

## Einführung in die Grammatikentwicklung: Übung 5 (22. Juni 2020)

### Ziele:

1. Verbesserung der Erfassung von Generalisierungen im Lexikon.
2. Einführung von Lexikonregeln in die Grammatik.

### Übungen:

1. TRALE starten und Grammatik 5 laden: Klicken Sie auf das grüne T, unter dem Task 5 steht.
2. Durch die Benutzung von Makros kann man Redundanz (Wiederholung gleicher Beschreibungen an verschiedenen Stellen) vermeiden und Grammatiken übersichtlich und wartbar machen. In der letzten Übung haben Sie Makros definiert, deren Benutzung zur Vereinfachung des Lexikons im Bereich der Nomina und Determinatoren geführt hat. Heute soll das konsequent zu Ende geführt werden und alle Redundanz aus der Datei lexicon.pl entfernt werden.

- (a) Führen Sie Makros für Verben und Präpositionen ein und fügen Sie Beschränkungen hinzu, die für jeweils alle Instanzen der entsprechenden Wortarten gelten.

```
verb :=  
  (word,  
   ... ).
```

```
prep :=  
  (word,  
   ... ).
```

- (b) Da die Möglichkeit, verschiedene Kasus, Numerus und Genera zu kombinieren, im Deutschen zu einer relativ großen Anzahl von Klassen führt, würde die Anzahl der verwendeten Makros sehr groß werden, wenn man nur Makros der Art, wie sie in 2a beschrieben wurde, verwendete. Die Grammatik würde unübersichtlich und schlechter wartbar. Man kann dieses Problem umgehen, indem man Makros mit Argumenten verwendet.

Das folgende Makro hat Werte für Numerus, Kasus und Genus. Die Werte werden im Lexikoneintrag spezifiziert.

```
noun(Num,Case,Genus) :=  
  (word,  
   head:(noun,  
         agr:(Agr,  
              num:Num,  
              per:third,  
              case:Case,  
              gen:Genus)),  
   subcat:[(head:(det,  
                agr:Agr)) ] ).
```

Ein Lexikoneintrag kann dann wie folgt aussehen:

```
hundes ---> @noun(sg,gen,mas) .
```

Wenn Sie für einen Wert keine Angabe machen wollen, können Sie ‘\_’ schreiben:

```
katze ---> @noun(sg,_,fem) .
```

- (c) Definieren Sie ein Makro für Verben, die für Subjekte mit bestimmten Numerus- und Person-Werten subkategorisiert sind. Dieses Makro kann unabhängig von der Stelligkeit des Verbs definiert werden. Will man nur über den Anfang einer Liste etwas aussagen und weiß nicht, wie viele Elemente eine Liste hat, kann man folgende Syntax verwenden:

[X|\_]

Hiermit wird eine Liste, die das Objekt  $X$  als erstes Element enthält, beschrieben.

- (d) Definieren Sie Makros für mono-, bi- und tri-valente Verben. Führen Sie Untermakros für bivalente Verben ein, die den Dativ bzw. Akkusativ regieren. Definieren Sie Makros für trivalente Verben. Beachten Sie die Möglichkeit der Umstellung von Argumenten, die sich in unserem Lexikon im Vorhandensein von zwei Lexikoneinträgen für *gibt* widerspiegelt.

Will man etwas über das zweite Element einer Liste sagen, ohne das erste zu spezifizieren, kann man das Zeichen ‘\_’ verwenden:

[\_ ,X]

- (e) Verwenden Sie die Makros, die Sie in 2c und 2d definiert haben, um komplexere Makros für Ihre Einträge in `lexicon.pl` zu definieren. Ein komplexes Makro kann man z. B. wie folgt definieren:

```
x :=
  (@y,
   @z,
   a:b).
```

Diese Definition drückt aus, dass das Makro  $x$  die Information der Makros  $y$  und  $z$  enthält und außerdem das Merkmal  $A$  den Wert  $b$  hat.

- (f) Ändern Sie Ihre Einträge in der Datei `lexicon.pl`, so dass sie die neu definierten Makros verwenden. Nach dieser Änderung sollten Lexikoneinträge nur noch aus einem Namen für den Lexikoneintrag, dem Namen für ein lexikalisches Makro und der orthographischen Form bestehen. Alles Andere sollte in `macros.pl` definiert sein. Es folgt ein Beispieleintrag:

```
bellen ---> @mono_val_verb(third,pl).
```

3. Die Makros erlauben uns, gewisse Generalisierungen in der Grammatik zu erfassen, aber wir brauchen einen anderen Mechanismus, um Ähnlichkeiten von morphologisch verwandten Wörtern mit unterschiedlichen syntaktischen Eigenschaften darstellen zu können. Wir führen Lexikonregeln in die Grammatik ein, die neue Wörter von bereits existierenden Wörtern ableiten. Heute wird der Zusammenhang zwischen den beiden Einträgen für *gibt* hergestellt, die zur Verarbeitung der beiden Sätze in (1) ins Lexikon aufgenommen wurden.

- (1) a. Gibt der Hund dem Schaf die Katze?  
b. Gibt der Hund die Katze dem Schaf?

- (a) Führen Sie für *word* das Merkmal `REORDER` ein: Der Wert ist vom Typ *bool*. Fügen Sie den Typ *bool* und seine beiden Untertypen *plus* und *minus* in die Signatur ein. Definieren Sie für *word* die beiden Untertypen *simple\_word* und *complex\_word*. Der Typ *complex\_word* führt ein Merkmal `DTR` ein, dessen Wert vom Typ *word* ist.
- (b) Öffnen Sie die leere Datei `lexrules.pl` und fügen Sie eine neue Lexikonregel mit der folgenden Struktur hinzu:

```
reorder_lr lex_rule
(reorder:minus,
 ...
 Dtr)
**>
(reorder:plus,
 ...
 dtr:Dtr)
morphs
  X becomes X.
```

Die Werte des Merkmals `REORDER` stellt sicher, dass die Lexikonregel nicht auf die Ausgabe der Lexikonregel erneut angewendet werden kann.

- (c) Fügen Sie die nötigen Beschränkungen für Merkmale ein, so dass die Regel als einziges Argument ein dreistelliges Verb nimmt und ein dreistelliges Verb mit anderer Reihenfolge in der Subcat-Liste produziert, d. h. zum Beispiel ein Verb mit der Argumentabfolge Nom, Dat, Acc nimmt und ein Verb mit der Abfolge Nom, Acc, Dat lizenziert.
- (d) Entfernen Sie die handgeschriebenen Lexikoneinträge für ditransitive Verben mit der Abfolge Nom, Acc, Dat aus der Datei lexicon.pl. Die sind jetzt nicht mehr nötig, da sie von der Lexikonregel erzeugt werden.