Exercise 1 (Classification): (3 Points)

In the lecture you have seen different categorisations of CFGs.

a) Provide a language that is recognised by an $LL(1)$ grammar for which there does not exists any $LL(0)$ grammar.

b) Show there are regular languages for which no $LR(0)$ grammars exist.

c) Show there is an $LR(0)$ grammar which is not an $LL(1)$ grammar. *Note, here we do not ask anything about their languages.*

Exercise 2 (LR(1) and LALR(1)): (4 Points)

Let grammar G be given by:

$$\begin{array}{lcl} S' & \rightarrow & S \\ S & \rightarrow & Aa \mid bAc \mid Bc \mid bBa \\ A & \rightarrow & d \\ B & \rightarrow & d \end{array}$$

a) Check whether $G \in LR(1)$ by computing the $LR(1)$-sets of G.

b) Is $G \in LALR(1)$? Justify your answer.

Exercise 3 (LR(0) parser): (3 Points)

Previously, you have generated the $LR(0)$ sets which constitute the states of the so called goto-automaton. Using this, we can implement our first parser which will be an $LR(0)$ parser. Please use the framework that can be found along with this sheet on our web page.

- Implement the method Parser.parse(List<Symbol>)

In the previous exercise, we have seen that the grammar of our WHILE language is not an LR(0) grammar. Therefore we have provided simpler grammars for testing your implementation.