3|Jan|18

Theory of Computation

String Σ = any set

→ String over the set Σ is a finite sequence of symbols of Σ

Σ^* = Set of all the string over Σ (including Empty String which is denoted by $\wedge$)

$\Sigma^+ = \Sigma^* - \{\wedge\}$

Basic Operation on Set of String

→ Concatenation

Suppose $x, y \in X^*$

Concatenation of $x \$ y = xy$

" $y \$ x = yx$

Suppose Σ = Set of all English Alphabets

x = ALGOL , Y = PASCAL

xy = ALGOLPASCAL

yx = PASCALALGOL

hence In general $xy \neq yx$

So this Opern is non commutative.

Suppose $z \in \Sigma^*$

$(xy)z = x(yz)$, So this Oper^n is Associative.

If $x \in \Sigma^*$

$x\Lambda = x = \Lambda x$ {Identity Element (Λ)} wrt. to Concatanation.

→ Concatanation is an binary operation (Concatanation) which is associative, has an Identity element is called monoid

Semigroup ⊆ monoid ⊆ group.

→ $zx = zy \Rightarrow x = y$ {Left Cancellation}

$xz = yz \Rightarrow x = y$ {Right " }

→ Cancellation law holds in Σ^* wrt to Concatanatn

→ If $x \in \Sigma^*$, x = ALGOL

then x^T = LOGLA

→ x^T = LOGLA, $x \in \Sigma^*$, $a \in \Sigma$

then

$(ax)^T = x^T a$

$(xa)^T = a x^T$

⇒ $|x|$ → length of string x

$|xy|$ → $|x| + |y|$ length of String x & y.

$|yx|$ → $|y| + |x|$

⇒ Palindrome

$x = x^T$ $\quad$ |malayalam| = 9

xx^T → palindrome of even length

xx^T = ALGOL LOGLA

x → any String

→ prefix of String is a substring of leading symbols of that string

→ Suffix " " " " Trailing " " " "

x = 123

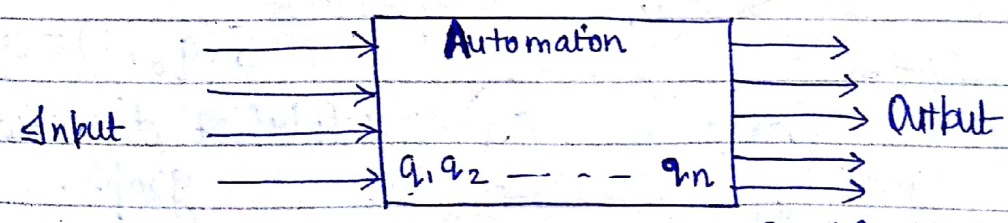
Prefixes {^, 1, 12, 123}

Suffixes {^, 3, 23, 123}

Automaton

$ Inform.

→ It is defined as a System where energy material ^ are transformed or transmitted. And used for performing some functions → without direct participation of human beings



Eg. Lift, photo copy Machine

→ State-Input Relation

→ final State

Defination (finite Automaton) (DFA)

A finite Automaton is defined as a 5 tuple $(Q, \Sigma, \delta, q_0, F)$, where

1.) Q is a finite non Empty set of States
2.) Σ is a finite non Empty set of Input Symbols
3.) $\delta : Q \times \Sigma \to Q$ called a direct transition function.
4.) $q_0 \in Q$ is the initial State of the automaton
5.) $F \subseteq Q$ is the set of final states.

for 3.)

$\delta(q, a) = q'$ It is also called Next State function.

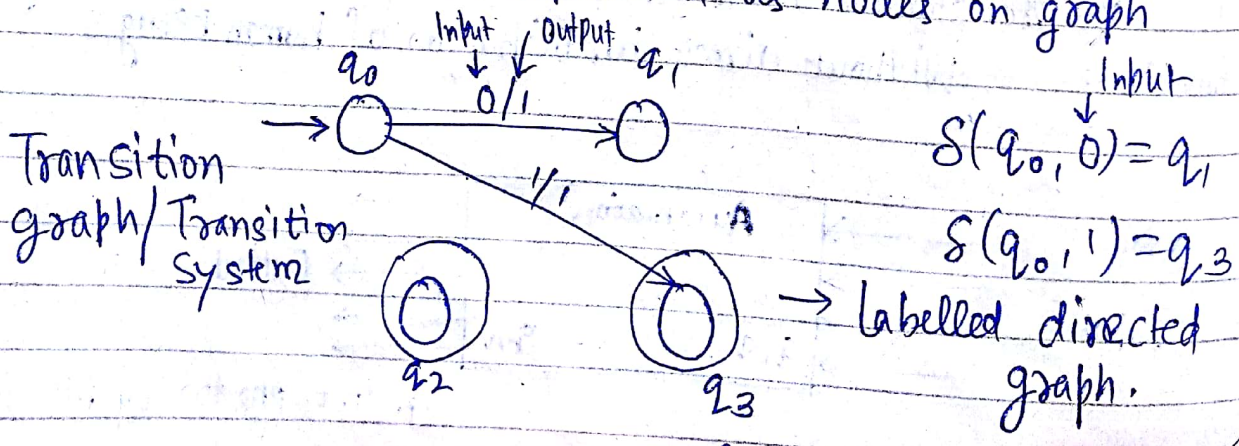
but $\delta(q, a_1 a_2 \ldots a_i) = q'$

Indirect transition funct.

Hence it is also called finite Set Machine (FSM).

