PAL labs 8

9 / 11 / 2022

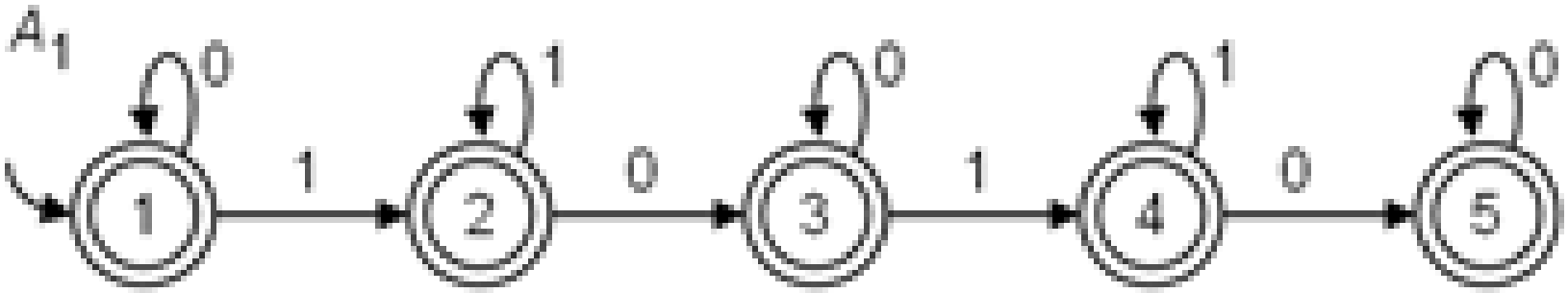
Let's have the alphabet $A = \{a, b, c, ..., z\}$. Let the symbol a be of order 1, b be of order 2, etc., up to z be of the order 26. We call a word over A *ordered* iff it holds that each symbol in the word is of smaller order than successors of that symbols in that world. Design an NFA which finds all *ordered* words in a text over the alphabet A.

Design an NFA over the alphabet $\{0, 1, 2\}$ which finds all occurrences of a string containing the same number of 0, 1 and 2 in a text.

Decide if these two regular expressions represent the same regular language.

1. $(01 + 0)^*0$
2. $0(10 + 0)^*$

Informally describe a which language is accepted by the automaton over the alphabet $\{0, 1\}$. Write down a regular expression describing that language.



Write down a regular expression describing the maximal (w.r.t. inclusion) set M of strings over the alphabet $\{a, b, c\}$ such that:

1. each string in M starts and ends with the symbol b,

2. each string in M contains exactly one symbol c ,

3. each string in M cannot contain the symbol a on odd position (positions are numbered from 1)

Automaton A_1 accepts language L_1, automaton A_2 accepts language L_2. Both automatons have n states. The alphabet is the same for both automatons and contains k letters. What is the asymptotic complexity of an algorithm which decide if the language $L_1 \cap L_2$ is finite?

In a text over the alphabet $\{a, b, c, d\}$, we are asked to find all occurrences of all substrings which begin and end with b, and they have Hamming distance greater than 2 from the pattern *abbbcdabbcdab*. Design a non-deterministic automaton to solve the task.

Design an automaton which finds all occurrences of words which have Levenshtein distance up to k from a pattern, using epsilon transition, in a text, for pattern of length 6 and $k = 3$. How will an equivalent automaton look after removing epsilon transition?