Regular Expressions and Non-deterministic Finite Automata

Prof. Shrisha Rao
IIIT Bangalore

srao@iiitb.ac.in

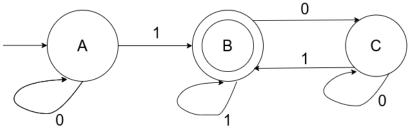
2024-08-08

Recap: Finite Automata

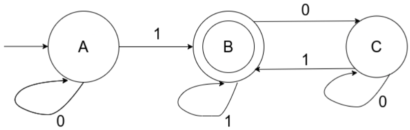
- A FA M is given by the 5-tuple

$$M = \langle Q, \Sigma, \delta, q_o, F \rangle$$

- Exercise: Give the 5-tuple definition for this automaton, and describe the language accepted by it:



Definition and Language of the FA



This FA clearly accepts any string that ends in a 1. The five elements are as follows.

- $Q = \{A, B, C\}$
- $\Sigma = \{0, 1\}$
- δ is given by:

δ	0	1
A	A	B
B	C	B
C	C	B

- $q_0 = A$
- $F = \{B\}$

Exercises

For all of these, give a bubble diagram representation as well as writing down the appropriate 5-tuple specification.

(a) An FA which accepts all binary strings where 0s and 1s alternate.

(b) An FA with only 0s, or alternating 1s and 0s.

(c) An FA that accepts all binary strings with an even number of 0s and an odd number of 1s.

(d) An FA which accepts all binary strings that begin and end with the same symbol.

The Language of a FA

- The language of a FA is typically an infinite set.
- Note that $\emptyset$ is a language, as it is a set.
- We also have a symbol ϵ for "the empty string" that consists of no symbols. ϵ may be a part of some languages if the corresponding FAs accept the empty string (i.e., start in an accepting state).
- Points to note: $\emptyset \neq \epsilon$, and $\{\epsilon\} \neq \emptyset$, i.e., the empty set is not the same as the empty string, and the language containing only the empty string is not the empty set.
- Exercise: give an FA whose language is $\{\epsilon\}$, and one whose language is $\emptyset$.

Regular Expressions

A *regular expression* is a concise way of denoting a language called a *regular language*. It has just these four syntax elements:

- Union, meaning $\cup$ but usually denoted by $+$.
- Concatenation, denoted by $\cdot$ or nothing.
- Star (also called Kleene-star), denoted by a superscript asterisk $*$.
- Parentheses (,) for explicit ordering.

Regular Expression Syntax

- $0+1$ literally means the set $\{0,1\}$, the union of $\{0\}$ and $\{1\}$.
- $0 \cdot 1$, more commonly 01, is just the string 01.
- 1^* literally means "1 repeated zero or more times."

$$1^* = \{\epsilon, 1, 11, 111, 1111, \ldots\}$$

- Question: What is $(01)^*$?

ज्ञानमुत्तमम्

Examples of Regular Expressions

- $(01)^* = \{\epsilon, 01, 0101, 010101, \ldots\}$
- $1(01)^*0 = \{10, 1010, 101010, 10101010, \ldots\}$
- $(0 + 1)^*$ is a very special regular expression. What language does it represent?

The RE $(0+1)^*$

- $(0+1)^*$ is the same as $\{0,1\}^*$.
- It can be expanded as
  $\epsilon + (0+1) + (0+1)(0+1) + (0+1)(0+1)(0+1)\dots$
- This comes to $\{\epsilon, 0, 1, 00, 01, 10, 11, \dots\}$
- Therefore, $(0+1)^*$ is the language of all possible binary strings including the empty string.

Writing a RE given a language description

(1) Write the regular expression for all non-empty strings over $\{0, 1\}$ that begin and end in the same symbol.

The RE for the Language

Solution idea:
We need to consider just 0 and 1, and then also one RE each for strings beginning and ending in 0, and beginning and ending in 1.
Answer:

$$0 + 1 + 0(0+1)^*0 + 1(0+1)^*1$$

Construct more REs

Give regular expressions for each of the following languages.

(2) All binary strings that contain at least one 0.

(3) All binary strings that contain at most one 1.

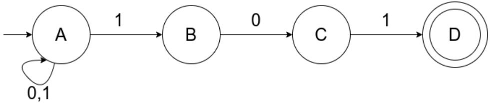
(4) All strings over $\{a, b, c\}$ where the symbols are in alphabetical order.

Non-Deterministic Finite Automaton (NFA)

- A FA that is *non-deterministic*, i.e., may choose one of several possible transitions given an input, is called a *non-deterministic finite automaton*, NFA for short.
- A FA that is not non-deterministic, i.e., has a deterministic transition function, is called a DFA to distinguish it from an NFA.
- An NFA is said to accept a string if for *some* possible transition given that string as input, it ends in an accept state.
- NFAs are often easier to construct to accept given strings/languages, than DFAs.

Example of an NFA

(5) Give the RE for the language accepted by the following NFA.



More NFAs

Give NFAs to accept the following languages:

(6) The set of all strings with only an even number of 0s, or only exactly two 1s.

(7) The language 0^*.

(8) The set of all strings where every odd position is a 1.