Date : 4/Jan/18

→ States can be represented as nodes on graph



Transition graph / Transition System

Input

$\delta(q_0, 0) = q_1$

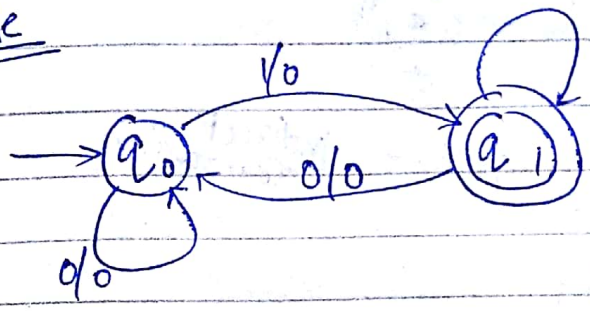
$\delta(q_0, 1) = q_3$

→ labelled directed graph.

Deterministic finite Automaton (DFA)
↳ Can be determine next state

Initial State represented by →
final State " " ◯
Eg. q_3, q_2

Example



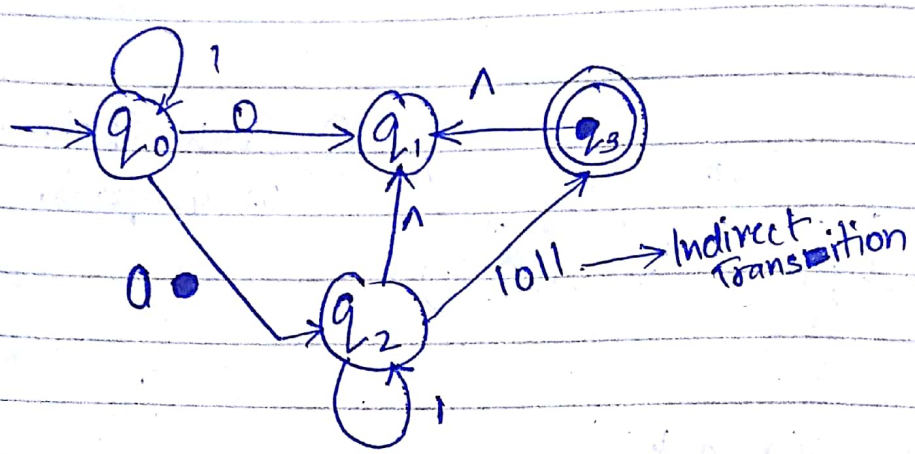
Q Draw Graph from DFA or Graph to DFA ?

$Q = \{q_0, q_1\}$
$\Sigma = \{0, 1\}$
$\delta : Q \times \Sigma \to Q$
→ $\delta(q_0, 0) = q_0$
$\delta(q_0, 1) = q_1$
$\delta(q_1, 0) = q_0$
$\delta(q_1, 1) = q_1$
q_0 = Initial State
$\{q_1\}$ = final State

Def^n A DFA accepts the string w iff
1) ∃ a path which originates from Initial State goes along the arrows & reaches to some final State.

2) The path value obtained by Concatanation of all edges labels of the path is equal to w

Q Consider the Transition graph



$W_1 = 101011$
$W_2 = 111010$
} Is this Accepted by String

① $q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_2 \xrightarrow{1011} q_3$ Accepted

$101011 == 1010\,11$

② $q_0 \xrightarrow{1} q_0 \xrightarrow{1} q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_2 \xrightarrow{1} q_2 \xrightarrow{0}$ No final State

Machine hacults

W_2 = is not Accepted by Machine

Properties :-

1) $\delta(q, \wedge) = q$
2) $\delta(q, aw) = \delta(\delta(q,a), w) \quad \forall\ w \in \Sigma^*, a \in \Sigma$
3) $\delta(q, wa) = \delta(\delta(q,w), a) \quad \forall\ w \in \Sigma^*, a \in \Sigma$

Prove that for any transmission func^n δ and for any Input String x & y

To prove :

$$\delta(q, xy) = \delta(\delta(q,x), y)$$

⇒ We will prove this by PMI on length of y.

Let $|y| = 1$, $y = a$ (say)

$\delta(q, xy) = \delta(q, xa) = \delta(\delta(q, x), a)$

$= \delta(\delta(q, x), y)$

Hence the result is True for length 1

Suppose the result is True is for any string $|y| = n$

Now let

$|y| = n+1$, $y = y_1 a$, $|y_1| = n$

$\delta(q, xy) = \delta(q, \underbrace{x y_1}_{x_1} a) = \delta(q, x_1, a)$

$= \delta(\delta(q, x_1), a)$

$= \delta(\delta(q, xy_1), a)$

$= \delta(\delta(\delta(q, x), y_1), a)$

$= \delta(\delta(q, x), y, a)$

$= \delta(\delta(q, x), y) = RHS$ Proved

→ Prove that if $\delta(q, x) = \delta(q, y)$ then

$\delta(q, xz) = \delta(q, yz) \; \forall \; z \in \Sigma^*$

⇒ $\delta(q, xz) = \delta(\delta(q, x), z)$

$= \delta(\delta(q, y), z)$

$= \delta(q, yz)$

→ Defn Let $w \in \Sigma^*$

w is accepted by DFA iff. $\delta(q_0, w) = q$

where $q \in F$ → final state (Accepting State)

Q

State	Input 0	Input 1
→ q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
(*q_3)	q_1	q_2

Write DFA

$Q = \{q_0 - - q_3\}$

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

$f = \{q_3\}$

$\delta(q_0, 0) = q_2$

$\delta(q_0, 1) = q_1$

⋮

→ Give the entire sequence of states for the input 110101

$\Rightarrow \delta(q_0, 110101) = \delta(\delta(q_0, 1), 10101)$

$= \delta(q_1, 10101)$

$= \delta(\delta(q_1, 1), 0101) = \delta(q_0, 0101)$

$= \delta(\delta(q_0, 0), 101) = \delta(q_2, 101)$

$= \delta(\delta(q_2, 1), 01) = \delta(q_3, 01)$

$= \delta(\delta(q_3, 0), 1) = \delta(q_1, 1) = q_0$

Sequence $q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_0 \xrightarrow{0} q_2 \xrightarrow{1} q_3 \xrightarrow{0} q_1 \xrightarrow{1} q_0$

hence the sequence ($q_0 q_1 q_0 q_2 q_3 q_1 q_0$)

but this is not Accepted by Machine.

