

# Cheatsheet - Automata Theory

Fabio Lama – fabio.lama@pm.me

## 1. Basics of (finite) Automata

An **alphabet**,  $\Sigma$ , is a non-empty set of symbols.

$$\Sigma = \{0, 1\} \text{ binary alphabet}$$

$$\Sigma = \{a, b, \dots, z\} \text{ collection of lowercase letters}$$

A **string** or word is a finite sequence of letters drawn from an alphabet. **Empty strings**,  $\varepsilon$ , are strings with zero occurrences of letters. Empty strings can be from any alphabet.

The **length** of a string  $x$  is denoted as  $|x|$ :

$$x = \text{'hello'}$$

$$|x| = 5$$

Other string related notations:

- The set of **all strings** composed from the letters in  $\Sigma$  is denoted by  $\Sigma^*$ .
- The set of **all non-empty strings** composed from letters in  $\Sigma$  is denoted by  $\Sigma^+$ .
- The set of **all strings of length  $k$**  composed from letters in  $\Sigma$  is denoted by  $\Sigma^k$ .

$$\Sigma = \{0, 1\}$$

$$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, \dots\}$$

$$\Sigma^+ = \{0, 1, 00, 01, 10, 11, \dots\}$$

$$\Sigma^2 = \{00, 01, 10, 11\}$$

Note that the size of  $\Sigma^k$  is denoted as  $|\Sigma|^k$ .

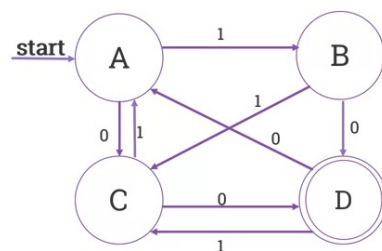
## 2. Automaton

A **finite automaton** is a simple mathematical machine; it is a representation of how computations are performed with *limited memory* space. It is a model of computation, which consists of a set of states that are connected by transitions. It has an input and it has an output.

An automaton  $M$  is a 5-tuple  $(Q, \Sigma, \delta, q_0, F)$  where:

- $Q$  is a finite set called the **states**.
- $\Sigma$  is a finite set called **the alphabet**.
- $\delta: Q \times E \rightarrow Q$  is the **transition function**.
- $q_0 \in Q$  is the **start state**.
- $F \subseteq Q$  is the set of **accepted states**.

### Example



- $Q = \{A, B, C, D\}$
- $\Sigma = \{0, 1\}$
- $q_0 = A$
- $F = \{D\}$
- $\delta = ?$

$\delta$	0	1
A	C	B
B	D	C
C	D	A
*D	A	C

Figure 1. Note: Bottom right is the transition table. D is the only accepted state.

For example, based on the automaton in the picture above:

- Input 0010 results in state  $D$ ; the input is **accepted** ( $D \in F$ ).
- Input 0011 results in state  $A$ ; the input is **rejected** ( $A \notin F$ ).

## 2.1. Language of Automaton

The set of all strings accepted by an automaton is called the **language** of that automaton. If  $M$  is an automaton on alphabet  $\Sigma$ , then  $\mathcal{L}(M)$  is the language of  $M$ :

$$\mathcal{L}(M) = \{x \in \Sigma^* \mid M \text{ accepts } x\}$$

## 3. Determinism

The difference between Deterministic Finite Automata (DFA) and Nondeterministic Finite Automata (NFA) is that DFA has exactly one transition and there is a unique starting state, which is not the case in NFA (hence, **nondeterministic**). Generally, in NFA there are many choices at one particular point and an input is accepted if at least one sequence of choices leads to an accepting state.

Last updated 2023-03-11 16:33:37 UTC