Morphologie

Übersicht

- Grundbegriffe
- Vollformen-Lexikon
- Morphologische Analyse mit DCGs
  - Trennen in Stamm und Endung
  - Flexionsklassen
  - Schnittstelle zwischen Syntax und Semantik

Ziel

- Verstehen des Problems
- In der Lage sein, eine DCG mit einfacher Morphologiebehandlung zu schreiben

Grundbegriffe

Womit befasst sich die Morphologie?

- »Aufbau der Wörter«

  bessere Definitionen → Lexika und Lehrbücher der Linguistik

Derivation vs. Komposition

- Derivation
  - *Frucht → Früchte, fruchten, fruchtbar, unfruchtbar, Unfruchtbarkeit, …*
  - *Gott → Gottes, Götter, Göttin, Göttinnen, vergöttern, Gottheit, …*
- Komposition
  - *Unfruchtbarkeit + Fugen-s + Gott = Unfruchtbarkeitsgott*
  - *Unfruchtbarkeitsgöttinen + Verehrung = Unfruchtbarkeitsgöttinenverehrung*
  - *Unfruchtbarkeitsgöttinenverehrung + Fugen-s + Zeremonienmeister = Unfruchtbarkeitsgöttinenverehrungszeremonienmeister*

Vollformen-Lexikon

```
n(nom, sg) --> [kind].
n(gen, sg) --> [kindes].
n(dat, sg) --> [kind].
n(dat, sg) --> [kinde].
n(akk, sg) --> [kind].
n(nom, pl) --> [kinder].
n(gen, pl) --> [kinder].
n(dat, pl) --> [kindern].
n(akk, pl) --> [kinder].
```

Flexionsformen von »Kind«

```
n(nom, sg) --> [lied].
n(gen, sg) --> [liedes].
n(dat, sg) --> [lied].
n(dat, sg) --> [liede].
n(akk, sg) --> [lied].
n(nom, pl) --> [lieder].
n(gen, pl) --> [lieder].
n(dat, pl) --> [liedern].
n(akk, pl) --> [lieder].
```

Flexionsformen von »Lied«

Vollformen-Lexikon in einer Definit-Klausel-Grammatik

- Regeln für »Präterminalsymbole«, Wörter zwischen [und]
- der Einfachheit halber befassen wir uns hier nicht mit Gross- und Kleinschreibung

Vollformen-Lexikon: Probleme

Flexion mit Vollformen-Lexikon ist aufwendig

- für Englisch (da flexionsarme Sprache): noch halbwegs praktikabel
- für Deutsch:
  - Substantive: Kasus × Numerus = 8 Formen
  - Adjektive, Verben: mehr Formen
- Finnische Verben: ~12'000 Formen

Vollformen-Lexikon: Probleme

Gibt es überhaupt endlich viele Wörter?

- Komposition kann beliebig (?) komplex werden:
  der Unfruchtbarkeitsgöttinenverehrungszeremonienmeister-ausbildungskommissionspräsidentensesselbezug
- *der erste, …, viermillioneneinhunderttausendundzweite, viermillioneneinhunderttausendunddritte, … Schluck*
- *das Während-der-Vorstellung-in-der-Nase-Bohren ist untersagt*

Lösungsansätze

Grammatiken für Wörter

- Auflisten von Wort-Bestandteilen (Stammformen, Endungen)
- *Regeln* zur Verbindung dieser Bestandteile
- Festhalten dieser Regeln als Kontextfreie Grammatiken, DCGs, …
  - die Formalismen vieler Syntaxtheorien (z.B. HPSG) werden auch zur Modellierung morphologischer Phänomene verwendet
- Nachfolgend vorgestellt: Miniatur-Beispiel mit DCGs

Weitere Ansätze

- Endliche Automaten, Zwei-Ebenen-Morphologie, …
- Empfehlenswert: Martin Volks Kurs »Lexikonaufbau und Morphologieanalyseverfahren«, Di 16 – 18 Uhr

Morphologie mit DCGs

Grundidee

- Satzgrammatik = Wörter + Kompositionsregeln
- Wortgrammatik = Buchstaben + Kompositionsregeln
  - ebenfalls denkbar wäre eine Wortgrammatik, welche die Verknüpfung von *Morphemen* beschreibt
  - dann müsste ein vorgelagerter Prozess die Buchstabenfolge in Morpheme zerlegen

ASCII-Codes in Prolog

Eine beliebige Zeichenkette, zwischen zwei " eingeschlossen, wird als Liste der ASCII-Codes dargestellt.

```
?- Kette = "Hallo Du".
Kette = [72,97,108,108,111,32,68,117].
```

32	Leerschlag
68	D
72	H
97	a
108	l
111	o
117	u

Vgl. Lektion »Ein-/Ausgabe« im Wintersemester

Trennen von Stamm und Endung

```
n_morph(Kas, Num) -->
  n_stamm,
  n_endung(Kas, Num).
```

Verknüpfungsregel

```
n_stamm --> "kind".
n_stamm --> "lied".
n_stamm --> "bild".
```

Nominalstämme

```
n_endung(nom, sg) --> "".
n_endung(gen, sg) --> "es".
n_endung(dat, sg) --> "".
n_endung(dat, sg) --> "e".
n_endung(akk, sg) --> "".
n_endung(nom, pl) --> "er".
n_endung(gen, pl) --> "er".
n_endung(dat, pl) --> "ern".
n_endung(akk, pl) --> "er".
```

