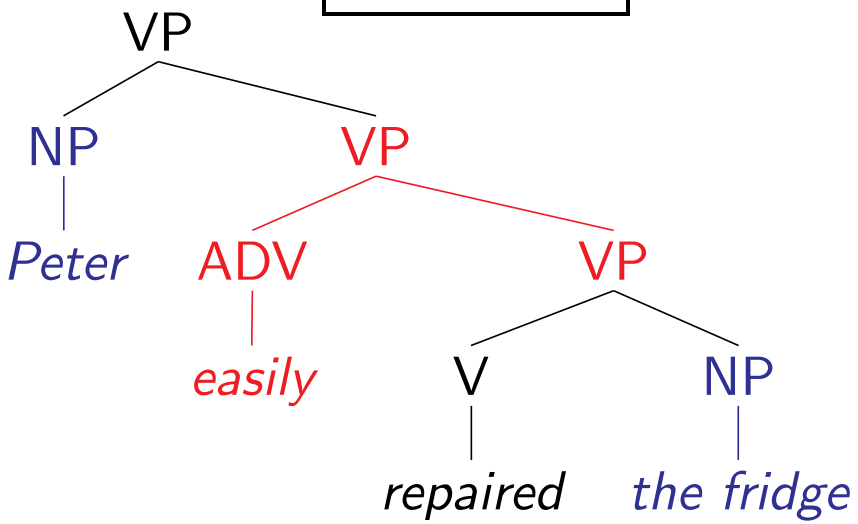
Combinatorial operations:

- substitution: replacing a non-terminal leaf with an initial tree
- adjunction: replacing an internal node with an auxiliary tree

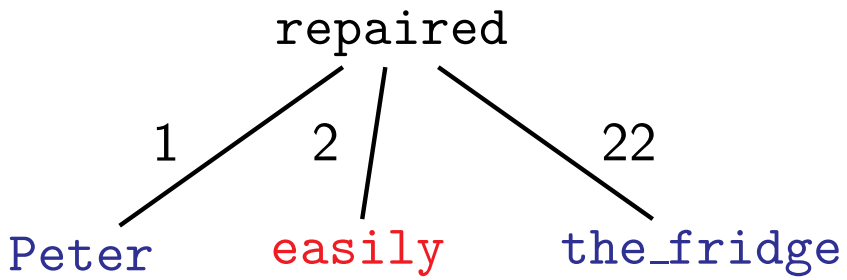
Tree-Adjoining Grammar - Example



derived tree



derivation tree



Tree-Adjoining Grammar - Basics

TAGs are **mildly context-sensitive:**

1. Polynomial time parsing complexity
2. Generation of limited crossing dependencies
3. Constant growth property (semilinearity)

Large TAG grammars:

- English and Korean (XTAG, UPenn)
- French TAG (Benoit Crabbé's PhD-thesis)
- ...

Why not TAG for German?

The order of complements (and adjuncts) of a verb is flexible.

(1) Peter liebt Susi.
1: `Peter loves Susi`
2: `Susi loves Peter`

(2) dass Peter heute den Kühlschrank repariert hat
dass den Kühlschrank heute Peter repariert hat
...
('that Peter has repaired the fridge today')

TAG is inappropriate for German, because it is:

- not powerful enough for some constructions (i.e., coherent constructions)
- not descriptively adequat (i.e., one elementary tree for each permutation)

Why not TAG for German?

The order of complements (and adjuncts) of a verb is flexible.

