P. Thiemann, M. Neubauer Wintersemester 2004/2005

**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

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

mit Produktionen $P$

$$\begin{array}{ll} E & \rightarrow \mathtt{c} \\ E & \rightarrow E\,\mathtt{sub}\,R \\ E & \rightarrow E\,\mathtt{sup}\,E \\ E & \rightarrow \{\,E\,\} \\ R & \rightarrow \epsilon \\ R & \rightarrow E\,\mathtt{sup}\,E \end{array}$$

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 kontextfreie 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*?