Tutorial 03

University of Victoria

CSC 320 - Spring 2023

Foundations of Computer Science

Teaching Team

Learning Outcomes:

- Design a regular expression for a language.
- Convert a regular expression to an NFA.
- Convert a DFA to a regular language.

Interesting Article:

"Compressing Regular Expressions' DFA Table by Matrix Decomposition" [1]

January 31st, 2023

Question 3.01

Design a regular expression for the following languages over the alphabet $\Sigma = \{0, 1\}$:

(a) $L_1 = \{w \mid$ **every odd position of** w **is a** $1\}$

(b) $L_2 = \{w \mid w$ **is a string of length at most** $5\}$

(c) $L_3 = \{w \mid w$ **contains an even number of** 0**'s or exactly two** 1**'s**$\}$

Question 3.02

Convert the following regular expression to an NFA:

$$R_1 = (a \cup b^*)a$$

DFA to Regular Expression

If a language is regular, then there exists some regular expression that describes it.

Transform the following DFA into a Regular Expression:

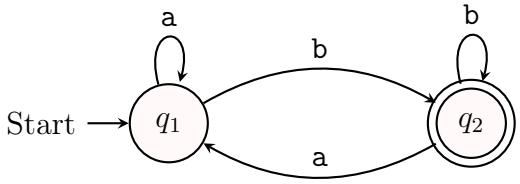

Figure 1: DFA

where the DFA can be describe by the following:

$$DFA = (\{q_1, q_2\}, \{a, b\}, \delta, q_1, \{q_2\})$$

and δ is defined as:

δ	a	b
q_1	q_1	q_2
q_2	q_1	q_2

Table 2: DFA - Transition Table

Bibliography

[1] Y. Liu, L. Guo, P. Liu, and J. Tan, "Compressing Regular Expressions' DFA Table by Matrix Decomposition," English, in *Implementation and Application of Automata*, ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 282–289, ISBN: 3642180973.