(1) Peter liebt Susi.
1: `Peter loves Susi`
2: `Susi loves Peter`

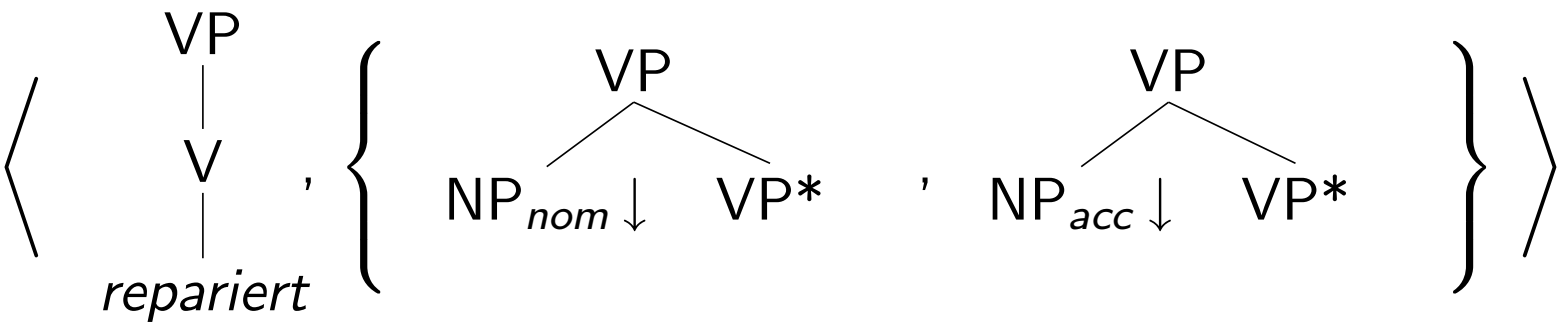
(2) dass Peter heute den Kühlschrank repariert hat
dass den Kühlschrank heute Peter repariert hat
...
('that Peter has repaired the fridge today')

TAG is inappropriate for German, because it is:

- not powerful enough for some constructions
  (i.e., coherent constructions)
- not descriptively adequat
  (i.e., one elementary tree for each permutation)

