Formal Languages and Compilers – 2013, session 2

[12 marks] Exercise 1

Let $\mathcal{G}$ be defined as follows:

$$\begin{array}{rcl} S & \rightarrow & aS \mid aSb \mid T \\ T & \rightarrow & aTa \mid a \end{array}$$

1. Show that $\mathcal{G}$ is not LALR(1).
2. Provide a LALR(1) grammar $\mathcal{G}'$ such that $\mathcal{L}(\mathcal{G}') = \mathcal{L}(\mathcal{G})$.
3. Using the LALR(1) parsing table for $\mathcal{G}'$, show the parsing steps on input *aaabb* and draw the resulting parse tree.

[8 marks] Exercise 2

Provide the minimum DFA to recognize the language generated by the following regular expression:

$$(ba)^*(b \mid a \mid \epsilon)(ba)^*(b^* \mid \epsilon).$$

[7 marks] Exercise 3

Let $\mathcal{G}$ be the following grammar for binary numbers:

$$\begin{array}{rcl} S & \rightarrow & L \\ L & \rightarrow & LB \mid B \\ B & \rightarrow & 0 \mid 1 \end{array}$$

1. Add attribution rules to $\mathcal{G}$ so that the attribute $S.val$ of the start symbol contains the decimal value of the generated binary number.
2. Show the evaluation of $S.val$ for the derivation of 101.

[3 marks] Exercise 4

Let $\mathcal{G}$ be the following grammar:

$$\begin{array}{rcl} S & \rightarrow & Aa \mid Bb \\ A & \rightarrow & aAb \mid ab \\ B & \rightarrow & aBbb \mid abb \end{array}$$

Explain why $\mathcal{G}$ is neither SLR nor LALR(1).