

## Einführung in die Grammatikentwicklung: Übung 6 (8. Juni 2021)

### Ziele:

1. Einführung von Prinzipien (Typimplikationen)
2. Kennenlernen der Maschinerie zur Analyse der Flexion.

### Übungen:

1. TRALE starten und Grammatik 6 laden: Klicken Sie auf das grüne T, unter dem Task 6 steht.
2. Die neue Grammatik besitzt eine Morphologiekomponente. Im Lexikon stehen nur noch Stämme, wie *bell-*, aus denen die Morphologiekomponente vollständig flektierte Wörter wie *bellt* und *bellst* ableitet. Sehen Sie sich den Eintrag für *bell-* in der Datei *lexicon.pl* an und testen Sie die Sätze.

- (1)
- a. Ich belle.
  - b. Du bellst.
  - c. Er bellt.
  - d. \*Er bell.

Die Folge in (1d) wird nicht analysiert. Das liegt daran, dass in der Grammatik zwischen Stämmen und Wörtern unterschieden wird. Nur vollständig flektierte Elemente (Wörter) können in der Syntax verwendet werden, unflektierte Stämme sind dort nicht zulässig. In der Datei *rules.pl* wurde spezifiziert, dass alle Töchter vom Typ *syntactic\_object* sein müssen. Nur Phrasen oder Wörter sind syntaktische Objekte, weshalb Stämme als Töchter in den Grammatikregeln ausgeschlossen sind.

Wenn Sie die Regeln in *rules.pl* betrachten, werden Sie feststellen, dass viele Information dort mehrfach spezifiziert ist. So ist zum Beispiel die Information, dass der Mutterknoten vom Typ *phrase* ist und dass die Töchter vom Typ *syntactic\_object* sind, in jeder Regel spezifiziert.

```
h_arg rule (phrase,  
            head:Head,  
            subcat:Subcat)  
====>  
cat> (syntactic_object,  
      head:Head,  
      subcat:[NonHeadDtr|Subcat]),  
cat> (syntactic_object,  
      NonHeadDtr).
```

Die Schreibweise für Regeln ist der Geschichte des Grammatikentwicklungssystems geschuldet. Die oben angegebene Regel beschreibt man normalerweise in HPSG mit der folgenden Merkmalsbeschreibung:

$$\left[ \begin{array}{l} \textit{phrase} \\ \text{HEAD} \quad \boxed{1} \\ \text{SUBCAT} \quad \boxed{2} \\ \text{HEAD-DTR} \quad \left[ \begin{array}{l} \textit{syntactic\_object} \\ \text{HEAD} \quad \boxed{1} \\ \text{SUBCAT} \quad \langle \boxed{3} \rangle \oplus \boxed{2} \end{array} \right] \\ \text{NON-HEAD-DTR} \quad \boxed{3} \textit{syntactic\_object} \end{array} \right]$$

Unsere Grammatik unterscheidet jetzt zwischen Stämmen, Wörtern und Phrasen. In syntaktischen Strukturen wollen wir nur Wörter oder Phrasen zu Phrasen kombinieren. Stämme sollen nicht als Töchter vorkommen. Das wird durch die Spezifikation des Typs der Kopftochter und der Nicht-Kopftochter sichergestellt. Die folgende Information ist allen Phrasen mit zwei Töchtern gemeinsam:

$$\left[ \begin{array}{l} \textit{phrase} \\ \text{HEAD-DTR} \quad \textit{syntactic\_object} \\ \text{NON-HEAD-DTR} \quad \textit{syntactic\_object} \end{array} \right]$$

Interessanterweise müssen wir diese Information nicht in den Regeln aufschreiben, denn sie ergibt sich von selbst aus der Signatur. Das Merkmal `HEAD-DTR` ist nur für Strukturen vom Typ *headed-phrase* definiert, weshalb sich aus der Anwesenheit des Merkmals in einer Struktur ergibt, dass diese ein Untertyp von *headed-phrase* sein muss. Der Wert des Merkmals `HEAD-DTR` ist in der Signatur als vom Typ *syntactic\_object* festgelegt. Gleiches gilt für den Wert von `NON-HEAD-DTR`.