Nominalendungen

Flexionsklassen

Unterschiedliche Wörter haben unterschiedliche Flexionsklassen.

```
n(nom, sg) --> [kind].
n(gen, sg) --> [kindes].
n(dat, sg) --> [kind].
n(dat, sg) --> [kinde].
n(akk, sg) --> [kind].
n(nom, pl) --> [kinder].
n(gen, pl) --> [kinder].
n(dat, pl) --> [kindern].
n(akk, pl) --> [kinder].
```

Flexionsformen von »Kind«

```
n(nom, sg) --> [brot].
n(gen, sg) --> [brotes].
n(dat, sg) --> [brot].
n(dat, sg) --> [brote].
n(akk, sg) --> [brot].
n(nom, pl) --> [brote].
n(gen, pl) --> [brote].
n(dat, pl) --> [broten].
n(akk, pl) --> [brote].
```

Flexionsformen von »Brot«

Flexionsklassen

```
n_morph(Kas, Num) -->
  n_stamm(Klasse),
  n_endung(Klasse, Kas, Num).
```

Verknüpfungsregel

```
n_stamm(1) --> "kind".
n_stamm(1) --> "lied".
n_stamm(1) --> "bild".
n_stamm(2) --> "brot".
```

Nominalstämme

```
n_endung(1, nom, sg) --> "".
n_endung(1, gen, sg) --> "es".
n_endung(1, dat, sg) --> "".
n_endung(1, dat, sg) --> "e".
n_endung(1, akk, sg) --> "".
n_endung(1, nom, pl) --> "er".
n_endung(1, gen, pl) --> "er".
n_endung(1, dat, pl) --> "ern".
n_endung(1, akk, pl) --> "er".
```

Nominalendungen Klasse Nr. 1

```
n_endung(2, nom, sg) --> "".
n_endung(2, gen, sg) --> "es".
n_endung(2, dat, sg) --> "".
n_endung(2, dat, sg) --> "e".
n_endung(2, akk, sg) --> "".
n_endung(2, nom, pl) --> "e".
n_endung(2, gen, pl) --> "e".
n_endung(2, dat, pl) --> "en".
n_endung(2, akk, pl) --> "e".
```

Nominalendungen Klasse Nr. 2

Schnittstelle Syntax/Morphologie

Schnittstelle zwischen Syntax und Morphologie

- Syntax: arbeitet mit Atomen
- Morphologie: arbeitet mit Zeichenlisten
- `brotes` ≠ `"brotes"`

Umwandlung: name/2

- das in Prolog eingebaute Prädikat name/2 verwandelt Atome in Listen aus ASCII-Codes (und zurück)
  - ```
    ?- name(brotes, Buchstaben).
    Buchstaben = [98,114,111,116,101,115]
    ```
  - ```
    ?- name(Wort, [98,114,111,116,101,115]).
    Wort = brotes
    ```

Schnittstelle Syntax/Morphologie

```
np(Kas, Num) -->
   det(Kas, Num),
   n(Kas, Num).

det(nom, sg) --> [das].
det(nom, pl) --> [die].

n(Kas, Num) :-
   [Wort],
   { name(Wort, Buchstaben),
     phrase(n_morph(Kas, Num), Buchstaben) }.
```

Vgl. Lektion »DCGs« im Wintersemester

Anfragen

- ```
  ?- phrase(np(Kasus, Numerus), [die, brote]).
  Kasus = nom, Numerus = pl
  ```
- ```
  ?- phrase(np(Kasus, Numerus), [die, broter]).
  no
  ```

Probleme, Literatur

Nachteile des vorgestellten Verfahrens

- recht ineffizient
- nicht direkt zum Generieren verwendbar
  - Anfrage `?- phrase(np(nom, pl), X).` führt zu Fehlermeldung

Literatur

- Morphologie in Prolog: [Covington, 1994]
- Morphologie allgemein:
  - A. Linke/M. Nussbaumer/P. R. Portmann: Studienbuch Linguistik. Tübingen: Niemeyer, 1991. ISBN 3-484-31121-5. Seiten 55 – 72.

Aufgaben: Morphologie

Programmiertechniken der Computerlinguistik 2 · Sommersemester 1999

1. Repetition

Stelle sicher, dass Du verstehst, wie Prolog DCGs verarbeitet. Welchen Zweck besitzen die geschweiften Klammern in den Folien zur Vorlesung? Schlage nötigenfalls in den Unterlagen des Wintersemesters nach.

2. Komposition

Definit-Klausel-Grammatiken können nicht zur in der Flexionsmorphologie eingesetzt werden, sondern auch in der Kompositionsmorphologie. Schreibe eine Grammatik für Nominalkompositaformen des Deutschen.

So sollen folgende Wörter erkannt werden, ohne dass die Komposita im Lexikon aufgeführt wären:

– wuerstchenbude
– arbeitspause
– pausenbrote
– arbeitszeit
– zeitarbeit

3. Addition

Füge eine Morphologiebehandlung zur Grammatik für deutsche Sätze aus dem letzten Aufgabenblatt hinzu.