Parsing mit neuronalen Netzen: Einlesen der Daten

In den folgenden Übungen werden Sie einen Parser auf Basis von neuronalen Netzwerken implementieren. Der Parser orientiert sich an dem folgenden Artikel von Gaddy et al. (2018): `https://arxiv.org/pdf/1804.07853.pdf`

In der ersten Übung geht es darum, **Parsebäume einzulesen**, mit denen der Parser trainiert werden soll. Die Daten finden Sie in `http://www.cis.uni-muenchen.de/~schmid/lehre/data/PennTreebank.zip`.

Beispiel:

Für die Eingabe

```
(TOP(S(NP(NNP Ms.)(NNP Haag))(VP(VBZ plays)(NP(NNP Elianti)))(. .)))
```

soll zurückgegeben werden:

- die **Wortliste**: `['Ms.', 'Haag', 'plays', 'Elianti', '.']`
- die **Konstituentenliste**: `[(TOP=S,0,5), (NP,0,2), (NNP,0,1), (NNP,1,2), (VP,2,4), (VBZ,2,3), (NP=NNP,3,4), (.,4,5)]`

  'TOP=S' und 'NP=NNP' ergeben sich durch Eliminierung von Kettenregeln, die der Parser ja nicht verarbeiten kann.

Am besten schreiben Sie dafür eine **rekursive** Funktion. Beim ersten Aufruf der Funktion werden die linke Klammer und das Label des Wurzelknotens eingelesen. Dann ruft sich die Funktion in einer Schleife selbst rekursiv auf, um die Tochterknoten einzulesen. Wenn die schließende Klammer erreicht ist, wird die Funktion beendet. Jeder Aufruf der Funktion liefert eine Konstituente.

Zusätzlich sollen Sie noch eine Funktion schreiben, welche aus der Wortliste und der Liste der Konstituenten wieder den **originalen Parsebaum** generiert und in Klammernotation ausgibt.

Vorüberlegungen

- Welche Argumente benötigt die rekursive Funktion und was liefert sie zurück?
- Wie kann der Parsebaum rekonstruiert werden?