Die Spezifikation der Typen in den Regeln kann also entfallen:

```
h_arg rule (head:Head,
            subcat:Subcat)
    ==>
cat> (head:Head,
      subcat:[NonHeadDtr|Subcat]),
cat> NonHeadDtr.
```

Zu dieser Regel gibt es eine parallele Variante, die sich nur dadurch von der ersten unterscheidet, dass die Töchter anders angeordnet sind:

```
arg_h rule (head:Head,
            subcat:Subcat)
    ==>
cat> NonHeadDtr,
cat> (head:Head,
      subcat:[NonHeadDtr|Subcat]).
```

Allen Regeln in `rules.pl` ist gemeinsam, dass die Kopfwerte (`head:Head`) der Kopftochter mit denen der Mutter geteilt werden.

So wie die Grammatik zur Zeit ist, können wir diese Gemeinsamkeit der Regelpaare und die Teilung der Kopfmerkmale schlecht ausdrücken. Wir können allerdings den Makro-Mechanismus verwenden und dann die Grammatikregeln durch Merkmalbeschreibungen beschreiben. Statt der obigen Regel schreiben wir:

```
h_arg rule @h_arg(HeadDtr,NonHeadDtr)
    ==>
cat> HeadDtr,
cat> NonHeadDtr.
```

und definieren `h_arg(HeadDtr,NonHeadDtr)` wie folgt:

```
h_arg(HeadDtr,NonHeadDtr) :=
(head:Head,
 subcat:Subcat,
 head_dtr:(HeadDtr,
            head:Head,
            subcat:[NonHeadDtr|Subcat]),
 non_head_dtr:NonHeadDtr).
```

Das können wir mit allen Grammatikregeln machen. Gemeinsamkeiten lassen sich dann mit den bekannten Mechanismen durch den rekursiven Aufruf von Makros erfassen. Statt der beiden Regeln, die zum großen Teil gleiche Merkmal-Wert-Paare enthalten, haben wir jetzt ein Makro und verwenden es in der einfachen Regel oben und in der folgenden:

```
arg_h rule @h_arg(HeadDtr,NonHeadDtr)
    ==>
cat> NonHeadDtr,
cat> HeadDtr.
```

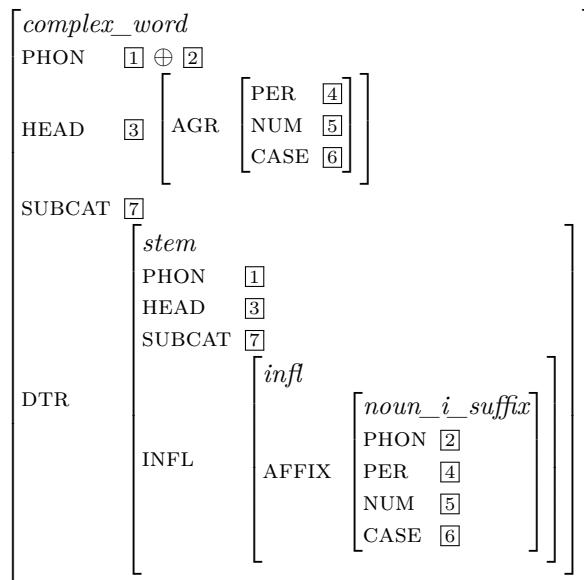
Die beiden Regeln unterscheiden sich nur in der Anordnung der `HeadDtr` und `NonHeadDtr` hinter dem Pfeil. Makros werden zur Abkürzung benutzt. Es gibt noch eine andere Möglichkeit, Gemeinsamkeiten von linguistischen Objekten auszudrücken: Implikationen. Die Regeln in `rules.pl` haben alle eine Kopftochter: Sie sind vom Typ *headed-phrase*. Für diese Phrasen können wir folgendes festlegen:

```
headed_phrase *>
(head:Head,
 head_dtr:head:Head).
```

Diejenigen von Ihnen, die HPSG kennen, werden in der obigen Implikation das Kopfmerkmalsprinzip der HPSG wiedererkennen, das besagt, dass in Strukturen mit einer Kopftochter, die Kopfmerkmale der Kopftochter mit denen der Mutter identisch sein müssen.