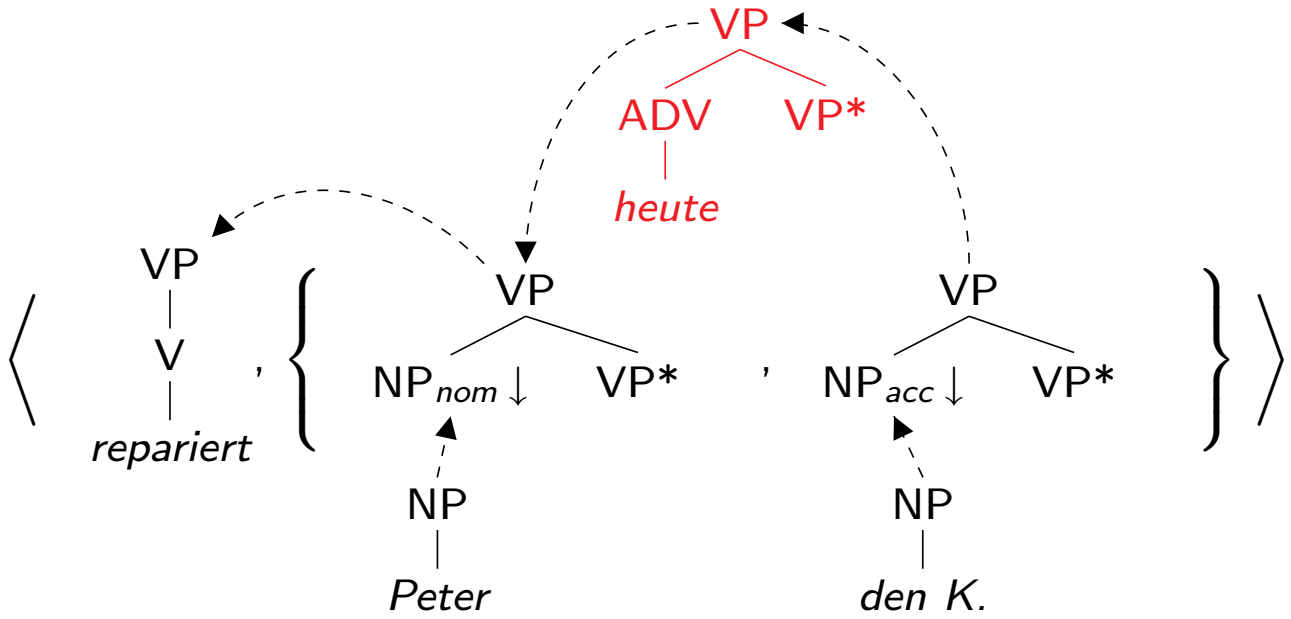
TT-MCTAG: a TAG-extension for German

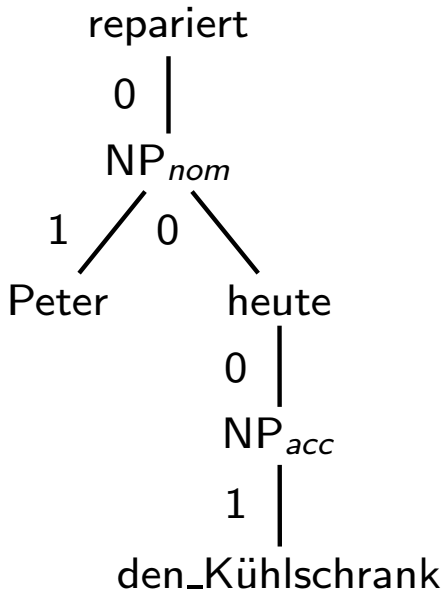
- Multi-Component TAG (MCTAG) with shared-nodes locality
- Elementary structures are **tuples** $\langle \gamma, \{\beta_1, ..., \beta_n\} \rangle$:
  - a lexicalized elementary tree γ (the head tree)
  - a tree set $\{\beta_1, ..., \beta_n\}$ (the complement trees)
- **Meaning of tree tuples:** During derivation, the β-trees have to attach to the γ-tree (via node sharing).
- **Node sharing:** In the derivation tree,
  1. a β-tree must either be the immediate daughter of its γ-tree,
  2. or the β-tree must be connected to the daughter of the γ-tree via a chain of root adjunctions.



