
Vorlesung “Computerlinguistische Techniken”

3. Übung (03.11.2015)

Wintersemester 2015/16 – Prof. Dr. Alexander Koller

1 Earley-Parser ausprobieren

Wenden Sie (von Hand) den Earley-Parser an, um den Satz “Hans beobachtet den Mann mit dem Fernrohr” mit der kontextfreien Grammatik aus Aufgabe 2 der 1. Übung zu parsen. Geben Sie dabei an, in welcher Reihenfolge Sie die einzelnen Parse-Items berechnen und mit welcher Regel jedes einzelne Item begründet ist.

Erklären Sie dann, wie man systematisch die Parsebäume des Satzes aus den berechneten Items rekonstruieren kann. Benutzen Sie Ihren Algorithmus, um die beiden Parsebäume des angegebenen Satzes auszurechnen.

2 Schemabasierte Implementierung von Shift-Reduce

In der Vorlesung haben wir ein Parsingschema für den Shift-Reduce-Erkennen gesehen. Implementieren Sie dieses Parsingschema in Python. Sie können `earley.py`, die schemabasierte Implementierung eines Earley-Erkenners im Piazza, als Grundlage verwenden.

Hinweis: Die Chart ist als Python-Menge (Set) von Items implementiert. Damit man ein Objekt als Element in ein Set eintragen kann, muss dieses Objekt eine Methode namens `__hash__` haben. Die Items Ihres Shift-Reduce-Erkenners enthalten unter anderem den Stack. Beim Versuch, diesen Stack als Liste zu implementieren, werden Sie entdecken, dass die Klasse `list` keine `__hash__`-Methode hat. Das liegt daran, dass man die Elemente einer Liste verändern kann. Mein Rat wäre, den Stack mit *Tupeln* (engl. tuples), also unveränderlichen Listen, zu implementieren. Tupel haben eine `__hash__`-Methode und können deshalb als Elemente von Sets verwendet werden. Viele Listenoperationen kann man auch auf Tupel anwenden, z.B. Konkatenation `+` und Slice-Operationen.

3 Korrektheitsbeweis mit Schemata

In der Vorlesung zum Earley-Parser haben Sie auf der Folie mit der Überschrift „CKY: nichtbinäre Version“ eine Variante des CKY-Erkenner gesehen, die direkt Grammatiken eines beliebigen maximalen Rangs r verarbeiten kann. Insbesondere darf $r > 2$ sein.

Das Parsingschema für diesen Algorithmus verwendet die folgende Regel als einzige Beweisregel:

$$\frac{[B_1, i_1, i_2] \quad \dots \quad [B_k, i_r, i_{r+1}] \quad A \rightarrow B_1 \dots B_k \text{ Produktionsregel}}{[A, i_1, i_{r+1}]}$$

Geben Sie den Rest des Parsingschemas an (Items, Start-Items, Ziel-Items). Erklären Sie kurz, warum das Parsing-Schema tatsächlich den Algorithmus aus der Vorlesung darstellt. Beweisen Sie dann auf der Grundlage des Parsingschemas die Korrektheit und Vollständigkeit des Algorithmus. Bestimmen Sie die asymptotische Laufzeit und begründen Sie anhand des Schemas, wie Sie auf diese Laufzeit kommen.