

# Programmierkurs Python II – SS 2010

## Übung 11

---

### 1 Chomsky-Normalform (2 Punkte)

Implementiere eine Funktion `cnf(rules)`, die eine Grammatik als Parameter nimmt und eine äquivalente Grammatik in Chomsky-Normalform zurück gibt. Die Rückgabe-Grammatik soll nur Regeln besitzen, deren rechte Seite entweder aus genau zwei Nichtterminalen oder einem Terminal-Symbol besteht. Der Parameter `rules` soll eine Liste von Regeln sein, eine Regel wiederum ist ein 2-stelliges Tupel, dessen erstes Element die linke Seite (ein String) und dessen zweites Element die rechte Seite (eine Liste von Strings) der Regel darstellen. (Alternative: Ein einfaches Dictionary.) Der Einfachheit halber kannst Du annehmen, dass das Lexikon vom Rest der Grammatik getrennt wurde und in den Regeln der Grammatik selbst keine Terminal-Symbole mehr vorkommen.

### 2 CYK-Erkenner (3 Punkte)

Implementiere einen Erkenner, der den CYK-Algorithmus benutzt. Falls Du Aufgabe 1 implementiert hast, soll der Erkenner unbeschränkte Grammatiken als Eingabe nehmen, falls nicht soll er eine Exception werfen, falls die Grammatik nicht wohlgeformt ist. Die Grammatik soll auf jeden Fall im gleichen Format angegeben werden wie in Aufgabe 1. Implementiere eine Methode `rcgnz_cyk(sentence, rules, lexicon)`, die als Parameter eine Grammatik und ein Lexikon erhält und `True` zurück gibt, falls der Satz `sentence` in der Sprache der Grammatik enthalten gibt (sonst `False`).

### 3 Bonusaufgabe: Earley-Parser (3 Punkte)

Füge zu der Earley-Implementierung aus der Vorlesung eine Methode `parse(sentence)` hinzu, die die Parsebäume eines Satzes zurück gibt. Du musst hierfür die Methode `complete()` und ggf. einige Datenstrukturen anpassen, wie in der Vorlesung besprochen. Der Code von den Folien hierzu ist auf der Homepage bei den Code-Beispielen.

---

Abgabe bis Mittwoch, 08.07.2010, 08:30 Uhr