Non Deterministic finite Automaton (NDFA)

It is defined as a 5tuple $(Q, \Sigma, \delta, q_0, f)$ where where

→ Q is finite non Empty set of states.
→ Σ " " " input Symbol
→ $\delta : Q \times \Sigma \to 2^Q$
→ q_0 is the initial state
→ $f \subseteq Q$ is the set of final States

for point 3.

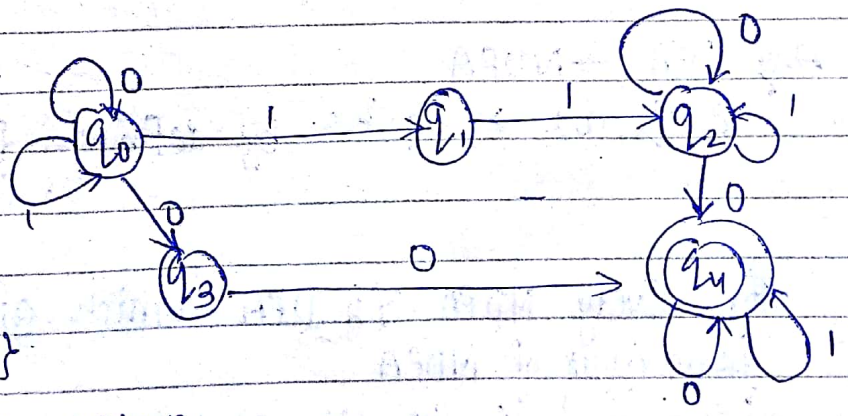
$\delta(q, a) = \{q_1, q_2 \dots q_n\}$

that's why it is non deterministic

Note: Every DFA is NDFA if we define:

$\delta(q, a) = \{q'\}$

Example



{NDFA}

W = 0100

$\delta(q_0, 0100) = \delta(\delta(q_0, 0), 100)$

$= \delta(q_0, 100)$

$= \delta(\delta(q_0, 1), 00) = \delta(q_0, 00)$

$= \delta(\delta(q_0, 0), 0) = \delta(q_3, 0) = q_4$

Seq. $q_0\ q_0\ q_0\ q_3\ q_4$

but $\delta(q_0, w) \subseteq Q$

Any String $w \in \Sigma^*$ is accepted by NDFA iff : $\delta(q_0, w)$ Contain Some final State

M = any finite automaton
(DFA/NDFA)

T(M) = Set of all those Strings which are accepted by Machine (Set accepted by M)

Date 8/Jan/18

* In DFA
$$\delta(q_0, w) = q' \in F$$
* but In NDFA
$$\delta(q_0, w) \subseteq Q$$
If Q Contains final State (any one)

Any DFA → NDFA
$\delta(q_0, a) \rightarrow \{q'\}$ → by defining $\{q'\}$ as Singleton Set

Theorm for Every NDFA ∃ a DFA which Simulates the behaviour of NDFA
i.e. if L is the Set accepted by NDFA then L is also accepted by DFA

Proof Suppose $M = (Q, \Sigma, \delta, q_0, F)$ is an NDFA and $L = T(M)$

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

1. $Q' = 2^Q$ any elt. of Q' is denoted by $[q_1 q_2 \dots q_j]$ if $q_1, q_2, \dots q_j \in Q$
2. $q_0' = [q_0]$
3. F' is the set of all those subsets of Q' which contains at least one final state of M.
4. Define δ' as follows

$$\delta(\{p_1 p_2 \dots p_j\}, a) = \{q_0, q_1 \dots q_n\}$$
iff $\delta'[[P_1 P_2 \dots P_j], a) = [q_1 q_2 \dots q_n]$

$$\delta(\{p_1, p_2 \dots p_j\}, a) = \delta(p_1, a) \cup \delta(p_2, a) \cup \delta(p_3, a) \dots \cup \delta(p_j, a).$$

To show that:
if $L = T(M)$ then $L = T(N')$.
, Before proving this we prove one auxiliary result.

$$\delta(q_0, x) = \{q_1 q_2 \dots q_j\} \text{ iff } \delta'(q_0', x) = [q_1 q_2 \dots q_j] \quad \forall x \in \Sigma^*$$

To show that
$$\delta(q_0, x) = \{q_1 q_2 \dots q_j\}.$$
$$\Rightarrow \delta'(q_0', x) = \{q_1 q_2 \dots q_j\}$$
Prove this by PMI on $|x|$ (length of x)
$\Rightarrow$ Let $|x| = 0$
$\delta(q_0, \wedge) = \{q_0\}$ $\quad \delta'(q_0', \wedge) = q_0' = [q_0]$
Result is true for $|x| = 0$

Suppose the result is true for any string of length $\leq m$

Let the $|x| = m+1$ then $x = ya$ where $|y| = m$, $a \in \Sigma$

Let $\delta(q_0, y) = \{p_1, p_2 \dots p_j\}$ & $\delta[q_i, ya] = \{r_1, r_2 \dots r_k\}$
i.e. $\delta(\delta(q_0, y), a) = \{r_1, r_2 \dots r_k\}$

$\Downarrow$ implies

$\delta'(q_0', y) = [p_1, p_2 \dots p_j]$ $\quad \delta(\{p_1 \dots p_j\}, a) = \{r_1, r_2 \dots r_k\}$

$\delta'([p_1 \dots p_j], a) = [r_1, r_2 \dots r_k]$

$\delta'(q_0', ya) = \delta'(\delta'(q_0', y)a)$

Proved $= \delta'([p_1 \dots p_j], a) = [r_1 \dots r_k]$

→ Similarly other part to be proved by yourself...

