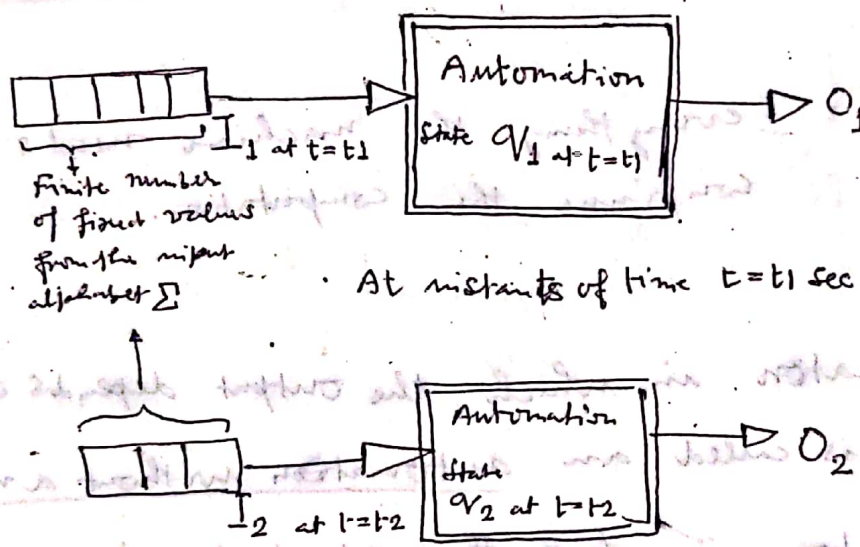
Automata

Automaton is an abstract computing device which performing some functions (tasks) without direct participation of man (human being)

In other words

Automata are abstract mathematical models of machines that perform computations on an input through a series of states.

Discrete automation model



Characteristics of Automata

1. **State:** → [At any instant of time the automaton can be in one of the states $q_1, q_2, q_3, \ldots q_n$]

2. **State relation: (Transitions)** → The next state of an automaton at any instant of time t = t1 sec. is determined by the present state and the present input. (2) is determine by the transition function)

   Output relation → Output is related to either state only or to both the input and the state.

3. **Input:** At each of the discrete instants of time $t_1, t_2, \ldots$ input values $I_1, I_2, \ldots$ each of which can take a finite number of fixed values from the input alphabet Σ, are applied to the input side of model.

4. **Output:** $O_1, O_2, \ldots$ are outputs of the model, each of which can take finite numbers of fixed values from an output O.

5. Computing a function or accepting a language.

6. **Configuration:** everything the machine needs to continue the computation.

NOTE: An automaton in which the output depends only on the input is called an automaton without a memory.

An automaton in which the output depends on the Input & State is called an automaton with a finite memory

- Automaton in which the o/p depends only on the States of the machine called **Moore machine**
- Automaton in which the o/p depends on the State & the Input at any instant of time is called a **Mealy machine**

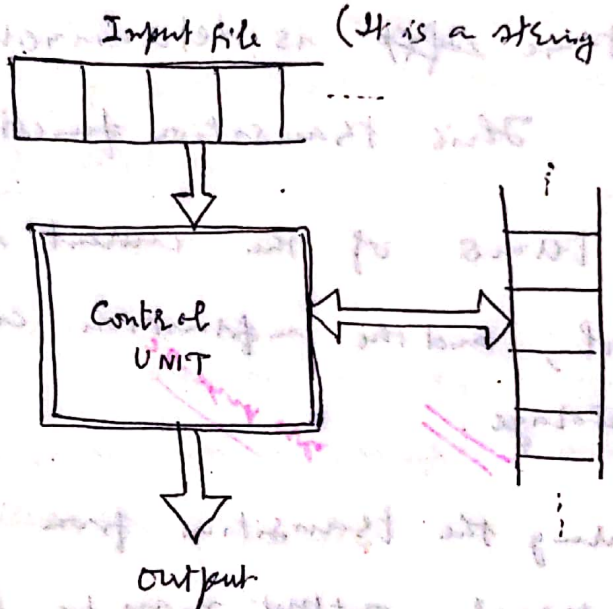
AUTOMATA

An automaton is an abstract model of a digital computer.

