

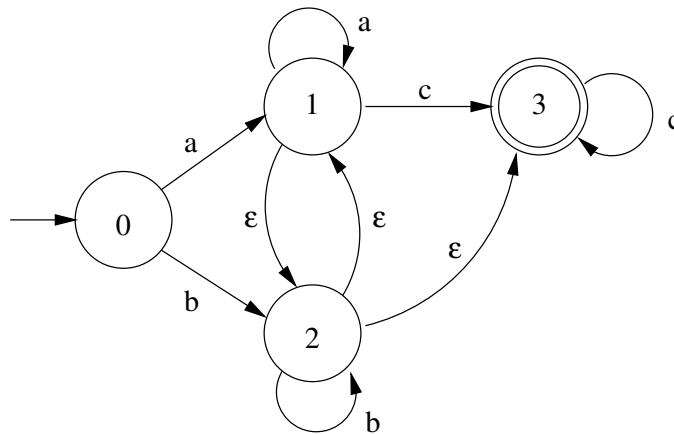
Formal Languages and Compilers - Exercises I

MSc in Computer Science, University of Camerino
prof. Luca Tesei

Note Regular expressions are written with the usual precedence order: operator $*$ has precedence on concatenation, which has precedence on $|$. Moreover, the usual shorthands $^+$ and $?$ may be used.

Exercise 1

Write a regular expression denoting the language accepted by the following automaton:



Exercise 2

Use Thompson algorithm to construct an NFA accepting the language denoted by $(ab|ac)^*d$.

Exercise 3

Write a minimal automaton for the language $(a|b)^* | (b|c)^*d$.

Exercise 4

Consider the following grammar:

$$\begin{aligned} S &\rightarrow aSb \mid Ad \mid Bc \\ A &\rightarrow Aa \mid c \\ B &\rightarrow ddA \mid dC \\ C &\rightarrow ac \end{aligned}$$

1. Formalise the language generated by the grammar
2. Is the grammar $LL(k)$ for some k ?
3. Construct the table of a top-down non-recursive predictive parser for the language.

Exercise 5

Consider the following grammar:

$$\begin{aligned} S &\rightarrow A \mid Bbb \\ A &\rightarrow aB \\ B &\rightarrow aAb \mid b \end{aligned}$$

1. Formalise the language generated by the grammar
2. Is the grammar LR(1)?
3. Is the string $aaAb$ a viable prefix? If the answer is yes, enumerate the valid $LR(0)$ items for this prefix.