→ Now show that if $L = T(M)$ then $L = T(N')$

Suppose $w \in T(M) \Rightarrow \delta(q_0, w)$ reaches to some final state of M

Hence, $\delta'(q_0', w) \in F'$

Any string which is accepted by NDFA is also accepted by DFA.

Question

Construct a DFA equivalent to NDFA $M = (\{q_0, q_1\}, \{0,1\}, \delta, q_0, \{q_0\})$

Soln Let DFA $= (Q', \Sigma, \delta', q_0', F')$

$Q' = 2^Q = \{\phi, \{q_0\}, \{q_1\}, \{q_0 q_1\}\} \approx \{\phi, [q_0], [q_1], [q_0 q_1]\}$

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

$q_0' = [q_0]$

$F' = \{[q_0], [q_0 q_1]\}$ ← All those subset of Q which contains atleast one final state.

for δ' we make Transition Table {q' is defined by Table given below}

State	Input	
	0	1
$[q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1]$	$[q_0\, q_1]$
$[q_0\, q_1]$	$[q_0\, q_1]$	$[q_0\, q_1]$

$$\delta'([q_0], 0) = \delta(\{q_0\}, 0) = [q_0]$$

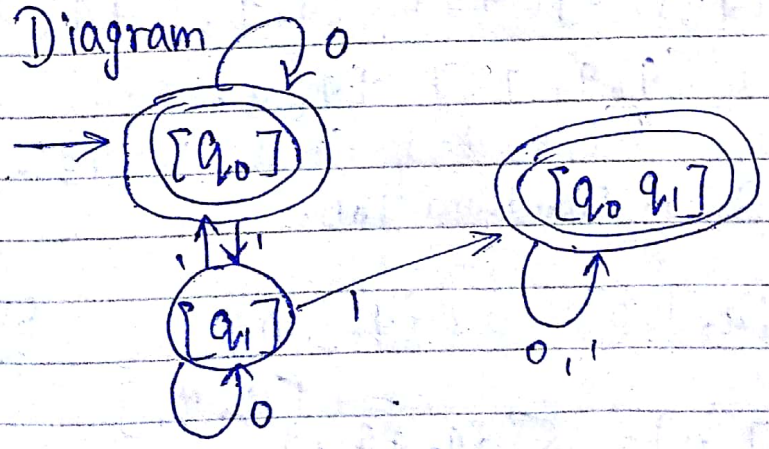
⋮

given

State/Σ	0	1
→ q_0	q_0	q_1
q_1	q_1	$q_0\, q_1$

$$\begin{aligned}\delta'([q_0\, q_1], 0) &= \delta'([q_0], 0) \cup \delta'([q_1], 0) \\ &= \delta(q_1, 0) \cup \delta(q_1, 0) \\ &= [q_0] \cup [q_1] \\ &= [q_0\, q_1]\end{aligned}$$

Diagram



Date 10/Jan/18

Example

NDFA =

$M = (\{q_0 q_1 q_2 q_3\}, \{0,1\}, \delta, q_0, \{q_3\})$

δ is defined by:

State/Σ	0	1
→ q_0	$q_0 q_1$	q_0
q_1	q_2	q_1
q_2	q_3	q_3
(*) q_3	-	q_2

Construct DFA ?

→ DFA

16 elements

$Q' = 2^Q = \{\phi, \{q_0\}, \{q_1\}, \{q_2\}, \{q_3\} \ldots\}$

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

$q_0' = \{q_0\}$

$F' = \{[q_3], [q_0 q_3], [q_1 q_3], [q_2 q_3], [q_0 q_1 q_3], [q_0 q_2 q_3], [q_1 q_2 q_3], [q_0 q_1 q_2 q_3]\}$,

Also find δ' → Transition Table.

$\delta'([q_0], 0) = \delta(\{q_0\}, 0) = [q_0 q_1]$

Similarly for others

$\delta'([q_0 q_1], 0) = \delta(\{q_0 q_1\}, 0)$

$= \delta(\{q_0\}, 0) \cup \delta(\{q_1\}, 0)$

$= \{q_0 q_1\} \cup \{q_2\} = [q_0 q_1 q_2]$

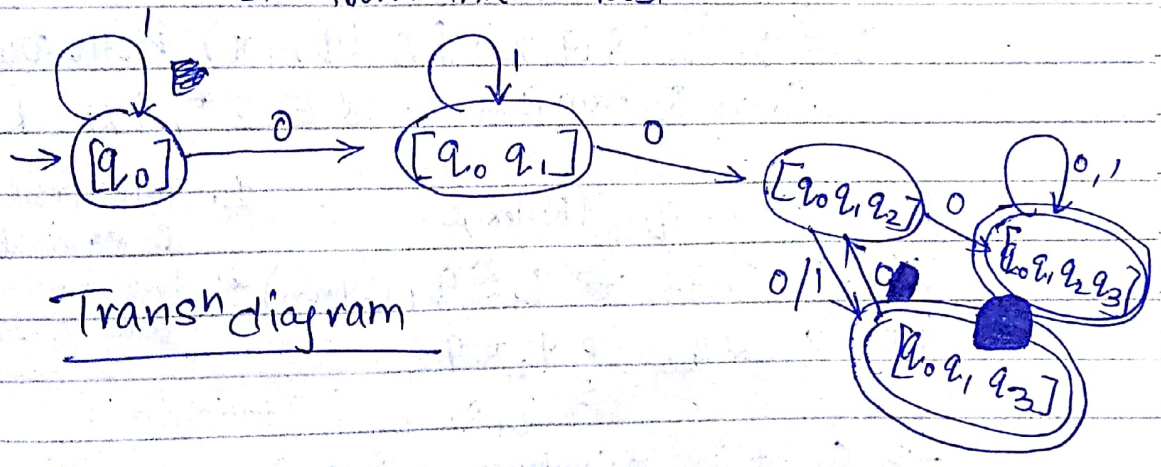
We know $\delta'([p_1 p_2 - - p_j], a) = [r_1 r_2 - - r_k]$