Essential features

1) — It has a mechanism for reading input. It can read the input file from left to right, one symbol at a time. This input mechanism can also detect the end of the input string. (EOF condition).

2) — The automaton can produce output of some form.



3) — It may have a temporary storage device, consisting of ~~an unlimited~~ number of cells, each capable of holding a single symbol from an alphabet (It may be different input alphabet). The automaton can read and change the contents of the storage cells.

4) The automaton has a control unit, which can be in any one of a finite number of internal states, automaton can change state in some predefined manner.

Mechanism:

An automaton is assumed to operate in a discrete time frame. At any given time, the control unit is in some internal state, and the input mechanism is scanning a particular symbol on the input file.

The internal state of the control unit at the next time is determined by the transition function. This transition function gives the next state in terms of the current state, the current input symbol, and the information currently in the temporary storage.

During the transition from one time interval to the next, output may be produced or the information in the temporary storage changed.

The term configuration will be used to refer to a particular state of the control unit, input tape and temporary storage.

The transition of the automaton from one configuration to the next configuration will be called a move

Deterministic automata

A deterministic automata is one in which each move is uniquely determined by the current configuration (input, state, storage) so we can predict the future behavior of the automaton exactly.
This is first type of automaton or that are deterministic in their operation.

Non deterministic automata

A non deterministic automata may have several possible moves, so we can only predict a set of possible actions.

Accepter An automaton whose output response is limited to a simple "yes" or "no" is called an accepter.

Transducer A more general automaton, capable of producing strings of symbols as output, is called a transducer.

Types of Automata

1. Finite automata — regular languages
2. Push-down automata — context-free languages
3. Linear-bounded automata — context-sensitive languages
4. Turing machines — recursively enumerable languages
5. Others: Random access machines, parallel random access machines, arrays of automata.

goto page 5

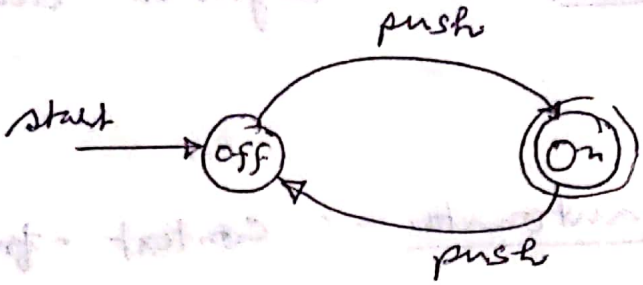
Applications

Finite automata are used as a model for:

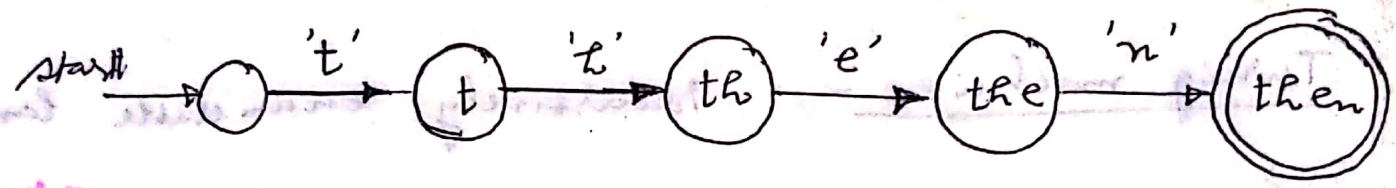
1. Software for designing digital circuits
2. Lexical analyzer of a compiler
3. Searching for keywords in a file or on the web
4. ~~Searching for keywords in a file or on the~~
4. Software for verifying finite state systems, such as communication protocols.

④

Example1: Finite Automaton modelling an on/off switch

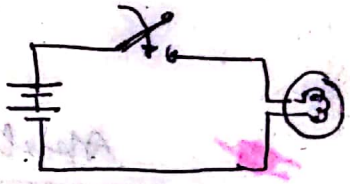


Example2: Finite Automaton recognizing the string "then"



Ex-1

$$M = \left(\{off, on\}, \{push\}, \delta, off, \{on\} \right)$$



δ:

	push
→ off	on
(on)	off

$\delta(off, push) = on$

$\delta(on, push) = off$

FINITE AUTOMATA / Deterministic Finite Acceptors (dfa)

A finite automata (F.A.) consists of a finite set of states and set of transitions from state to state that occur on input symbols chosen from an alphabet 'Σ'. For each input symbol there is exactly one transition out of each state.

