lec 1

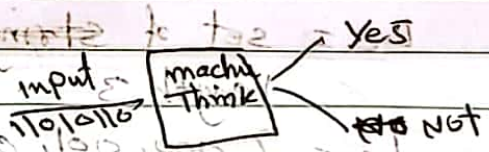
Dr. Hassen mogaibel

Theory of Computation

Introduction

one of the most fundamental course of computer science.
This course will help you to understand how people have Thought about computer science as a science In the past 50 years.
it mainly about what kind of things can you really compute mechanically, how fast and how much space does it take to do so.

For example, assum we want to build machine that accept all binary string that end in 0 and reject all other string.
ex 1101011101 . we need to check whether the string is accepted or not.



example2: accepts all valid Java codes.
Question: is it possible to design such machine (Yes / No)
This done by compiler. The compiler can check whether your code is valid or not.

example3: Accept all valid Java codes and never goes into infinite loop. (✓)

Question: Can I design machine to do above Task ? No, I cannot. (X)

Fsm → Finite state machine
CFL → Context free Language
Turing machine →
i.e
set of string
Undecidable → for Those problem cannot be solved.

Symbol → a, b, c, 0, 1, 2, 3, ---

Alphabet → Σ → collection of symbols

ex: $\{a, b\}$, $\{d, e, F, g\}$.

$\{0, 1, 2\}$

String → sequence of symbols, ex

a, b, 0, 1, aa, bb, ab, 01, aaab --

Language → set of string

ex → $\Sigma = \{0, 1\}$

L_1 = set of all string of length 2

i.e $L_1 = \{00, 01, 10, 11\}$

L_2 = set of ~~string~~ all string of length 3

$L_2 = \{000, 001, 011, 100, 101, 110, 111, 010\}$

L_3 = set of all string that begin with 0

$L_3 = \{0, 00, 01, 000, 001, 010, 011, 0000 \dots\}$

L_3 → This is infinite set

L_1, L_2 is Finite set

Power of Σ: $\Sigma = \{0, 1\}$

Σ^0 = set of all string of length 0 $\Sigma^0 = \{\epsilon\}$

Σ^1 = " " " " " " $\Sigma^1 = \{0, 1\}$

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

Σ^n = " " " " " " n

Cardinality : number of elements in a set

For example $|\Sigma^1| = 2$

$|\Sigma^2| = 4$

$|\Sigma^n| = 2^n$

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cdots$

$= \{\in\} \cup \{0,1\} \cup \{00,01,10,11\} \cup \cdots$

= set of all possible strings of all lengths over $\{0,1\}$

Finite Automata (FA)

Finite Automata is an abstract computing device. it is a mathematic model of systems with discrete mpute, outputs, states and set of transition From state To state that occurs on input symbols From Alphet "Alphabet Σ."

it representations

1) graphical (Transition Diagram)
2) Tabular (Transition table)
3) Mathematical (mapping function).

A finite automata is a 5-tuples, they are

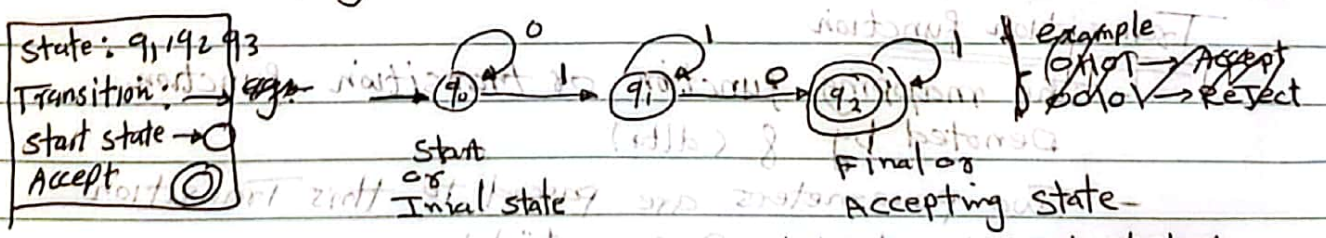
$$M = (Q, \Sigma, \delta, q_0, F)$$

where

Q : is finit set called states
Σ : is a finit set called alphabets
δ : $Q \times \Sigma \to Q$ is transition function
$q_0 \in Q$ is the start state or initial state
$F \subseteq Q$ is set of accept states, also called final state.

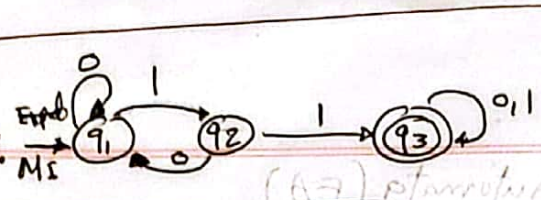
Transition Diagrams (Transition graph)

it is a Direct graph associated with vertics of "graph represents to state of finite automata.