$\delta(\{p_1 p_2 - - p_j\}, a) = \{r_1 r_2 - - r_k\}$

δ' →

State / Input	0	1
$[q_0]$	$[q_0 q_1]$	$[q_0]$
$[q_0 q_1]$	$[q_0 q_1 q_2]$	$[q_0 q_1]$
$[q_0 q_1 q_2]$	$[q_0 q_1 q_2 q_3]$	$[q_0 q_1 q_3]$
$[q_0 q_1 q_2 q_3]$	$[q_0 q_1 q_2 q_3]$	$[q_0 q_1 q_2 q_3]$
$[q_0 q_1 q_3]$	$[q_0 q_1 q_2]$	$[q_0 q_1 q_2]$

Construct the transn table for those state which are are reachable from Initial table



Transn diagram

$\delta'([q_0 q_1 q_2], 0) = \delta(\{q_0 q_1 q_2\}, 0)$

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

$= \{q_0 q_1\} \cup \{q_2\} \cup \{q_3\}$

$= \{q_0 q_1 q_2 q_3\} = [q_0 q_1 q_2 q_3]$

Minimization of Automaton

Defn Two State q_1 & q_2 are said to be k equivalent ($k \geq 0$) iff
$\delta(q_1, x)$ & $\delta(q_2, x)$ both are final or non final states for all $\forall x \in \Sigma^*, |x| \leq k$

Defn If q_1 & q_2 are k equivalent for then they are said to be equivalent

Defn if q_1 & q_2 are $(k+1)$ equivalent then they are k equivalent.

Defn of k equivalence $Q = \{q_1, q_2 - - - q_n\}$
$q_1 \overset{k}{\approx} q_2$ if $\delta(q_1, x)$ & $\delta(q_2, x)$ both are final & non final $\forall x \in \Sigma^*, |x| \leq k$

1) $q_1 \overset{k}{\approx} q_1$ (Reflexive)
2) $q_1 \overset{k}{\approx} q_2 \Rightarrow q_2 \overset{k}{\approx} q_1$ (Symm)
3) $q_1 \approx q_2$ & $q_2 \approx q_3$
$\Rightarrow q_1 \approx q_3$ (Transitive)

1. Two final state are 0-equivalent
2. Two non final states are 0 equivalent.

→ So it is equivalence relation on Q

→ Π_k Set of k equivalence classes
$\Pi_k = \{ [\,][\,] - - \}$
Union → of them Π_k
∩ of them { } empty set

→ Set of all states of Q_i^k which are k-equivalent to each other

$\Pi_k = \{Q_1^k, Q_2^k - - - Q_i^k\}$

Date 11/Jan/18 → Notes from Keerthi

Date: 15/Jan/18

Moore to Mealy Machines

Present State	Next State		Output
	0	1	
$-q_0$	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

	Next State			
	$a=0$		$a=1$	
Present State	State	Output	State	Output
$-q_0$	q_3	0	q_1	0
q_1	q_1	1	q_2	1
q_2	q_2	0	q_3	0
q_3	q_3	0	q_0	0

Chapter 2 →

Formal Language

→ Any language which computer can understand is formal language

$S \rightarrow$ <noun> <Verb> <adverb> → description of Sentences

S → Start Symbol.

How the Sentence is described

{noun, verb, adverb} → Variable Set

{Ram, Shyam, ate, ran, walked, quickly, slowly} → Terminal Set

$S \rightarrow$ <noun> <verb> <adverb> → description of Set

S → <noun> <verb>.

Rules (Replaced left by right)

<noun> → Ram
<noun> → Shyam
<verb> → ate
<verb> → ran
<verb> → walked
<adverb> → quickly
<adverb> → slowly

Grammer

$G = (V_N, \Sigma, P, S)$ where :

1) V_N is a finite non empty Set of non terminals (variables)
2) Σ is a " " " " of terminals.
3) $V_N \cap \Sigma = \phi$
4) $S \in V_N$ is the Start Symbol.
5) P Contains element of the type $\alpha \rightarrow \beta$ where α & β are Strings on $(V_N \cup \Sigma)$ Such that α Contains at least one Symbol from V_N. P is known as the set of production rules.

Defn

→ Sentence is always a string of terminals.

→ Variables are represented by capital letters.

→ terminals " " " lower "

$S \to AB$

$S \to \langle noun \rangle \langle verb \rangle$

but

$AB \to S$ wrong

$S \to a$ ✓

$A \to a$ ✓

$a \to A$ wrong

→ if $\alpha \to \beta$ is in P

then $\gamma, \delta, \in (V \cup \Sigma)^*$

$\gamma \alpha \delta$ derives $\gamma \beta \delta$

$\gamma \alpha \delta \Rightarrow \gamma \beta \delta$ first step derivation.

$\langle noun \rangle \langle verb \rangle \langle adverb \rangle \Rightarrow \langle noun \rangle$ ate $\langle adverb \rangle$ (first step)

$\Rightarrow$ Ram ate $\langle adverb \rangle \Rightarrow$ Ram ate quickly (string on $(V_N \cup \Sigma)$)

No production Left to Replace all are terminals it is Sentences.