Automaton starts with initial state q_0, some states are designated as final or accepting states.

A finite automata is a collection of 5-tuple.

$M = (Q, \Sigma, \delta, q_0, F)$, where

(i) Q is a finite nonempty set of states.

(ii) Σ is a finite nonempty set of inputs, called input alphabet

(iii) δ is the transition function mapping $Q \times \Sigma$ to Q. This is the function which describes the changes of states during transition. This mapping is usally represented by a transition table or transition diagram

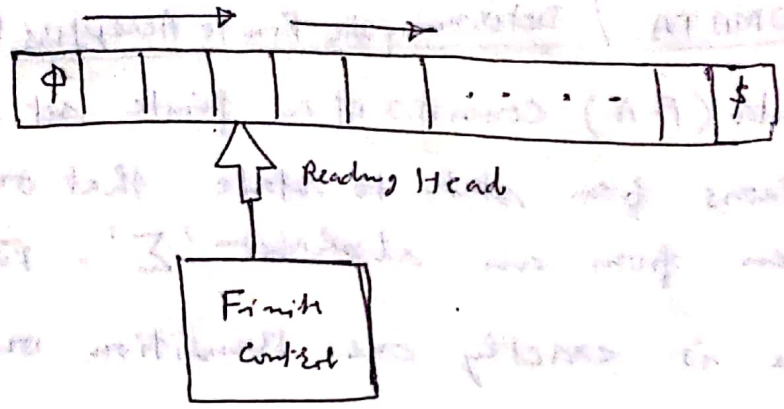
(iv) $q_0 \in Q$ is the initial state.

(v) $F \subseteq Q$ is the set of final states. It is assumed here that there may be more than one final state.

example — ex 4

Mechanism of dfa A dfa operates in the following manner. At the initial time ($t=0$) it is assumed to be in the initial state q_0, with its input mechanism on the leftmost symbol of the input string.

During each move of the automaton, the input mechanism advances one position to the right, so each move consumes one input symbol. When the end of the string is reached, the string is accepted if the automaton is in one of its final states. Otherwise the string is rejected.



The input mechanism can move only from left to right and reads exactly one symbol on each step.

The transitions from one internal state to another are governed by the transition function δ. e.g.

$$\delta(q_0, a) = q_1$$

If the dfa is in state q_0 and the current input symbol is a, the dfa will go into state q_1.

Notations for Transition diagram

(i) q — where q is the name of state

(ii) a → — Transition from one state to another when a is current input

(iii) → S OR $\overline{S}$ — Start or Initial state

(iv) S_F OR $\overleftarrow{S_F}$ — Final state

Languages & DFA : The language accepted by a dfa $M=(Q, \Sigma, \delta, q_0, F)$ is the set of all strings w on Σ accepted by M.

$$L(M)=\{w \in \Sigma^* : \delta^*(q_0, w) \in F\}$$

NOTE: if $M=(Q, \Sigma, \delta, q_0, F)$ is deterministic finite accepter (dfa) then its associated transition graph G_M has exactly $|Q|$ vertices, each one labelled with a different $q_i \in Q$. For every transition rule $\delta(q_i, a)=q_j$ the graph has an edge (q_i, q_j) labeled a.

