

---

# Vorlesung “Grammatikformalismen”

## 4. Übung (14.06.2019)

Sommersemester 2019 – Prof. Dr. Alexander Koller

---

In dieser Übung entwickeln Sie mit dem XLE-System schrittweise eine etwas größere LFG-Grammatik für das Niederländische. Sie können in jeder Teilaufgabe auf die Grammatik der vorherigen Teilaufgabe aufbauen. Als Ausgangspunkt können Sie die Grammatik `dutch.lfg` aus dem Piazza nehmen.

Ihre Grammatik soll alle angegebenen grammatische Sätze parsen können und alle angegebenen ungrammatischen Sätze ablehnen. Wo Beispiele (unvollständig) mit *c*-Strukturen annotiert sind, soll Ihre Grammatik diese Sätze auch so analysieren. Natürlich soll Ihre Grammatik auch Varianten dieser Sätze richtig analysieren. Überlegen Sie sich dazu weitere eigene Beispielsätze.

Geben Sie eine ZIP-Datei ab, in der für jede Teilaufgabe eine eigene XLE-Grammatikdatei abgelegt ist. Bitte erklären Sie kurz Ihre Lösung, entweder in Kommentaren in den Grammatikdateien oder in einer separaten Datei.

### 1 Zum Aufwärmen

Schreiben Sie eine Grammatik für die folgenden niederländischen Nebensätze:

(1)  $[_S \text{ Jan } [_{VP} \text{ Henk } [_{V'} \text{ zag } ]]]$   
Jan Henk sah

(2)  $[_S \text{ Jan } [_{VP} [_{V'} \text{ zwom } ]]]$   
Jan schwamm

(3) \* Jan Henk zwom  
Jan Henk schwamm

Ihre Grammatik soll Regeln für die Nichtterminale S, NP und VP enthalten sowie Lexikoneinträge für die Eigennamen *Jan*, *Henk*, *Piet* und *Marie* und die Verben *zwom* (intransitiv) und *zag* (transitiv). Beachten Sie, dass XLE-Grammatiken nur eine Regel pro Nichtterminal-Symbol zulassen. Sie müssen für VP also Optiona-  
lität oder Disjunktion verwenden.

## 2 Cross-serial dependencies

Im Niederländischen gibt es wie im Schweizerdeutschen cross-serial dependencies, die man allerdings im String nicht auf den ersten Blick erkennt, weil es keinen Kasus gibt. Erweitern Sie Ihre Grammatik so, dass sie Sätze wie den folgenden parsen kann:

- (4) Jan Henk Piet zag helpen zwemmen  
Jan Henk Piet sah helpen zwimmen

‘(dass) Jan Henk Piet zwimmen helpen sah’

Orientieren Sie sich dabei an der Grammatik auf S. 628 ff. von Bresnan et al. (1982); der Artikel ist auf der Webseite verlinkt. Ersetzen Sie dabei die grammatische Funktion `VCOMP` durch den modernen Namen `XCOMP`. Achten Sie besonders darauf, dass die Valenzen der Verben korrekt gefüllt werden.

## 3 Finitheit und Kongruenz

Erweitern Sie Ihre Grammatik nun so, dass sie (5) und (6) als ungrammatisch ablehnt, aber (7) parsen kann. Beispiel (5) zeigt, dass das Verb-Komplement von *helpen* im Infinitiv stehen muss (*zwom* ist eine finite Vergangenheitsform). Die Verbform *zwemmen* ist entweder Infinitiv oder Plural Präsens. Beispiel (6) ist also ungrammatisch, weil das Matrixverb entweder im Infinitiv steht oder im Numerus nicht mit dem Subjekt kongruiert. Schließlich ist Beispiel (7) grammatisch, aber nur wenn *zwemmen* finit ist.

- (5) \* Jan Henk Piet zag helpen zwom  
Jan Henk Piet sah helpen.INF schwamm.FIN

- (6) \* Jan zwemmen  
Jan zwimmen.INF/.PL

- (7) kinderen zwemmen  
Kinder zwimmen.PL/\*.INF

## 4 Subkategorisierung von PP-Argumenten

Das Verb *brenghen* ‘bringen’ subkategorisiert für ein Objekt (`OBJ`) und eine PP, die eine Richtung angibt (`OBL-DIR`). Erweitern Sie Ihre Grammatik so, dass sie

Sätze wie (8) parsen kann. Dabei ist es wichtig, dass die Präposition *naar* ‘nach’ eine Richtung angibt. Die Präposition *voor* ‘für’ wird als Kopf von beneficiary obliques (OBL-BEN) verwendet; daher soll der Satz (9) als ungrammatisch erkannt werden.

- (8) Jan Henk Piet naar Marie zag brengen  
 Jan Henk Piet nach Marie sah bringen

‘Jan sah Henk Piet zu Marie bringen’

- (9) \* Jan Henk Piet voor Marie zag brengen  
 Jan Henk Piet für Marie sah bringen

Richten Sie Ihre Grammatik so ein, dass für (8) die folgende f-Struktur herauskommt. Denken Sie dabei an die Konstruktion  $(\uparrow (\downarrow \text{PCASE})) = \downarrow$ .