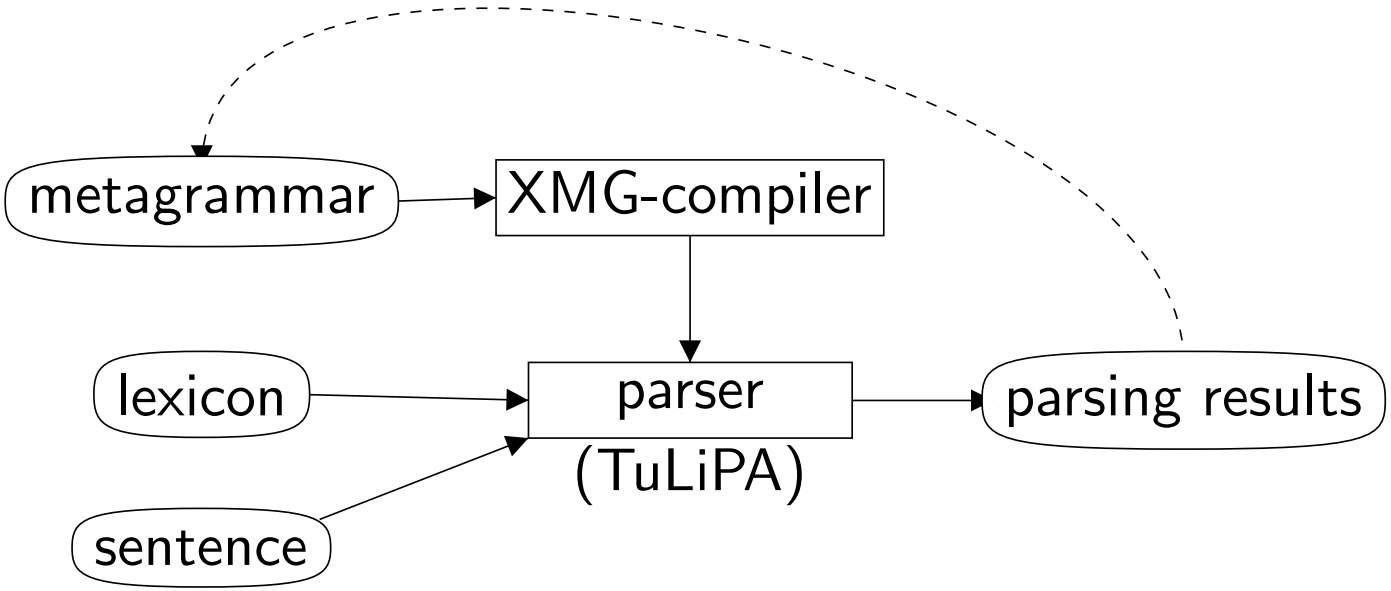
TT-MCTAG example

(3) dass den Kühlschrank heute Peter repariert
("that Peter repairs the fridge today")





The implementation framework:



- XMG: eXtensible MetaGrammar (Duchier et al, 2004)
- TuLiPA: Tübingen Linguistic Parsing Architecture (Parmentier et al, 2008)

eXtensible MetaGrammar (XMG)

(Duchier et al, 2004)

XMG lets one **construct a grammar semi-automatically** by describing tree fragments and their combination. The output structures are **unlexicalized trees** (tree schemata).

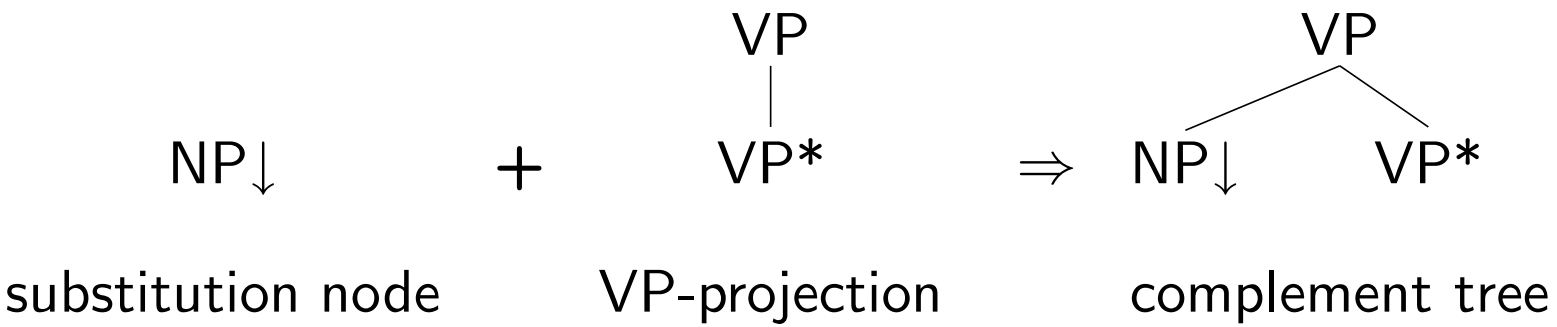
Essential for: consistency, design and maintainance efforts

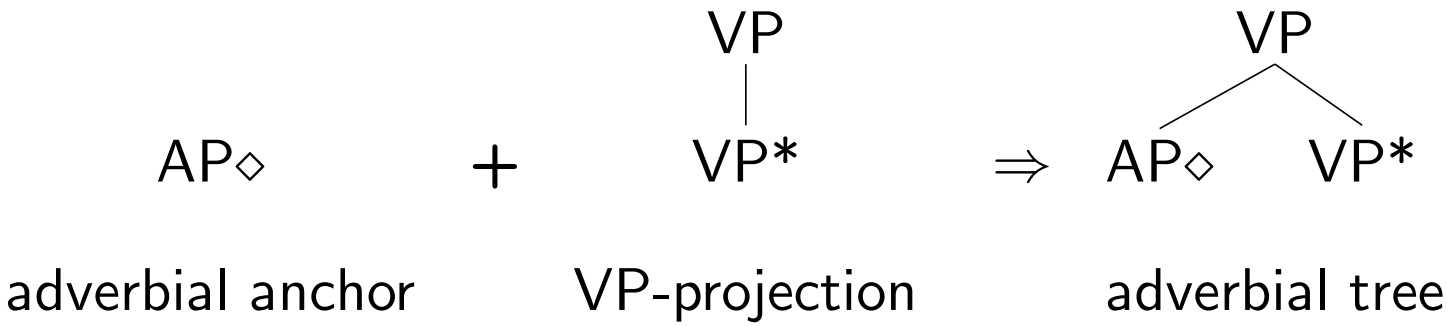
Components:

