

# A Multilayered Declarative Approach to Cope with Morphotactics and Allomorphy in Derivational Morphology

Johannes Handl    Carsten Weber

Friedrich-Alexander-Universität Erlangen-Nürnberg  
Department Germanistik und Komparatistik  
Professur für Computerlinguistik  
Bismarckstr. 6, 91054 Erlangen

May 20, 2010

# Overview

- 1 Introduction
  - JSLIM
  - Left-Associative Grammar
  - Allomorph Method
- 2 Derivational Morphology
- 3 Evaluation
- 4 Further Improvements

# JSLIM - Parser for Natural Language Analysis

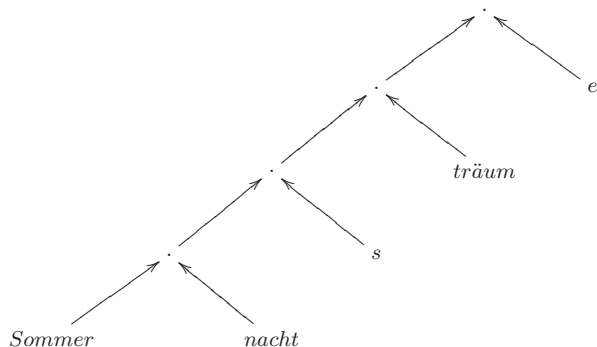
- a software system for writing grammars
- implemented in Java
- grammars for morphology, syntax and semantics (cf. Handl et al. 2009)
- framework of the SLIM theory of language (cf. Hausser 2001)

# JSLIM - Key Features

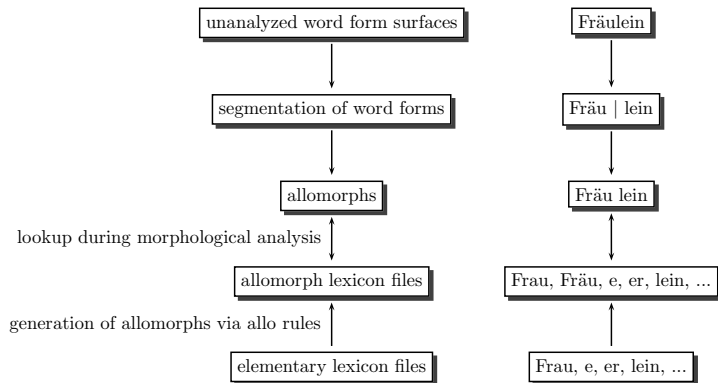
- non-recursive feature structures (proplets)  
nesting can be simulated (cf. Hausser 2006)
- declarative syntax
- rule-based grammar development
- easy upscaling

# Left-Associative Grammar as our Grammar Formalism

- time-linear derivation order from left to right
- principle of possible continuations instead of principle of possible substitutions (PS-Grammar)



# The Allomorph Method



# Allomorphy Phenomena of German

Allomorphic inflectional forms are not sufficient for building derivational forms.

- ⇒ Vowel mutation, example *Fräulein* (miss): Allomorphic inflectional form *Frau*, but allomorphic derivational forms *Frau* and *Fräu*
- ⇒ e-elision, example *Erdling* (earthling): Allomorphic inflectional form *Erde*, but allomorphic derivational form *Erd*
- ⇒ e-elision and vowel mutation, example *Schüler* (scholar) and *Schulung* (schooling): Allomorphic inflectional form *Schule*, but allomorphic derivational forms *Schul* and *Schül*

# Desiderata

- Logical subdivision of the allo rules
  - Paradigmatic rules for handling inflectional allomorphy
  - Distinct rules to generate allomorphic variants for derivation, i.e., rules which are applied independent of a given paradigm
- Logical subdivision of the lexicon files
  - Paradigmatic lexicon files to describe inflection
  - Separate lexicon files to describe derivation only
- No artificial redundancy!