In general:

$\langle noun \rangle \langle verb \rangle \langle adverb \rangle \overset{*}{\underset{G}{\Rightarrow}}$ Ram ate quickly

(not directly reachable)

$\alpha_1 \underset{G}{\Rightarrow} \alpha_2 \Rightarrow \alpha_3 \cdots \underset{G}{\Rightarrow} \alpha_n$

(directly Reachable)

$\alpha_1 \overset{*}{\underset{G}{\Rightarrow}} \alpha_n$

Date 17/Jan/18

Now Suppose Our Grammer

$$G = (\{S\}, \{0,1\}, \{S \to 0S1, S \to 01\}, S)$$

$V_N = \{S\}$

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

$P = \{S \to 0S1, S \to 01\} \approx \{S \to 0S1/01\}$

S is Start Symbol.

$\Rightarrow$

$S \to 0S1 \xrightarrow{S \to 01} 0011 \to 0^2 1^2$

first Step derivation

but $S \overset{*}{\underset{G}{\Rightarrow}} 0^2 1^2$

$S \to 0S1 \xrightarrow{S \to 0S1} 0^2 S 1^2 \xrightarrow{S \to 0S1} 0^3 S 1^3 \to - - -$

$- - - \to 0^{n-1} S 1^{n-1} \xrightarrow{S \to 01} 0^n 1^n$

$S \overset{*}{\underset{G}{\Rightarrow}} 0^2 S 1^2$

$S \overset{*}{\underset{G}{\Rightarrow}} 0^3 S 1^3$

In Generally

$S \overset{*}{\underset{G}{\Rightarrow}} 0^{n-1} S 1^{n-1}$

also $S \overset{*}{\underset{G}{\Rightarrow}} 0^n 1^n$

Language generated by the grammer

Suppose G is any grammer

→ language Generated by Grammer → $L(G)$
or Set of String terminals such that Sentence is derivable from Symbol. Start

$$L(G) = \{ w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w \}$$

↓ Sentential-form

⇒ 2 Grammers G_1 & G_2 are said to be equivalent iff
$L(G_1) = L(G_2)$
$G_1 \approx G_2$ iff $L(G_1) = L(G_2)$

Example

$G = \{ \{S\}, \{0,1\}, \{ S \to 0S1 , S \to \wedge \}, S \}$

find $L(G)$?

Sol^n $L(G) = \{ w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w \}$

$S \to \wedge$ i.e $\wedge \in L(G)$

$S \to 0S1 \xrightarrow{S \to \wedge} 01$, i.e $01 \in L(G)$

$S \to 0S1 \xrightarrow{S \to 0S1} 0^2S1^2$, i.e $0^2 1^2 \in L(G)$

⋮

$S \to 0S1 \xrightarrow{S \to 0S1} 0^2S1^2 \xrightarrow{S \to 0S1} 0^3S1^3 \cdots\cdots 0^nS1^n \xrightarrow{S \to \wedge} 0^n1^n$

hence $0^n1^n \in L(G)$ $\forall$ $n \geq 1$

i.e $\{0^n1^n, n \geq 1\} \subseteq L(G)$

↓ ← for empty String

$\{0^n1^n, n \geq 0\} \subseteq L(G)$

To Show $L(G) \subseteq \{0^n1^n, n \geq 0\}$

→ We need to show $L(G)$ is subset of $\{0^n 1^n, n \geq 0\}$

Let $w \in L(G)$ i.e $S \overset{*}{\Rightarrow}_G w$

$S \to \wedge$, $w = \wedge$

$S \to 0S1 \xrightarrow{S \to \wedge} 01$, $w = 01$

hence

$\longrightarrow w = 0^n 1^n$

hence $L(G) \subseteq \{0^n 1^n, n \geq 0\}$

since all $w \in L(w) \in \{0^n 1^n, n \geq 0\}$

hence

$L(G) = \{0^n 1^n, n \geq 0\}$

Proved

Ex $G = (\{S\}, \{a\}, \{S \to SS\}, S)$

$L(G) = ?$

$S \to SS \xrightarrow{S \to SS} S^2 S^2$ - - - No Terminal

hence

$L(G) = \phi$

Ex Suppose our product changes

We have Grammer $S \to aS/bS/a/b$ find $L(G) = ?$

Soln

$S \to a$ → Terminal , i.e $a \in L(G)$

$S \to b$ i.e $b \in L(G)$

$S \to aS \xrightarrow{S \to a} a^2$

$S \to aS \xrightarrow{S \to b} ab$

$S \to aS \xrightarrow{S \to bS} abS$

$S \to aS \xrightarrow{S \to aS} a^2 S$

$L(G) = \{a,b\}^+$ Nonempty

Q. find the Grammer G if

$L(G) = \{b, bb, bbb\}$

→ $S \rightarrow b/bb/bbb$

$G = (\{S\}, \{b\}, \{S \rightarrow b/bb/bbb\}, S)$

Q. find G if non empty
$L(G) =$ {any String on Set $\{0,1\}$}

$S \rightarrow 0$
$S \rightarrow 1$
$S \rightarrow 0S/1S$

Q. Suppose L — set of String over $\{a,b\}$ Containing aa Somewhere

⇒ [redacted]

$S \rightarrow AaaA$
$A \rightarrow \Lambda/aS/bS$

$S \rightarrow aa/aS/bS/Sb$

22/Jan/18

$$L(G) = \{ w \in \Sigma^* : S \overset{*}{\underset{G}{\Rightarrow}} w \}$$

Chomsky Classification of language

All the language are divided in to :

(i) Type 0
(ii) Type 1
(iii) Type 2
(iv) Type 3

All the productions are of type 0
A grammer in which all the productions are of type 0 is called type 0 grammer
Phase Structure grammer
A language generated by a type 0 grammer is called type 0 language