1. a descripton language
2. a compiler
3. a viewer
4. output format: XML

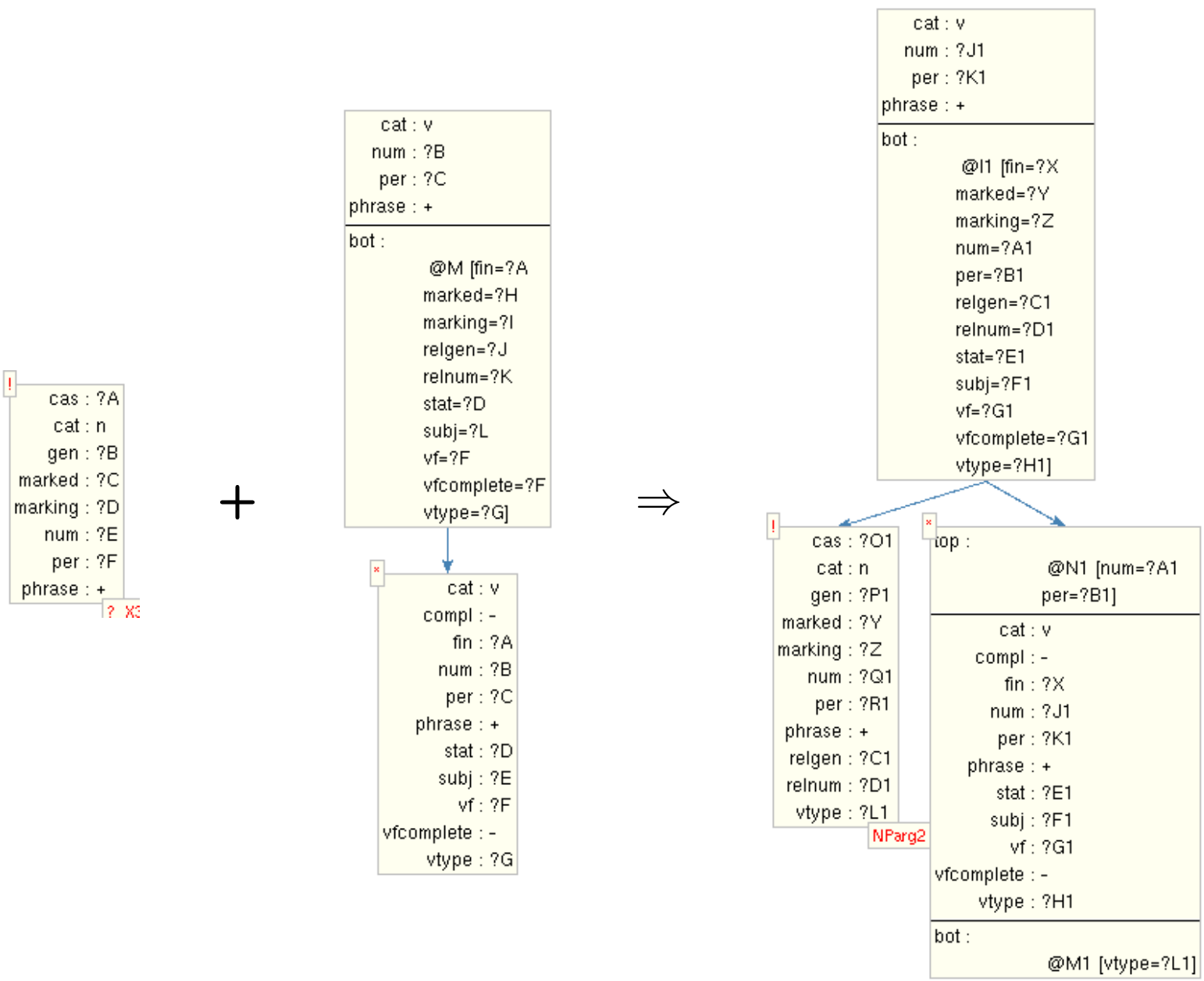
$\Rightarrow$ XMG has been extended to describe tree sets.