{0,1} is Input
output : Accep or reject

Computation process: begin at start state, read input symbols, Follow Corresponding transition, Accept If end with accept state Reject If Not

Example: 

Example
0110 → Accept
00101 → Reject.

M_1 accept exactly those string in A where $A = \{w \mid w$ contain substring $11\}$

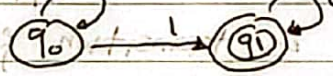
Transition Table

it is basically a tabular representation of the transition function that takes two argument (state and symbol) and return a value (the "next state").

→ Row represents to state
→ Column represents to input symbols
→ entries represents to next state.
→ The start state is marked with an arrow (→)
→ The accepted state is marked with star (*)

	ε	ε	ε	ε
Start state → Q →				
Q →				
Q →				
Q →				

$\delta = Q \times \Sigma \rightarrow Q$
i.e state x symbol → Next state.

example: 

q_0 is initial state & q_1 is final state

	0	1
→ q_0	q_0	q_1
* q_1	q_1	q_1

$q_0 \times 0 \rightarrow q_0$
$q_1 \times 1 \rightarrow q_1$
$q_1 \times 0 \rightarrow q_1$
$q_1 \times 1 \rightarrow q_1$

Transition function

- The mapping function or transition function denoted by δ (dlta)
- Two parameters are passed to this Transition function 1) Current state
  2) Input symbol.

prerequisities

Language: Finite set of none empty string.

If Σ is an alphabet, and $L \subseteq \Sigma^*$, Then L is a Language.

example: The set of legal english words
The set of // C programs
The set of string consist of n 0's Followed by n 1's. $\Rightarrow \{\epsilon, 01, 0011, 000111, \ldots\}$

operations on Languages

① Complementation :-
let L be a language over an alphabet Σ
The complementation of L, denoted by $\overline{L}$

$$\overline{L} = \Sigma^* - L$$

② Union :
let L_1 and L_2 are language over an alphabet Σ. The Union of L_1 and L_2 is denoted by $L_1 \cup L_2$ is $x \mid x$ is in L_1 or L_2

③ Intersection :
let L_1 and L_2 be language over an alphabet Σ. The Intersection of L_1 & L_2 denoted by $L_1 \cap L_2$ is $\{x \mid x$ in L_1 and $L_2\}$

④ Concatenation :-
let L_1 and L_2 be Language over an alphabet Σ.
The Contatenation of L_1 & L_2 is $L_1 \cdot L_2$
$L_1 \cdot L_2 = \{w_1 \cdot w_2 \mid w_1$ is in L_1 and w_2 in $L_2\}$

(v) Reversed:

let L be a Language over an alphabet Σ.

The Reversed of L_1 denoted by

$$L_1^r = \{ w^r / w \text{ is in } L \}$$

(vi) Kleene clouse:

let L be a language over an alphabet Σ

The kleenes clouse of L denoted by

L^*, is $\{ x / \text{ For an Integer } n \geq 0 \}$

For example $x = x_1, x_2, x_3 \cdots x_n$

$$L^* = \bigcup_{i=0}^{\infty} L^i, \quad \left(\text{eg. } a^* = \{ \epsilon, a, aa, aaa, \cdots \} \right)$$

(vii) positive Clog:

let L be a Language over an alphabet Σ

The closue of L, Denoted by L^*

$L^+ = \{ \alpha / \text{ For an integer } n \geq 1, \alpha = x_1, x_2 \cdots x_n$ and $\alpha_1, \alpha_2 \cdots x_n$ are in L $\}$

$$L^+ = \bigcup_{i=1}^{\infty} L^i \quad \left(\text{eg. } a^+ = \{ a, aa, aaa \cdots \} \right)$$

the transition function is always return a state which can be called as next state.

$$\delta(\text{current state}, \text{current-input-symbol}) = \text{next state}$$

i.e $Q \times \Sigma \longrightarrow Q$ (next state)

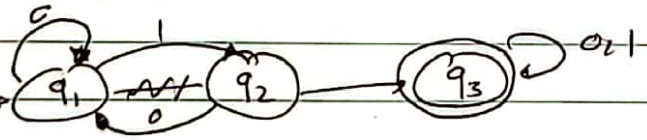
example: $\delta(q_0, 0) = q_1$

$\delta(q_0, 1) = q_1$

Application

* it plays an important role in compiler design
* In switching Theory and design and analysis of digital circuit automata Theory is applied.
* Design and analysis of complex software and hardware systems
* To prove correctness of the program
* To design Finite state machine such as Moore machine
* it is base of the formal languages and these Formal languages are useful of the programming languages.

Example



$M_I = (Q, \Sigma, \delta, q, F)$

$Q = \{q_1, q_2, q_3\}$

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

$F = \{q_3\}$

δ =	0	1
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_3	q_3