Software-Modellierung und Verifikation
Informatik 2
Prof. J.-P. Katoen
RWTH Aachen

Priv.-Doz. T. Noll noll@cs.rwth-aachen.de
Ch. Jansen christina.jansen@cs.rwth-aachen.de

5. Exercise sheet *Compiler Construction 2010*

Due to Wed., 24 November 2010, *before* the exercise course begins.
Hand in your solutions in groups of three or four!

Exercise 5.1: **(3 points)**

Show that every regular language can be generated by a $LL(1)$-grammar.

Exercise 5.2: **(1+1+1 points)**

- Eliminate left-recursion in the following grammar:

$$\begin{array}{lcl} S & \to & a \mid (T) \\ T & \to & T, S \mid S \end{array}$$

- Apply left factorisation to the following grammar:

$$\begin{array}{lcl} S & \to & a \mid (T') \\ T' & \to & S, T' \mid S \end{array}$$

- Are the resulting grammars $LL(1)$? Prove your answer?

Exercise 5.3: **(2 points)**

Consider again the ambiguous grammar G from Ex. 4.3

$$S \quad \to \quad (S) \mid S \vee S \mid S \wedge S \mid \neg S \mid \text{true}$$

as well as the priorities

$\neg$ binds stronger than $\vee$ and $\wedge$,

$\wedge$ binds stronger than $\vee$.

While in Ex. 4.3 you should have given a method to choose a suitable analysis from the set of resulting ones, now modify the grammar in a way, that the binding priorities are implicitely given. Of course the modified grammar should induce the same language as G (cf. Ex. 9.1 from lecture).

Exercise 5.4: **(1+3+1 points)**

Consider the grammar G given by:

$$\begin{array}{lcl} S' & \to & S \\ S & \to & MR \\ M & \to & LM \mid m \\ L & \to & LL' \mid l \\ L' & \to & LL' \mid l' \\ R & \to & RR' \mid r \\ R' & \to & RR' \mid r' \end{array}$$

a) Show that $G \notin LR(0)$.

b) Give a grammar $G' \in LR(0)$ (and show that $G' \in LR(0)$) such that for $w \in \Sigma^*$, $a \in \Sigma$:

$$wa \in \mathcal{L}(G) \Leftrightarrow wm \in \mathcal{L}(G')$$

c) Compute an accepting run of $NBA(G')$ for input $lll'mrm$