

# Übung 06: Syntax

Programmiertechniken in der Computerlinguistik II, FS 16

Abgabetermin: 08.06.2016

## Hinweise zur Abgabe

- Bitte gib jedes Python-Programm in einer eigenen Datei ab, die die Dateiendung `.py` hat und *ausführbaren* Python-Code enthält.
- Geize nicht mit Kommentaren direkt im Programm-Code, wo Erläuterungen angebracht sind. Umfangreiche Erklärungen werden hingegen besser in einer separaten README-Datei mitgeliefert (vorzugsweise Plain-Text oder PDF).
- Um das Hochladen der Abgabe auf OLAT zu erleichtern, kannst du die Dateien mit **zip** oder **tar** (oder einem anderen verbreiteten Format) archivieren/komprimieren.

## 1 Manuelles CYK-Parsing

Gegeben ist die folgende kontextfreie Grammatik:

```
S -> NP VP
PP -> P NP
NP -> Det N | Det N PP | 'I'
VP -> V NP | VP PP
Det -> 'an' | 'my'
N -> 'elephant' | 'pajamas'
V -> 'shot'
P -> 'in'
```

Parse den Satz *I shot an elephant in my pajamas* mithilfe des CYK-Algorithmus. Welche Probleme ergeben sich?

Abzugeben sind die Tabelle und die entstandenen Bäume.

## 2 Kontextfreie Grammatiken in Python

Für diese Aufgabe können verschiedene Parser aus NLTK verwendet werden. Wähle einen Parser aus und implementiere die folgenden Funktionalitäten für die Grammatik aus Aufgabe 1:

- a) Parsen eines einzelnen Satzes als String
- b) Parsen von Sätzen und Grammatiken aus Plain-Text-Dateien
- c) Eine schöne Darstellung der Hierarchien auf der Kommandozeile
- d) Ausgabe von kompilierbarem \*TeX-Quellcode aller möglichen Bäume eines Satzes

### 3 Probabilistische kontextfreie Grammatiken in Python

In den bisherigen Aufgaben wurden Ambiguitäten, welche aus der Grammatik entstanden, unverändert weitergegeben. In dieser Aufgabe soll dieses Problem anhand einer probabilistischen kontextfreien Grammatik (PCFG) reduziert werden.

Ergänze die bestehende Grammatik aus Aufgabe 1 mit Wahrscheinlichkeiten und implementiere die Funktionalitäten der vorherigen Aufgabe.

- Welche Schwierigkeiten ergeben sich bei PCFG?
- Welche Wahrscheinlichkeitsverteilung lässt immer noch Ambiguitäten zu?

#### Hinweise zu L<sup>A</sup>T<sub>E</sub>X

- Der von den Skripten ausgegebene \*TeX-Quellcode sollte (idealerweise) kompilierbar sein.
- Für die Darstellung von Syntaxbäumen bietet sich die `qtree`-Syntax an, welche von verschiedenen Seiten her unterstützt wird.
- Ein Beispiel für \*TeX-Quellcode mit einem Syntaxbaum ist mit der Übung mitgeliefert.

#### Reflexion/Feedback

- a) Fasse deine Erkenntnisse und Lernfortschritte in ein paar Sätzen zusammen.
- b) Wie viel Zeit hast du in diese Übungen investiert?