$\phi A \psi \Rightarrow \phi \alpha \psi$

ϕ : Left context, ψ : Right context

Ex. $abA\,bcd \longrightarrow ab\,ABbcd$

$\phi = ab$, $\psi = bcd$
$\alpha = AB$

$AC \longrightarrow A$

l.c $\rightarrow A$, r.c $\rightarrow \Lambda$
$a = \Lambda$

→ A product of the type $\phi A \psi \rightarrow \phi \alpha \psi$ if $\alpha \neq \Lambda$ is called type 1 production

→ if all the production in a grammer is of type 1 then it is called type 1 grammer & language generated by type 1 grammer is type 1 language. It is also called context sensitive grammer , C.S. language
→ C & L
→ CS G

Type 2 grammer

A production is of form $A \rightarrow \alpha$, where $A \in V_n$ & $\alpha \in (V_N \cup \Sigma)^*$ is called a type 2 production. if all the productions are of type 2 then the grammer is called type 2 grammer
It is also called context free grammer CFG , language generated by CFG is called context free lang.

Type 3 grammer

A grammer in which all the productions are of type $A \rightarrow a$ or $A \rightarrow aB$, where $A, B \in V_N$ & $a \in \Sigma$ is called called a type 3 grammer, also known an regular grammer, language generated by RG is Regular language

Notations:

$\mathcal{L}_0$ → Set of all type 0 language
$\{L : L \text{ is of type } 0\}$

$\mathcal{L}_{CSL}$ → Set of Context Sensitive Lang.

$\mathcal{L}_{CFL}$ → " " " free ",

$\mathcal{L}_{RL}$ → " " " Regular ".

Relationships between above grammer

$\mathcal{L}_{CSL} \subseteq \mathcal{L}_0$	$\mathcal{L}_{CFL} \subseteq \mathcal{L}_{CSL}$
$\mathcal{L}_{CFL} \subseteq \mathcal{L}_0$	$\mathcal{L}_{RL} \subseteq \mathcal{L}_{CFL}$
$\mathcal{L}_{RL} \subseteq \mathcal{L}_0$	

$\therefore \mathcal{L}_{RL} \subseteq \mathcal{L}_{CFL} \subseteq \mathcal{L}_{CSL} \subseteq \mathcal{L}_0$

Result:-
If G any grammer, $\alpha \to \beta$ ← (Next Chapter)
We can generate equivalent grammer G_1, $\alpha' \to \beta'$
Such that terminals can be replaced by Variable

$Aab \to \alpha\alpha\alpha$
$C_n \to a$
$AC_n b \to \alpha\alpha$ ..

Theorm $\mathcal{L}_i$ for $\forall i = 0(1)3 \simeq 0,1,2,3$ is closed under union

Proof

Let $L_1, L_2 \in L_i$

$G_1 = (V_N', \Sigma_1, P_1, S_1)$

& $G_2 = (V_N'', \Sigma_2, P_2, S_2)$

S.t. $L_1 = L(G_1)$ & $L_2 = L(G_2)$

$G_U = (V_N' \cup V_N'' \cup \{S\}, \Sigma_1 \cup \Sigma_2, P, S)$

Where $S \notin V_N' \cup V_N''$

$P = P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}$

→ To show that $L(G_U) = L_1 \cup L_2$

let $w \in L_1 \cup L_2$ w is any string in $L_1 \cup L_2$.

i.e $w \in L_1$ OR $w \in L_2$ i.e $S_1 \overset{*}{\underset{G_1}{\Rightarrow}} w$ or $S_2 \overset{*}{\underset{G_2}{\Rightarrow}} w$

⇒

But

$S \underset{G_U}{\to} S_1 \overset{*}{\underset{G_1}{\Rightarrow}} w$ i.e. $S \overset{*}{\underset{G_U}{\Rightarrow}} w$ or

$S \underset{G_U}{\to} S_2 \overset{*}{\underset{G_1}{\Rightarrow}} w$ i.e $S \overset{*}{\underset{G_U}{\Rightarrow}} w$

i.e $w \in L(G_U)$ mean $L_1 \cup L_2 \subseteq L(G_U)$

Other way Round:

To Show Now [redacted] $L(G_U) \subseteq L_1 \cup L_2$

Suppose $w \in L(G_U)$

i.e $S \overset{*}{\underset{G_U}{\Rightarrow}} w$ i.e

$S \to S_1 \dots \to w$ or $S \to S_2 \dots \to w$

i.e

$S_1 \overset{*}{\underset{G_1}{\Rightarrow}} w$ or $S_2 \overset{*}{\underset{G_2}{\Rightarrow}} w$

$w \in L_1$ or $w \in L_2$

$w \in L_1 \cup L_2$

i.e

$\therefore L(G_0) \subseteq L_1 \cup L_2$

Hence

$L(G_0) = L_1 \cup L_2$ proved by both parts

date 24/Jan/18

Theorm $\mathcal{L}_i, \forall i = 0(1)3$ is Closed under Concatenation

Proof $L_1, L_2 \in \mathcal{L}_i$ Then $\exists\ G_1 = (V_N', \Sigma_1, P_1, S_1)$ & $G_2 = (V_N'', \Sigma_2, P_2, S_2)$ such that

$L_1 = L(G_1)$ & $L_2 = L(G_2)$

Construct $G_{con} = (V_N' \cup V_N'' \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_{con}, S)$

