

Compilerbau - Wintersemester 2021/22

Praktisches Übungsblatt 7

Besprechung der Aufgaben am 10.12.21 um 16:30 Uhr in 25.12.02.55 und gleichzeitig online per BBB
(evtl. früher)

Fragen an Lukas.Lang@hhu.de

Die Bearbeitung ist freiwillig.

Bringen Sie Ihre Lösung mit zum Übungstermin

Aufgabe 6.1

Schreiben Sie einen LR0-Automaten Generator. Ihnen wird eine Vorgabe und Tests gestellt.

1. Schreiben Sie eine Methode `closure`: **`Set<LR0Item> closure(Set<LR0Item> items)`**. Diese berechnet die Hülle einer LR0Item-Menge. (Siehe Kapitel 4c Folie 18)
2. Schreiben Sie eine Methode `goto`: **`Set<LR0Item> gotoState(Set<LR0Item> items, String symbol)`**. Diese berechnet für eine LR0Item-Menge und ein Symbol (Terminal oder Nichtterminal) die Hülle der Nachfolgeitems. D.h. der Punkt wird hinter das Symbol bewegt und `closure` aufgerufen. (Siehe Kapitel 4c Folie 21)
3. Schreiben Sie eine Methode `getStates`: **`Set<LR0Node> getStates()`**. Diese berechnet die Menge aller Zustände des LR0-Automaten. Bestimmen Sie hier alle Symbole hinter einem Punkt eines Knotens und nutzen Sie Ihre Implementierung von `closure` und `goto`.
4. Erzeugen Sie eine LR(0)- und SLR Parsingtabelle. Übergeben Sie im SLR Fall neben dem Automaten auch die Follow-Menge der Grammatik an den Konstruktor.
5. Erzeugen Sie einen LR(0)- und SLR-Parser.

Hinweis zur Lösung:

Die Klasse ViewUtil enthält eine Methode toDot(). Mit dieser lässt sich eine dot Ausgabe erzeugen. Mit dem Aufruf **java ViewUtil -dot > automaton.dot** gefolgt von **dot -Tpng automaton.dot > automaton.png** können Sie so eine png Datei ihres Automaten erzeugen.

Mehr auf: <http://www.graphviz.org/pdf/dotguide.pdf>

Die Klasse ViewUtil enthält eine Methode toTex(). Mit dieser lässt sich eine tex Ausgabe erzeugen. In beiden Fällen muss vorher eine Grammatik in getGrammar() kodiert werden. Im Testcase finden Sie 8 Beispielgrammatiken.