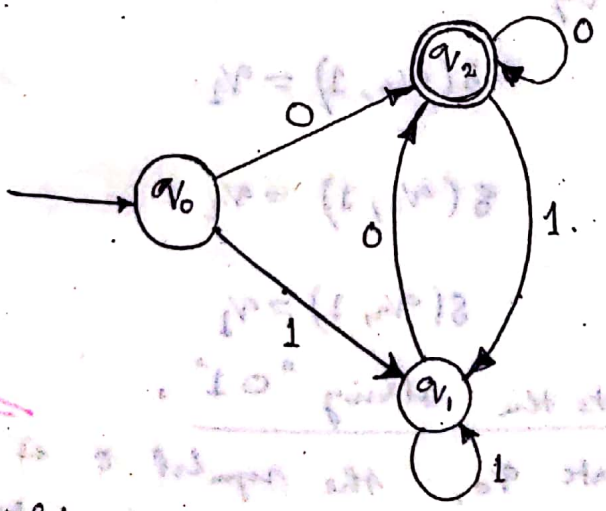
If transition system has more than one initial state, then it will not be a finite automaton.

Q-1 Design a DFA which checks whether the given binary number is even

or

Find a deterministic finite accepter that recognizes the set of all even binary No on $\Sigma=\{0,1\}$

Sol Binary No made up of 0 and 1, when any binary no ends with 0 it is always even and when a binary no ends with 1 it is always odd.



δ	0	1
$\rightarrow q_0$	q_2	q_1
q_1	q_2	q_1
(q_2)	q_2	q_1

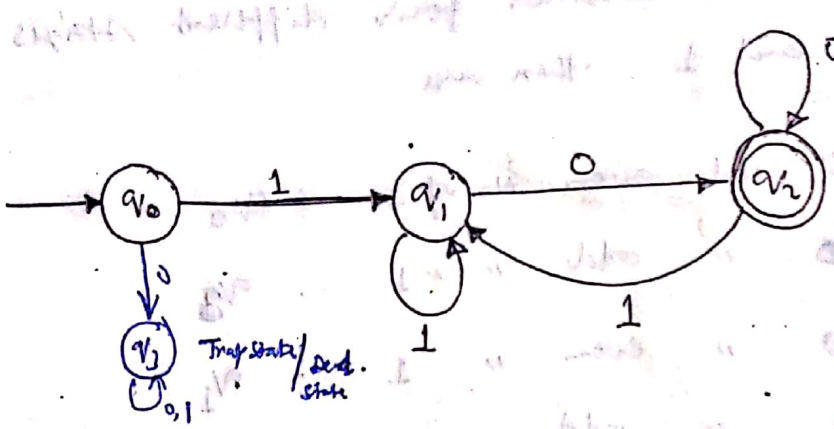
$M=(\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_2\})$

While designing a FA we will assume one state, state q_0 one state ending with 0 i.e q_2 and other state q_1 ending with 1

To check whether given binary no is even or not we will make the state q_2 for 0, the final state

Q-2 Design FA which accepts only those strings which start with 1 and end with 0 over $\Sigma = \{0, 1\}$