XMG: An example





XMG: An example



A 2-layered lexicon

Morphological lexicon

maps an (inflected) token to some lemma form, while preserving morphological information in a feature structure.

vergisst	vergessen	[pos=v; num=sg; per=3;]

Lemma lexicon

maps a lemma onto tree tuple families, while also containing selectional restrictions (e.g., case assignment).

*ENTRY: vergessen
*CAT: v
*SEM: BinaryRel[pred=vergessen]
*ACC: 1
*FAM: Vnp2
*FILTERS: []
*EX:
*EQUATIONS:
NParg1 → cas = nom
NParg2 → cas = acc
*COANCHORS:

A 2-layered lexicon

Morphological lexicon

maps an (inflected) token to some lemma form, while preserving morphological information in a feature structure.

vergisst	vergessen	[pos=v; num=sg; per=3;]

Lemma lexicon

maps a lemma onto tree tuple families, while also containing selectional restrictions (e.g., case assignment).

*ENTRY: vergessen
*CAT: v
*SEM: BinaryRel[pred=vergessen]
*ACC: 1
*FAM: Vnp2
*FILTERS: []
*EX:
*EQUATIONS:
NParg1 $\rightarrow$ cas = nom
NParg2 $\rightarrow$ cas = acc
*COANCHORS:

Tübingen Linguistic Parsing Architecture (TuLiPA)

(Parmentier et al, 2008)

Components:

1. TT-MCTAG-to-RCG converter (on-line)
2. RCG parser $\longrightarrow$ RCG derivation forest $\longrightarrow$ TT-MCTAG derivation forest
3. Parse viewer (derived tree, derivation tree, dependency view, semantic representation)

Availability of TuLiPA:
written in Java and released under the GNU GPL
(`http://sourcesup.cru.fr/tulipa/`)

TuLiPA: Why RCG?

