

Exercise Sheet 4

Assignment 4.1 Top-Down-Parsing

Consider the grammar $G = (N, T, \delta, S)$ with $T = \{s, a\}$, $N = \{S, A\}$, start state S and the following production rules

$$\begin{aligned}\delta : \quad S &\rightarrow A s \\ A &\rightarrow a A \mid \epsilon\end{aligned}$$

1. Construct the Item Pushdown Automaton M_G^L for G following the algorithm introduced in the lecture. Split your set of transitions in the three disjoint sets expansions, shifts, and reductions.
2. Can the automaton M_G^L be used to construct a deterministic $LL(0)$ -Parser? Justify your answer!
3. Construct the lookahead sets First_1 for G !
4. Construct the Follow_1 sets for G !
5. Construct the lookahead table for M_G^L with lookahead 1!

Assignment 4.2 Recursive descent parser

Complete the implementation (`Parser.java`) of the recursive descent parser for the grammar of Assignment 4.1. If the input is accepted a success message should be printed out. Otherwise the reason why the parsing failed should be printed out.

Assignment 4.3 Grammar for regular expressions

We want to prepare a implementation of a parser for string-reperesented regular expressions. (The implementation is part of the next exercise sheet.)

1. Give a grammar for regular expressions that is $LL(1)$. We consider regular expressions as defined in the lecture, i.e., we have operators \cdot , $|$, $*$.
2. Prove that the grammar is $LL(1)$.

Hint: You may start with any grammar for regular expressions and then transform this grammar if it is not already $LL(1)$.