

1 Shift-Reduce Parser (3 Punkte)

Implementiere eine Klasse für Parsebäume und erweitere den Beispielcode des Shift-Reduce Erkenners zu einem Parser. Idee: Speichere Parsebäume statt Nichtterminalsymbole auf dem Stack. Die Reduce-Operation muss dann prüfen, ob die Wurzel-Symbole der Parse-Bäume auf dem Stack zu der rechten Seite einer Regel passt. Es reicht aus, wenn der Parser einen Parsebaum zurückliefert (vgl. Aufgabe 2).

2 Parser als Iterator (1 Punkte)

Ändere die Implementierung aus der ersten Aufgabe so, dass der Parser als Iterator realisiert wird, der über alle Ableitungsbäume für die Eingabe iteriert.

3 Top-Down Erkenner (3 + 2 Punkte)

Implementiere einen Top-Down Erkenner. Wie in der Vorlesung besprochen, ist es dabei ist es sinnvoll, zwischen Lexikoneinträgen und Regeln zu unterscheiden, und Predict- und Scan-Operationen entsprechend leicht zu modifizieren.

Bonus: Implementiere statt eines Erkenners einen Parser (evtl. knifflig).

Abgabe bis Donnerstag, 2012-06-21, 08:30 Uhr