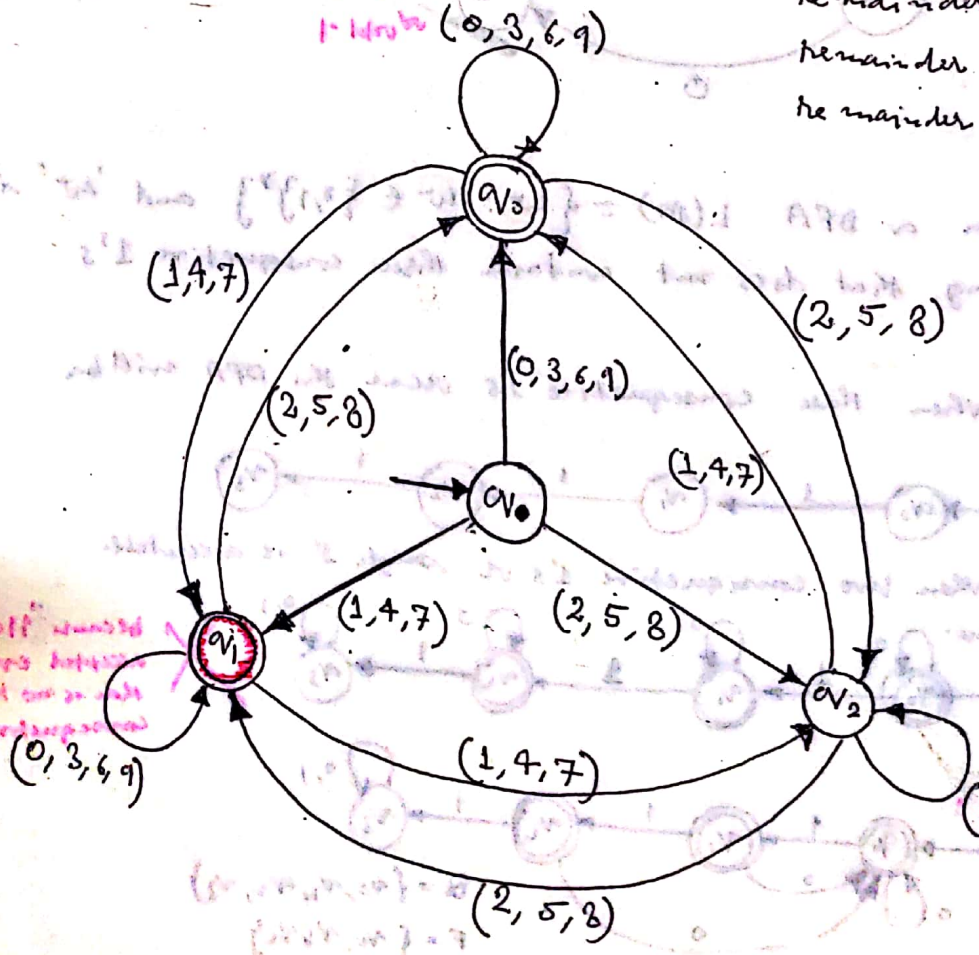
Sol



Q-3 (5) Design FA to check whether given decimal number is divisible by three or not

Sol decimal no consist of $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. To determine whether the given decimal number is divisible by three, we need to take the input no digit by digit and divide it by 3. Possible remainder will be 0, 1 or 2. So we need to group these digits according to their remainders

Remainder 0 group q_0: (0,3,6,9)
Remainder 1 group q_1: (1,4,7)
Remainder 2 group q_2: (2,5,8)



To check whether 2583

$\delta(q_0, 2583) = \delta(q_2, 583)$

$= \delta(q_1, 83)$

$= \delta(q_2, 3)$

$= q_0$

$q_0 \to q_2 \to q_1 \to q_2 \to q_0$

So, 2583 is divisible by 3

Q-4 (4) Design FA which accepts even number of 0's and even number of 1's

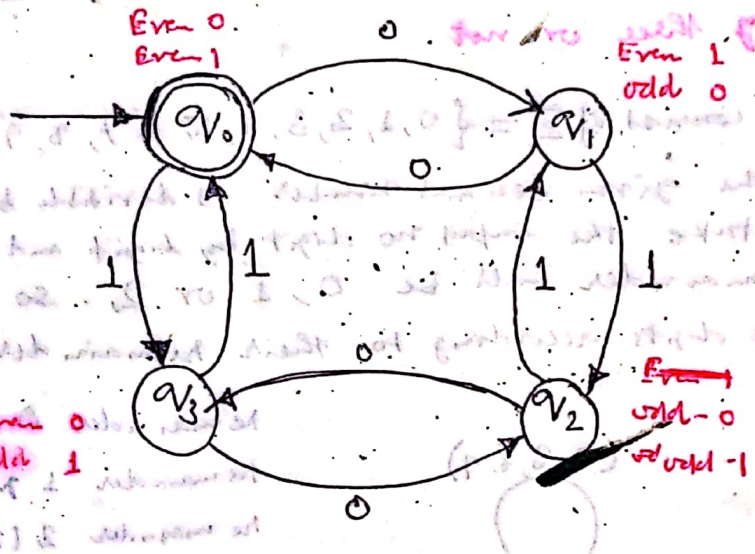
Sol This FA will consider four different stages for input 0 and 1. Then are

even No of 0 and even No of 1 : q_0

" " 0 " odd " " 1 : q_3

odd " " 0 " even " 1 : q_1

" 0 " odd " 1 : q_2



Transition Table

δ	0	1
→ q_0	q_1	q_3
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_2	q_0