

Derivation in JSLIM und Dokumentation der JSLIM-Syntax

Stefan Reihl

Praktikumsbericht

Übersicht

- 1 Dokumentation der JSLIM-Syntax
- 2 Derivation in JSLIM

Ziel

Ziel war es, das in der Dokumentation der JSLIM-Morphologie verwendete Projekt „german1“ bezüglich der Behandlung der deutschen Syntax zu erweitern und ausführlich zu dokumentieren.

Grundlagen

- Aufbau analog zur Morphologie-Dokumentation
- Erweiterung des Projektes „german1“
- Basis: das von Herrn Carsten Weber im Kurs „Einführung in die Grammatikentwicklung“ erstellte Projekt

Resultat

- Erläuterung sämtlicher verwendeter Dateien
- Erläuterung der nötigen Regeln
- Erweiterung auf freie Wortstellung
- Erläuterung der möglichen Operationen und Optionen
- Anpassung und Erweiterung der Morphologie-Dokumentation
- Jede Dokumentation ist jeweils als HTML und als PDF verfügbar

Ziel

Ziel dieses Projekts ist die Behandlung der Allomorphie bei der Derivation: bei der Wortbildung durch Derivation erfolgt beispielsweise bei der Suffigierung oftmals eine Umlautung oder Vokal-/Konsonantentilgung.¹ Diese soll, sofern die „Derivationsallomorphe“ nicht den bereits generierten Pluralallomorphen entsprechen (wie etwa bei Buch - **Bücher** → **Büchlein**), durch den Allomorphiegenerator erfolgen.

¹Bei der Präfigierung ist das nicht der Fall.

Arten der Derivation

- Präfigierung
 - keine Allomorphie
 - Wortklasse, Flexionsklasse und Genus werden vom Basiswort übernommen

Arten der Derivation

- Präfigierung
 - keine Allomorphie
 - Wortklasse, Flexionsklasse und Genus werden vom Basiswort übernommen
- Konversion
 - kein Wortbildungselement
 - Wortklasse, Flexionsklasse und Genus wird durch die Art der Konversion bestimmt

Arten der Derivation

- Präfigierung
 - keine Allomorphie
 - Wortklasse, Flexionsklasse und Genus werden vom Basiswort übernommen
- Konversion
 - kein Wortbildungselement
 - Wortklasse, Flexionsklasse und Genus wird durch die Art der Konversion bestimmt
- Suffigierung
 - Wortklasse, Flexionsklasse und Genus werden durch das Suffix bestimmt (aber auch hier gibt es Ausnahmen wie bei **-el**: die Wurzel, der Kreisel, das Bündel)
 - mögliche Phänomene: Umlautung, e-Tigung, en-Tilgung, Pluralform als Basis, Fugenelemente, mehrteilige Ableitungen, ...

Wortklassen der abgeleiteten Wörter

- Nomenableitung
- Verbableitung
- Adjektivableitung
- Adverbableitung
- (Pronomenableitung)

Wortlisten

Für die Nomen-, Verb-, Adjektiv- und Adverbableitung gibt es Wortlisten, die aus Canoo extrahiert wurden. Diese dienen als Testlisten und geben auch einen guten Anhaltspunkt, wo es noch Handlungsbedarf gibt. Die Listen sind unterteilt in Wortklasse (Nomen-, Verb-, Adjektiv-, Adverbableitung) - Basiswort (Nomen-Nomen-, Verb-Nomen-, Adjektiv-Nomenableitung) - Suffix unterteilt.

Für jedes Suffix gibt es in den entsprechenden Ordnern eine .words-Datei.

Kombiregeln

STEM+DER aus combi.rul:

STEM+DER {STEM+FLX}

```
[cat: (_), der: (A), flx: _]
```

```
[cat: (_ A der), flx: _]
```

```
ecopy(nw.cat.1 SS.cat)
```

```
ecopy(nw.sem SS.sem)
```

```
set(nw.flx SS.flx)
```

```
sappend(SS.sur "|" nw.sur SS.sur)
```

Lexikon

Aufteilung in:

- Suffixlexikon (suffix.lex): enthält sämtliche Suffixe
- Grundformlexikon (USER.lex): dient als ‚Dustbin‘-Kategorie und enthält die für die Suffigierung nötigen Grundformen zu Testzwecken

Lexikon

Beispielintrag aus suffix.lex:

```
!template[ cat: (m-g ling der)
           sem: (m sg)
           flx: C_Haeftling
           ]
!+[sur: ling]
```

Lexikon

Beispieleintrag aus snreihl.lex (A_ling_nn):

```
!template[allo: A_ling_nn
           cat:  (x der)
           sem:  (x x)
           flx:  C_Haeftling
           ]
```

!+sur:

Arm

Bein

Daumen

Dichter

Ehrgeiz

Erde

...

