

Programmierkurs Python II – SS 2013

Übung 6

Für alle Aufgaben kannst Du immer “S” als Startsymbol der Grammatik annehmen.

1 Shift-Reduce-Parser (4 Punkte)

Benutze die vorgegebene Klasse für Bäume (`Tree`, \approx Übung 1) und transformiere den Shift-Reduce-Erkenner aus der Vorlesung zu einem Parser. Die Grundidee ist, Parsebäume statt Nichtterminalsymbole auf dem Stack zu speichern; die Reduce-Operation prüft, gleicht Wurzel-Symbole der Parse-Bäume auf dem Stack mit den rechten Regel-Seiten ab.

- Implementiere eine Methode `parse(sentence, rules)`, die als Generator alle Parsebäume erzeugt und zurückgibt (jeden Parse-Baum genau einmal, Reihenfolge beliebig.)
- Ändere die Methoden `shift`, `reduce` und `match` entsprechend ab.

2 Top-Down-Erkenner (4 Punkte)

Implementiere eine Methode `recognize(sentence, rules, lexicon)`, die einen Top-Down-Erkenner umsetzt (. Hier ist es sinnvoll, zwischen Lexikoneinträgen und Regeln zu unterscheiden, und die vorgestellten Predict- und Scan-Operationen entsprechend zu modifizieren.

3 Top-Down-Parser (2 Punkte, Bonus)

Implementiere einen Top-Down-Parser als Iterator `TopDownParser`, der alle Parsebäume eines Satzes zurück gibt. In `__init__(self, sentence, rules, lexicon)` sollen Satz, Grammatik und Lexikon übergeben werden. `__next__(self)` jede Parsebaum nacheinander zurück und wirft danach `StopIteration`.

Abgabe bis Donnerstag, 13.06.2013, 11:00 Uhr