
Vorlesung “Computerlinguistische Techniken”

2. Übung (23.10.2015)

Wintersemester 2015/16 – Prof. Dr. Alexander Koller

Die Aufgaben 2 und 3 setzen voraus, dass Sie auf Ihrem Computer die NLTK-Bibliothek installiert haben. Eine Anleitung finden Sie im Piazza. NLTK ist korrekt installiert, wenn man in Python `import nltk` ohne Fehlermeldung ausführen kann.

1 Fibonacci-Zahlen

Die Fibonacci-Zahlen f_1, f_2, \dots sind wie folgt definiert:

$$\begin{aligned} f_1 &= f_2 = 1 \\ f_n &= f_{n-1} + f_{n-2} \text{ für alle } n \geq 3. \end{aligned}$$

- Schreiben Sie eine rekursive Python-Funktion `fibr`, so dass `fibr(n)` die Zahl f_n zurückgibt.
- Schreiben Sie eine nichtrekursive Python-Funktion `fibi`, so dass `fibi(n)` die Zahl f_n zurückgibt. Sie brauchen entweder eine `for`-Schleife mit `range` oder eine `while`-Schleife.
- Berechnen Sie mit beiden Funktionen die Zahl f_{40} . Welche Funktion ist schneller?
- Geben Sie die asymptotischen Laufzeiten der beiden Funktionen in Abhängigkeit von n an. Bei exponentiellen Laufzeiten können Sie einfach “exponentiell” sagen; Sie müssen nicht die genaue Basis angeben. Begründen Sie Ihre Antwort.

2 CKY-Erkenner mit NLTK

In der Vorlesung haben wir den Kern einer Python-Implementierung des CKY-Erkenners gesehen. Vervollständigen Sie diese Implementierung, so

dass sie Grammatiken aus einer Datei und Sätze von der Konsole einlesen kann. Ihr Programm soll sich folgendermaßen verhalten:

```
koller$ python cky.py grammatik
Hans isst ein Käsebrod
True
```

Dabei ist `grammatik` der Name einer Datei, in der eine kontextfreie Grammatik im NLTK-Format steht. Der zu parsende Satz soll von der Konsole eingelesen werden, z.B. mit der Funktion `raw_input()`. Schließlich soll Ihr Programm `True` oder `False` ausgeben, je nachdem, ob der Satz in der Sprache ist.

3 CKY-Parser

Erweitern Sie den CKY-Erkennen aus der Vorlesung (oder aus Aufgabe 2) zu einem CKY-Parser, der nicht nur das Wortproblem löst, sondern alle Parsebäume eines Satzes ausgibt.

Sie müssen dazu in der Chart neben den einzelnen Nichtterminalsymbolen auch *Backpointer* speichern, in denen Sie sich merken, wie man einen Chart-Eintrag aus kleineren Chart-Einträgen bauen kann. Um die Parsebäume zu repräsentieren, können Sie die Klassen des Moduls `nltk.tree` verwenden.

Probieren Sie Ihren Parser mit der Grammatik aus Aufgabe 2 der 1. Übung aus.