Lexikon

Beispielintrag aus snreihl.lex (A_ling_an):

```
!template[allo: A_ling_an
           cat:  (x der)
           sem:  (x x)
           flx:  C_Haeftling
           ]
```

```
!+sur:
acht
bitter
blau
blöd
breit
...
```

Lexikon

Beispielintrag aus snreihl.lex (A_ling_vn):

```
!template[allo: A_ling_vn
           cat:  (x der)
           sem:  (x x)
           flx:  C_Haeftling
           ]
```

!+sur:

abhängen

abkommen

ankommen

anlehren

anlernen

blenden

...

Allotabellen

Auszug aus snreihl.tbl (A_ling_nn):

table A_ling_nn: [sur]	=> [sur,der]
/^((?:.*[[^] aou])?)([AOUaou])(.*?)(e en)?\$/	=> /\$1\$2"\$3/ (ling).
/^(.*)(e en l ig)\$/	=> /\$1/ (ling).
/(.*)/	=> /\$1/ (ling).

Allotabellen

Auszug aus snreihl.tbl (A_ling_an):

table A_ling_an: [sur]	=> [sur,der]
/^((?:.*[[^] aou]))?([AOUaou])(.*?)\$/	=> /\$1\$2"\$3/ (ling).
/^(.*)(l ig e)\$/	=> /\$1/ (ling).
/(.*)/	=> /\$1/ (ling).

Allotabellen

Auszug aus snreihl.tbl (A_ling_vn):

table A_ling_vn: [sur]	=> [sur,der]
/^((?:.*[[^] aou])?)([AOUaou])(.*?)(en)?\$/	=> /\$1\$2"\$3/ (ling).
/^((?:.*[[^] aou])?)([AOUaou])(.*?)([[^] e])n\$/	=> /\$1\$2"\$3\$4/ (ling).
/^(.*) (en)\$/	=> /\$1/ (ling).
/^(.*) ([[^] e])n\$/	=> /\$1\$2/ (ling).

jetzige Vorgehensweise

- „Taskforce JSLIM“ wurde gegründet
- Nutzung von Subversion als Versionsmanagementsystem
- Aufteilung der Suffixe
- jeder Teilnehmer arbeitet an separaten Dateien (.tbl, .lex)

Probleme

- Übergenerierung

Probleme

- Übergenerierung
- Wortlisten von Canoo sind nicht perfekt (fragliche Einträge, Komposita vorhanden, Schweizerdeutsch)

Probleme

- Übergenerierung
- Wortlisten von Canoo sind nicht perfekt (fragliche Einträge, Komposita vorhanden, Schweizerdeutsch)
- Wie werden die kombinierten Derivationsarten behandelt?

Probleme

- Übergenerierung
- Wortlisten von Canoo sind nicht perfekt (fragliche Einträge, Komposita vorhanden, Schweizerdeutsch)
- Wie werden die kombinierten Derivationsarten behandelt?
- Sollen auch Fremdsuffixe behandelt werden? (siehe Canoo)

Probleme

- Übergenerierung
- Wortlisten von Canoo sind nicht perfekt (fragliche Einträge, Komposita vorhanden, Schweizerdeutsch)
- Wie werden die kombinierten Derivationsarten behandelt?
- Sollen auch Fremdsuffixe behandelt werden? (siehe Canoo)
- Sollen Diminutivsuffixe als eigener Typ zwischen Derivations- und Flexionssuffixen behandelt werden?

Probleme

- Übergenerierung
- Wortlisten von Canoo sind nicht perfekt (fragliche Einträge, Komposita vorhanden, Schweizerdeutsch)
- Wie werden die kombinierten Derivationsarten behandelt?
- Sollen auch Fremdsuffixe behandelt werden? (siehe Canoo)
- Sollen Diminutivsuffixe als eigener Typ zwischen Derivations- und Flexionssuffixen behandelt werden?
- Soll Konversion separat behandelt werden?

Probleme

- Übergenerierung
- Wortlisten von Canoo sind nicht perfekt (fragliche Einträge, Komposita vorhanden, Schweizerdeutsch)
- Wie werden die kombinierten Derivationsarten behandelt?
- Sollen auch Fremdsuffixe behandelt werden? (siehe Canoo)
- Sollen Diminutivsuffixe als eigener Typ zwischen Derivations- und Flexionssuffixen behandelt werden?
- Soll Konversion separat behandelt werden?
- Allomorphiegenerator muss angepasst werden

weiterführende Ideen

- evtl Zusammenfassen mehrere Suffixe zu einer Regel (-lich: Adnomen- und Adverbableitung)
- Übersicht bzgl. Vorkommen der Phänomene (Umlautung, Tilgung, etc) prozentual erfassen
- Abgleich der Wortlisten mit Korpusmaterial