

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.

2 Parser als Iterator (1 Punkte)

Ändere die Implementierung aus der ersten Aufgabe so, dass der Parser als Iterator realisiert wird. Die Bearbeitung der Aufgabe erübrigt sich, wenn der Parser in Aufgabe 1 bereits als Iterator realisiert wurde.

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, 2011-06-09, 08:30 Uhr