
Compilerbau

<http://proglang.informatik.uni-freiburg.de/teaching/compilerbau/2004/>

Übungsblatt 4

Abgabe: 24.11.2004

Aufgabe 1 (LR(0)-Grammatiken):

Zeige, dass eine kontextfreie Grammatik G genau dann eine LR(0)-Grammatik ist, wenn der charakteristische Automat LR-DFA(G) konfliktfrei ist.

Aufgabe 2 (LR-DFA): Berechne den LR-DFA für die Grammatik

$$G = (\{E, R\}, \{c, \{, \}, \text{sub}, \text{sup}\}, P, E)$$

 mit Produktionen P

$$\begin{aligned} E &\rightarrow c \\ E &\rightarrow E \text{ sub } R \\ E &\rightarrow E \text{ sup } E \\ E &\rightarrow \{ E \} \\ R &\rightarrow \epsilon \\ R &\rightarrow E \text{ sup } E \end{aligned}$$

 Ist G eine LR(0)-Grammatik?

Aufgabe 3 (Elimination indirekter Linksrekursion):

Eine linksrekursive, kontextfreie Grammatik kann in zwei Schritten in eine Grammatik ohne Linksrekursionen transformiert werden: durch Entfernen indirekter Linkrekursion und durch Entfernen direkter Linksrekursion.

Gib einen Algorithmus an, der zu einer kontextfreien Grammatik eine äquivalente Grammatik ohne indirekte Linksrekursion konstruiert.

Aufgabe 4 (Grenzen kontextfreier Sprachen):

Abschnitt 8.1.1 der *Java Language Specification*¹ beschreibt *modifier* von Java-Klassendeklaration wie folgt:

A class declaration may include class modifiers.

ClassModifiers:

ClassModifier

ClassModifiers ClassModifier

ClassModifier: one of

public protected private

abstract static final strictfp

¹<http://java.sun.com/docs/books/jls/>

Not all modifiers are applicable to all kinds of class declarations. The access modifier `public` pertains only to top level classes (§7.6) and to member classes (§8.5, §9.5), and is discussed in §6.6, §8.5 and §9.5. The access modifiers `protected` and `private` pertain only to member classes within a directly enclosing class declaration (§8.5) and are discussed in §8.5.1. The access modifier `static` pertains only to member classes (§8.5, §9.5). A compile-time error occurs if the same modifier appears more than once in a class declaration.

Gib eine alternative Grammatik für `ClassModifier` an, so dass der Compilerfehler, der signalisiert, dass ein *modifier* mehrfach auftritt, schon während der Syntaxanalyse auftritt. Wie viele Nonterminale hat die alternative Grammatik bei n *modifiern*?