

## 8. Exercise sheet *Compiler Construction 2010*

Due to Wed., 15th December 2010, *before* the exercise course begins.

Hand in your solutions in groups of three or four!

**Exercise 8.1:** (3 points)

Show that  $LL(1) \not\subseteq LALR(1)$  by providing a counterexample.

**Exercise 8.2:** (4\* points, optional)

Prove that for every regular language  $L \in REG \setminus \mathcal{L}(LR(0))$  it holds, that  $L$  is not prefix-free.

**Exercise 8.3:** (5 points)

Consider again the toy programming language given in Ex. 6.1. Use yacc to write a parser for this grammar (excluding the while-loop). Your parser should give the variable evaluation after program execution (you are allowed to restrict the number of allowed variables in advance).

Please send your source code with at least one nontrivial example run to `christina.jansen@cs.rwth-aachen.de`. Include your names and matriculation numbers in the subject!

**Exercise 8.4:** (2+2+2 points)

To conclude the syntactic analysis, answer the following questions.

- Which approaches towards the following parsing methods do you know? What are their differences?
  - top-down parsing
  - bottom-up parsing
- Sketch the class diagram of the grammar characterisations learned so far. Try to give example languages, that justify your diagram.
- As observed in the lecture, every language which can be described by a regular expression can be described by a grammar, too. Give reasons why we should use regular analysis for lexical analysis, though.