TD 1 – RE and $LL(k)$

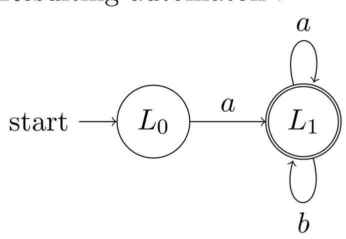
A. Bonenfant, H. Cassé, E. Rio

Lexical Analysis

1) A simple way to transform a RE r into an automaton is to express r in left linear form with different languages. Each language will represent an automaton state, state with λ are final and expressions like $a\ L$ a transition to state L with a. Let take as example $a.(a+b)^*$:

$$
\begin{aligned}
L_0 &= a.(a+b)^* \\
&= a.L_1 && \text{Introduction of } L_1 \\
L_1 &= (a+b)^* \\
&= (a+b).(a+b)^* + \lambda && \text{Reverse Arden} \\
&= a.(a+b)^* + b.(a+b)^* + \lambda && \text{Distribution} \\
&= a.L_1 + b.L_1 + \lambda && \text{Re-use of } L_1
\end{aligned}
$$

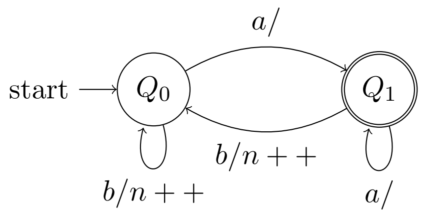
Resulting automaton :



2) Compute the automaton for the RE $a.b^* + (a+b).c^*$.

3) Actions can be added to automata along the edge $a/action$ with $a \in \Sigma$ and *action* any algorithmic language. For example, for the RE $(a+b)^*a$, we may count the number of b :

let $n \leftarrow 0$



Following this example, propose a RE to represent signed integers, propose the corresponding automaton and add actions to compute in n the value of the integer. It has to recognize 123, -39, $+1024$, ...

$D = \{0, 1, ..., 9\}$ is the set of decimal digits, p is the positive symbol $+$ and m is the negative symbol $-$. The function $v : \Sigma \to \mathbb{N}$ converts a character to its digit value.

Syntactic Analysis

1) Let the augmented grammar G_1 with axiom S' below :

$$\begin{array}{ll} S' \to S\ \$ & (0) \\ S \to a\ S\ b\ S & (1) \\ S \to \lambda & (2) \end{array}$$

Show that G is $LL(1)$, infer the analysis table and parse the word "$abab\$$".

2) Let the augmented grammar G_2 with axiom S' below :

$$\begin{array}{ll} S' \to S\ \$^k & (0) \\ S \to b\ R\ S & (1) \\ S \to R\ c\ S\ a & (2) \\ S \to \lambda & (3) \\ R \to a\ c\ R & (4) \\ R \to b & (5) \end{array}$$

Find k s.t. G_2 is $LL(k)$. Give the analysis table and parse the words "$acb\$^k$" and "$bcbcaa\$^k$".

3) Let the augmented grammar G_3 with axiom S' below :

$$\begin{array}{ll} S' \to S\ \$^k & (0) \\ S \to a\ R & (1) \\ S \to R\ b & (2) \\ S \to a\ b & (3) \\ R \to c\ R & (4) \\ R \to \lambda & (5) \end{array}$$

Compute the language $L(G_3)$. Find k s.t. G_3 is $LL(k)$. Give the analysis table and parse the word "$ccb\$^k$".

4) Let the grammar G_4 below :

$$
\begin{array}{lr}
S' \to S\ \$^k & (0) \\
S \to a\ A & (1) \\
A \to b\ B & (2) \\
B \to c & (3)
\end{array}
$$

Find k s.t. G_4 is $LL(k)$. Give the analysis table.

5) Let the partial grammar G_5 below :

$$
\begin{array}{rll}
< \mathit{program} > & \to \texttt{program}\ < \mathit{declList} >\ \texttt{begin}\ < \mathit{instList} >\ \texttt{end} & (0) \\
< \mathit{instList} > & \to < \mathit{inst} > & (1) \\
 & \to < \mathit{instList} >\ \texttt{;} < \mathit{inst} > & (2) \\
< \mathit{inst} > & \to \texttt{if} < \mathit{exp} > \texttt{then} < \mathit{instList} > \texttt{else} < \mathit{instList} > \texttt{endif} & (3) \\
 & \to \texttt{while} < \mathit{exp} > \texttt{loop} < \mathit{instList} > \texttt{endloop} & (4) \\
 & \to \texttt{repeat} < \mathit{instList} > \texttt{until} < \mathit{exp} > \texttt{endloop} & (5) \\
 & \to \mathit{ID}\ \texttt{:=}\ < \mathit{exp} > & (6) \\
< \mathit{declList} > & \to ... & \\
< \mathit{exp} > & \to ... &
\end{array}
$$

Is G_5 $LL(k)$? Why do we need to transform G_5 to make it $LL(k)$? And how ?

6) Let the grammar G_6 below :

$$
\begin{array}{lr}
S' \to S\ \$^k & (0) \\
S \to a\ A\ a\ b & (1) \\
\quad \to b\ A\ b & (2) \\
A \to c\ A\ B & (3) \\
\quad \to a & (4) \\
\quad \to \lambda & (5) \\
B \to \lambda & (6)
\end{array}
$$