Deterministic Finite Automata

Prof. (Dr.) K.R. Chowdhary

Former Professor & Head, Department of Computer Sc. & Engineering MBM Engineering College, Jodhpur

SPRINGER NATURE



K.R. Chowdhary

Theory of Computation

Automata, Formal Languages, Computation and Complexity

Focuses on pedagogy in its writing, that represents a refreshing approach

Ensures comprehensive and enjoyable learning

Undergone a rigorous classroom testing



©2025



Get 20% off with this code: SPRAUT

Available on Springer Nature Link



link.springer.com/book/ 9789819762347

Plaus note that promotional courses are only wait for tright-language Sartinger, Areney, and Palgrave Maxmillan books devices and are received to relate software only "This software that the book size is a set working resist and this temporary in at available on the system of the book size to be software and the software works, and the devices and the system method size is a set working of the software and the software works, and the devices associated work is a software of the system for the system of the system method size is not received by non-tem corresponding book with the system of the system. The system of the system

Introduction

- Any physical body or machine, or animal, or even solar system, which changes its **states** with time, can be represented by Automata.
- Given any initial state and final state, we can think of intermediate discrete states, through which transition have taken place.
- Objective: Use automata to model the behaviour of computer, and other real life machines, having finite states or assumed to have finite states.
- Automata theory: Abstract mathematical representation of computational procedures. FA is used in design of lexical analyzers in compilers, for searching patterns in arbitrarily large texts, natural language processing, text processing, etc.
- FA is simplest computing model, it is a restricted program without variables.
- FA shares its features with computer. It has finite a very limited memory, present in CPU only.



Figure 1: Finite Automaton

- A finite automaton has finite set of states Q it can undergo, an alphabet set Σ , a set of accepting or final states F, where $F \subseteq Q$, a starting state s, and a transition function δ , where $\delta : Q \times \Sigma = Q$. Thus, a FA $M = (Q, \Sigma, \delta, s, F)$.
- If a FA is in state p and makes a transition to state q on reading of symbol $a \in \Sigma$, then this transition is represented as $\delta(p, a) = q$.

Finite automata

- If there is an input w = abcd on tape, and transitions are like this: $\delta(p,a) = q, \delta(q,b) = r, \delta(r,c) = s, \delta(s,d) = t$. Thus, at the begin we have state and input as (p, abcd), which is called as initial **configuration** or **ID**(Instantaneous description) of the FA.
- The sequence of transitions through which it will go are:

 $(p, abcd) \vdash_M (q, bcd) \vdash_M (r, cd) \vdash_M (s, d) \vdash_M (t, \varepsilon)$ or we can say that configuration (p, abcd) goes to configuation (t, ε) through zero or more transitions, written as: $(p, abcd) \vdash_M^* (t, \varepsilon)$; the symbol \vdash is called "derives."

- The language of M:
 L = L(M) = {w|w ∈ Σ*, (p, abcd) ⊢^{*}_M (t, ε), and t ∈ F}, ⊢^{*}_M is transitive relation, and defined as ⊢_M: Q × Σ* → Q × Σ*
- Given any input symbol and current state there is a definite next state. Thus, given a start state, and input string it is possible to determine entire behaviour of a FA. Hence, this FA is called DFA (deterministic FA).

FA Example

• What is regular expressions and FA for:

 $Q=\{q_0,q_1,q_2\}, \Sigma=\{0,1\}, F=\{q_1\},$ and δ is given as:

CL	irrent	in	put		
st	ate	0		1	
q_0)	q_0		q_1	
q_1	q_1			q_2	
q_2	q_2			q_1	
	0	(q1)	1	q2	0

Figure 2: Transition diagram of corresponding FA.

Regular expression = $0^*1(0^+1+10^*1)^*$

• Which of following are FA? Justify.

- Digital computer,
- analog computer,
- I digital voltmeter,
- analog voltmeter,
- ohemical reaction,
- interaction of radiation with matter,
- Itransformation of water into vapor.

Chowdhary, K.R. (2025). Finite Automata and Regular Expressions. In: Theory of Computation. Springer, Singapore. https://doi.org/10.1007/978-981-97-6234-7_2