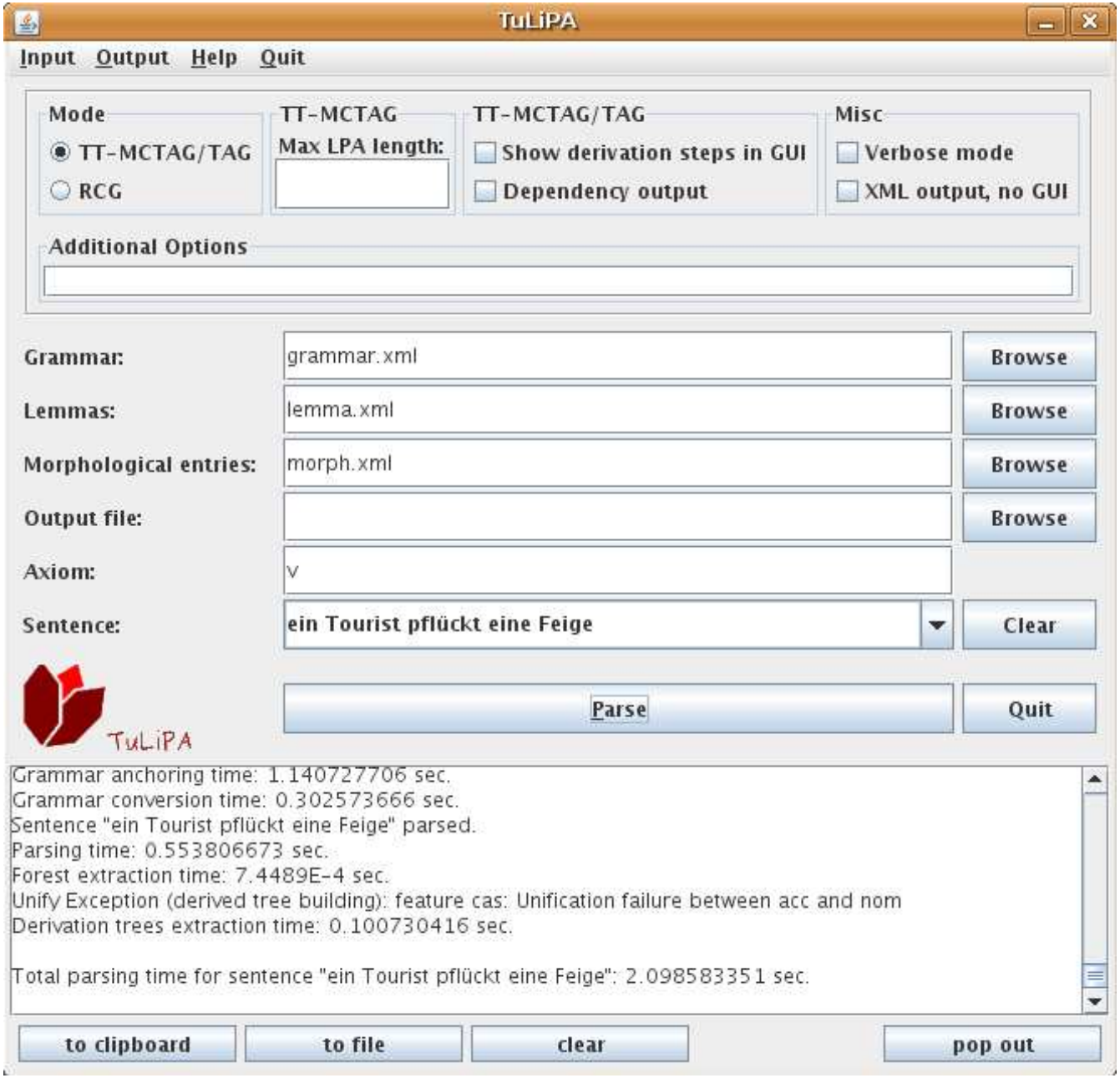
RCG is useful, because:

- it has attractive formal properties (polynomially parsable, full expressive power of MCS-languages);
- there exist parsing algorithms.

