apl. Prof. Dr. Thomas Noll
Sebastian Junges, Christoph Matheja, Matthias Volk

Compiler Construction 2017
— Exercise Sheet 4 —

Hand in until June 13th before the exercise class.

General Remarks

- If your group has less than 3 members, please post in the L2P forum.
- Submitted solutions which are plagiarised will be graded with 0 points.

Exercise 1 (3 Points)

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

Exercise 2 (5 Points)

Consider the grammar $G = (N, \Sigma, P, start)$ covering some boolean expressions:

- $N := \{\text{start}, \text{guard}, \text{rel}\}$
- $\Sigma := \{\texttt{AND}, \texttt{OR}, \texttt{ID}, \texttt{EQ}, \texttt{LEQ}\}$
- P:

start	$\to$	guard
guard	$\to$	rel $\mid$ guard `AND` guard $\mid$ guard `OR` guard
rel	$\to$	`ID EQ ID` $\mid$ `ID LEQ ID`

(a) Construct $NTA(G)$.
(Either give a transition table as in the lecture or depict the automaton and specify what the edge labelling means. Do not forget to give a numbering to the grammar rules.)

(b) Provide a run of $NTA(G)$ on the input `ID EQ ID AND ID LEQ ID`.

(c) Construct an equivalent grammar G' with $G' \in LL(1)$. Prove $G' \in LL(1)$.

(d) Specify the deterministic top-down parsing automaton $DTA(G')$.
(Again, either give a transition table as in the lecture or depict the automaton and specify what the edge labelling means. As before, do not forget to give a numbering to the grammar rules of G'.)

(e) Provide a run of $DTA(G')$ on the input `ID EQ ID AND ID LEQ ID`.