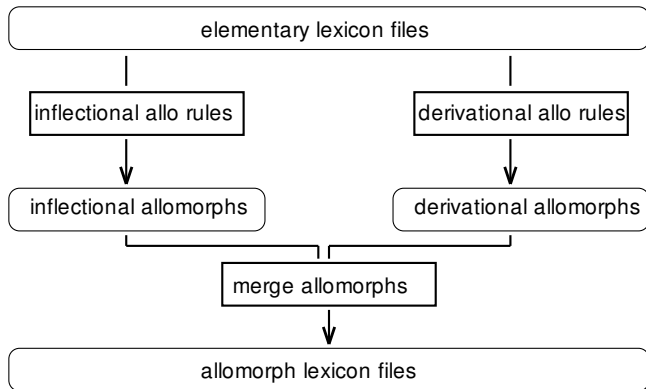
# Motivation

- Easy lexicon compilation
- Transparent lexicon structure which facilitates the task of maintaining, extending and debugging the morphology grammar
- Faster construction period as most of the allo rules can be applied paradigmatically
- Reduced memory consumption as the paradigmatic structure of most of the lexicon entries allows structure sharing

# Treatment of Inflectional and Derivational Allomorphs

- Generation of allomorphs for inflection in a first preprocessor step
- Generation of allomorphs for derivation in a second preprocessor step
- Separate elementary lexicon files for inflectional and derivational allomorphs
- Two different sets of inflectional and derivational allo rules
- Merging of the generated allomorphs with the aim of avoiding redundant entries

# Flowchart of Allomorph Generation



# Generation of Derivational Allomorphs

1) Entries of the derivational elementary lexicon

```
!template[allo: A_chen]
```

```
![sur]
```

```
Balkon
```

```
Blume
```

```
...
```

2) Applying derivational allo rules

```
table A_chen: [sur]          => [sur, noun, der]
```

```
/(.*)((aou))([\^aeiou])*e?/ => $1"$2$3/ /$0/ (chen) .
```

3) Generated derivational allomorphs

```
!template[allo: A_chen
```

```
      der:  chen]
```

```
![sur, noun]
```

```
Balkön Balkon
```

```
Blüm  Blume
```

```
...
```

# Required Entries in the Allomorph Lexicon

	nouns	verbs	adj.	all
inflectional forms	28545	10565	6777	45887
derivative forms	10393	907	1194	12494
total	38938	11472	7971	58381
merged	28387	10557	6771	45715
reduction rate	27.1%	8.0%	15.1%	21.7%

- Most of the inflectional and derivational allomorphs are equal
- The merging reduces 21,7% of the generated allomorphs

# Further Improvements

- Generation of allomorphs based on composition
- Extension of the rule-based system, e.g. treatment of hyphens
- Addition of derivational suffixes which are borrowed from foreign languages

# References

-  Handl, Johannes; Kabashi, Besim; Proisl, Thomas; Weber, Carsten (2009). *JSLIM - Computational morphology in the framework of the SLIM theory of language*. In Mahlow, Cerstin; Piotrowski, Michael (eds.): State of the Art in Computational Morphology: Workshop on Systems and Frameworks for Computational Morphology, SFCM 2009. Zürich: Springer.
-  Hausser, Roland (2001). *Foundations of Computational Linguistics. Human-Computer Communication in Natural Language*. 2nd edition. Berlin, New York: Springer.
-  Hausser, Roland (2006). *A Computational Model of Natural Language Communication*. Berlin, Heidelberg: Springer.
-  Trost, Harald (1990). *The application of two-level morphology to non-concatenative german morphology*. Research Report RR-90-15. DFKI. Saarbrücken

# Contact

Johannes Handl

jshandl@linguistik.uni-erlangen.de

+49 9131 85 25905

Carsten Weber

cnweber@linguistik.uni-erlangen.de

+49 9131 85 29250

Visit also our JSLIM online demo at  
**<http://www.linguistik.uni-erlangen.de/clue/en/research/jslim/online-demo.html>**