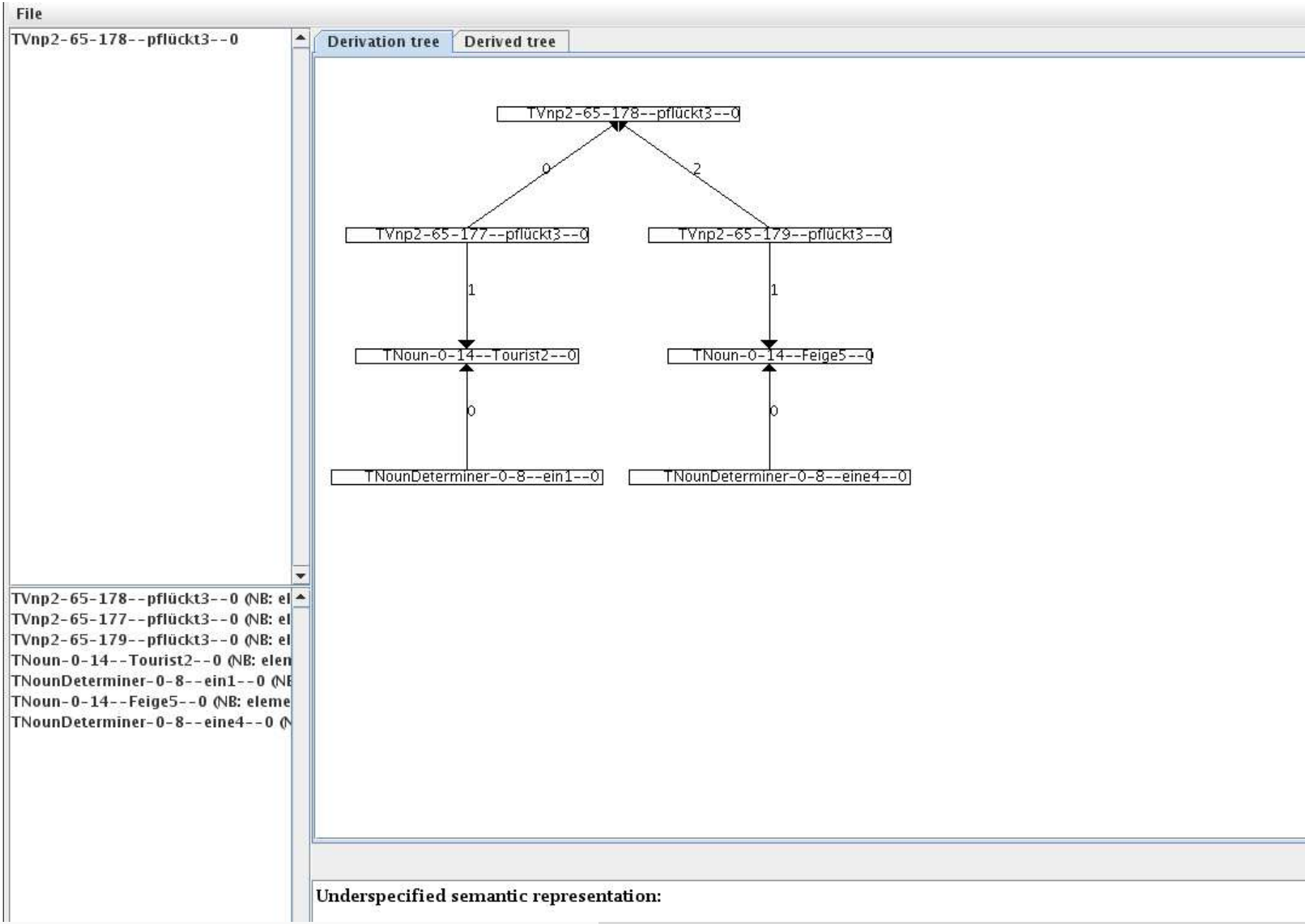
$\Rightarrow$ Parser can be reused for other mildly context-sensitive formalisms!

NB: RCG properly includes MCS. We use a restricted RCG, called *simple RCG*, that is included in MCS.

TuLiPA: The graphical frontend



TuLiPA: The graphical frontend



Ongoing grammar development

GerTT (German TT-MCTAG)

Large-coverage TT-MCTAG for German, including semantics.

Linguistic principals:

- no empty elements such as traces and PRO
- no control and raising in the syntax

State of implementation:

- free word order phenomena:
  scrambling, coherent constructions, verbal clustering
- extraction phenomena:
  relative clauses, wh-questions, bridging constructions
- ca. 70 XMG-classes

Currently, coverage testing is prepared based on the TSNLP test suite.

Summary

TT-MCTAG:

- More natural support of flexible word order languages, but still mildly context-sensitive (in fact only k-TT-MCTAG).

The implementation framework:

- XMG + TuLiPA: Immediate control over implementational (consistency) and linguistic (coverage) aspects of the grammar.
- XMG: Effortless means for making systematic changes in the grammar.
- TuLiPA: Easiliy adoptable to other MCS formalisms (given a RCG conversion algorithm).

And **GerTT** is on his way ...

References

Denys Duchier,Joseph Le Roux,Yannick Parmentier (2004): *The Metagrammar Compiler: An NLP Application with a Multi-paradigm*. Second International Mozart/Oz Conference (MOZ'2004)Architecture.

Yannick Parmentier, Laura Kallmeyer, Wolfgang Maier, Timm Lichte, Johannes Dellert (2008): *TuLiPA: A syntax-semantics parsing environment for mildly context-sensitive formalisms*. Proceedings of the The Ninth International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+9).