University of Camerino
School of Science and Technologies

Formal Languages and Compilers
Project Paper

Prof. Andrea Polini

The project to be developed asks students to derive a compiler for a regular expression grammar that generate a program able to distinguish words belonging to the language generated by the defined Regular Expression (RExp). In order to derive the program students can refer to the regular automaton defined by Thompson as corresponding acceptance automaton for a RExp. Such rules need to be codified in a programming language (e.g. Java). The result will be a program in the selected language that then will be compiled to provide the needed funtionality.

For istance the program generated by the compiler, given the following generic RE on the alphabet $\Sigma = \{a, b, c, \cdots, z, A, B, C, \cdots, Z, 0, 1, 2, \cdots, 9, ., -, \_\}$ should reply "OK" or "KO" as specified in the following:

RE	**word**	**Result**
a+b	a	OK
	b	OK
	ac	KO
a*b	aab	OK
	b	OK
	aaaab	OK
	abb	KO

In order to derive the compiler the following grammar for RE should be considered:

$$
\begin{array}{lll}
\mathit{RE} & ::= & \mathit{union} \mid \mathit{simpleRE} \\
\mathit{union} & ::= & \mathit{simpleRE} + \mathit{RE} \\
\mathit{simpleRE} & ::= & \mathit{concatenation} \mid \mathit{basicRE} \\
\mathit{concatenation} & ::= & \mathit{basicRE}\ \mathit{simpleRE} \\
\mathit{basicRE} & ::= & \mathit{group} \mid \mathit{any} \mid \mathit{char} \\
\mathit{group} & ::= & (\mathit{RE}) \mid (\mathit{RE})* \mid (\mathit{RE})+ \\
\mathit{any} & ::= & ? \\
\mathit{char} & ::= & a \mid b \mid c \mid \cdots \mid z \mid A \mid B \mid C \mid \cdots \mid Z \mid 0 \mid 1 \mid 2 \mid \cdots \mid 9 \mid . \mid - \mid \_
\end{array}
$$

Suggestion: before starting the ANTLR project can be useful to write a simple program able to behave according to a statically defined regular automaton with ϵ-transitions. For such an automaton actions are driven by a transition table that need to be suitably represented. In particular it will be necessary to define a data structure that can be used to represent the transition table of a NDFSA.