Verwenden Sie Makros wie das oben angegebene und die oben angegebenen Typimplikationen, um Generalisierungen zu erfassen und die Regeln in `rules.pl` zu vereinfachen. Nehmen Sie die Makros und Implikationen in die Datei `syntax.pl` auf.

- Überprüfen Sie das Ergebnis, um sicherzustellen, dass die Abdeckung der Grammatik konstant geblieben ist.
- Die Morphologiekomponente verwendet die Techniken, die Sie soeben kennengelernt haben. Eine Lexikonregel für die Flexion von Nomina hat die folgende Form:



Information, die von Flexion nicht betroffen ist, wird von der Tochter zur Mutter hochgereicht (HEAD, SUBCAT).

Der Lexikoneintrag mit seiner flexionsrelevanten Information (INFL) steht unter DTR. Suffixe für die Nominalflexion haben den Typ `noun_i_suffix`. Für diesen ist die folgende Implikation definiert (`suffixes.pl`):

```
noun_i_suffix *>
((phon:[(a_ [])],
  fk:nfk_p,
  num:sg,
  case:nom_dat_acc);

(phon:[(a_ [])],
  fk:nfk_e1,
  dat_e:(minus;both),
  num:sg,
  case:nom_dat_acc);

(phon:[(a_ [])],
  fk:nfk_e1,
  dat_e:plus,
  num:sg,
  case:nom_acc);
```

...

Das ‘;’ steht dabei für Disjunktionen (d. h. eine Verknüpfung mit „oder“). Durch die Impliktion werden sowohl die phonologische Information als auch die morphosyntaktischen Merkmale unter DTR|INFL|AFFIX festgelegt. Die morphosyntaktischen Merkmale werden durch Strukturteilung zu Kongruenzmerkmalen des flektierten Wortes. Der PHON-Wert des flektierten Wortes ergibt sich aus der Verknüpfung des PHON-Wertes des Stammes und des PHON-Wertes des Affixes. Morphophonologische Veränderungen werden in der vorliegenden Grammatik nicht berücksichtigt. Die entsprechenden Effekte werden durch zusätzliche flexionsrelevante Merkmale erreicht. Die Merkmale gehen auf Dorothee Reimann (1992) zurück und sind in (Müller 1999) dokumentiert. (Mit Eingabe von `gv reihmann-morph.ps` in einem xterm im Verzeichnis `GE/Grammatik6` kann man sich die relevanten Teile des Buches ansehen).<sup>1</sup>

Sammeln Sie erste Erfahrungen mit der Morphologiekomponente, indem Sie die Phrase *die kluge Katze* analysieren lassen. Inspizieren Sie die Kanten in der Chart und versuchen Sie deren interne Struktur zu verstehen.

## References

- Müller, Stefan. 1999. *Deutsche Syntax deklarativ: Head-Driven Phrase Structure Grammar für das Deutsche* (Linguistische Arbeiten 394). Tübingen: Max Niemeyer Verlag. <https://hpsg.hu-berlin.de/~stefan/Pub/hpsg.html>.
- Reimann, Dorothee. 1992. *ASL-MEMO-41-92/ZSB*. Tech. rep. Berlin: Forschungsgruppe Computerlinguistik.

---

<sup>1</sup>Terminal öffnen (kleinen Computer unten im Task-Bar anklicken). Dann `cd GE/Grammatik6` eingeben (`cd` = change directory), dann `gv reihmann-morph.ps` eingeben (`gv` = ghost view).