Where $S \notin V_N' \cup V_N''$

and $P_{con} = P_1 \cup P_2 \cup \{S \to S_1 S_2\}$

To Show that

$L(G_{con}) = L_1 L_2$

Let $w \in L_1 L_2$ i.e $w = w_1 w_2$ where $w_1 \in L_1, w_2 \in L_2$

$w_1 \in L_1$ means

$$S_1 \overset{*}{\underset{G_1}{\Rightarrow}} w_1$$

and $w_2 \in L_2$ means

$$S_2 \overset{*}{\underset{G_2}{\Rightarrow}} w_2$$

Basically we want to show

$$w \in L(G_{con})$$

i.e $$S \overset{*}{\underset{G_{con}}{\Rightarrow}} w$$

$$S \underset{G_{con}}{\rightarrow} S_1 S_2 \overset{*}{\underset{G_{con}}{\rightarrow}} w_1 S_2 \overset{*}{\underset{G_{con}}{\rightarrow}} w_1 w_2$$

i.e $$S \overset{*}{\underset{G_{con}}{\Rightarrow}} w_1 w_2 = w$$

i.e $w \in L(G_{con})$

Hence $L_1 L_2 \subseteq L(G_{con})$

Converse part

T.S.T. $L(G_{con}) \subseteq L_1 L_2$

Let $w \in L(G_{con})$

i.e $$S \overset{*}{\underset{G_{con}}{\Rightarrow}} w$$

but first step derivation is

$$S \underset{G_{con}}{\rightarrow} S_1 S_2 \overset{*}{\underset{G_{con}}{\rightarrow}} w_1 S_2 \overset{*}{\underset{G_{con}}{\rightarrow}} w_1 w_2$$

i.e $$S \overset{*}{\underset{G_{con}}{\Rightarrow}} w_1 w_2$$

i.e $w = w_1 w_2$

where $w_1 \in L_1$ & $w_2 \in L_2$

$\therefore w \in L_1 L_2$

$L(G_{con}) \subseteq L_1 L_2$

Hence $L_1 L_2 = L(G_{con})$ Proved

Now Suppose

$$A = \{u \mid u \in \Sigma^*\}$$
$$A^T = \{u^T \mid u \in A\}$$

Theorm

$\mathcal{L}_i \; \forall \; i = 0(1)3$ is closed Under Transpose operation.

Proof

Let $L \in \mathcal{L}_i$ Suppose the corresponding grammer

$$G = (V_N, \Sigma, P, S) \text{ s.t } L = L(G)$$

$$G^T = (V_N, \Sigma, P^T, S) \text{ where}$$
$$P^T = \{\alpha^T \to \beta^T \text{ s.t. } \mid \alpha \to \beta \text{ is in } P\}$$

To Show that

$$L^T = L(G^T)$$

Suppose $w \in L^T$ i.e ~~$S \overset{*}{\Rightarrow}_G w$~~ means $w^T \in L$

i.e $S \overset{*}{\Rightarrow}_G w^T$ i.e $S \overset{*}{\Rightarrow}_{G^T} w$

mean

$$w \in L(G^T)$$

→ Prove the 2nd part yourself

~~Algorithm~~ to test whether $w \in L(G)$ or not

Steps 1) $W_0 = \{S\}$ $\quad |w| = n.$

Step 2.2) $W_{i+1} = W_i \cup \{\beta \in (V_N \cup \Sigma)^* : \exists\, \alpha \in W_i,$
$\alpha \Rightarrow \beta \;\&\; |\beta| \leq n\}$

$W_i \subseteq W_{i+1} \quad \forall i \geq 0$

Terminate the Construction where $W_{k+1} = W_k$ for the first time

Now if $w \in W_k$ then $w \in L(G)$ o.w. $w \notin L(G)$

Example

Consider grammer G given by

$S \to 0SA_12$, $S \to 012$

, $2A_1 \to A_12$, $1A_1 \to 11$

→ Test whether 00112 $\in L(G)$ & 001122 $\in L(G)$ or not ?

Sol Case 1)

$w = 00112$

$|w| = 5$

Construct $W_0 = \{S\}$

$W_1 = W_0 \cup \{012, 0SA_12\}$

$= \{012, 0SA_12, S\}$

$W_2 = \{012, S, 0SA_12\} = W_1$

Since $0SA_12 \Rightarrow 00SA_12A_12$ X

↓ ↘ length > 5

$0012A_12$ X

$00112 \notin W_2 \Rightarrow 00112 \notin L(G)$

(ii) 001122

$|w| = 6$

Construct $W_0 = \{S\}$

$W_1 = \{S, 012, 0SA, 2\}$

$W_2 = \{S, 012, 0SA, 2\} \cup \{0012A, 2\}$

$0SA, 2 \rightarrow 0012A, 2$

$W_3 = \{S, 012, 0SA, 2, 0012A, 2\} \cup \{00A, 22\}$

$W_4 = \{S, 012, 0SA, 2, 0012A, 2, 00A, 22\} \cup \{001122\}$

$\therefore W_5 = W_4$ Stop

$w = 001122 \in W_4 \Rightarrow 001122 \in L(G)$

25/Jan/18

Regular Expressions

→ Recursive definition for regular Expression over Σ

1) Any Terminal Symbol (ie $a \in \Sigma$) and $\wedge$ are reg. Exp.
2) The union of 2 regular Expression R_1 & R_2 written $R_1 + R_2$ is also r.e.
3) The concatanation of 2 r.e. R_1 & R_2, written $R_1 R_2$ is also a r.e
4) The iteration of a r.e. R denoted by R^* is a r.e.
5) if R is a r.e. then (R) is also a r.e.
6) Any Expression over Σ obtained on application of the above 1-5 rules are also r.e.

Iteration → Concatenation → Union

Defn Any Set represented by a r.e. is called a regular Set.

for Example {a} r.e. a

r.e. → a+b = {a,b}

ab = {ab}

a* = {^, a, aa, aaa, a---}

(a+b)* = {a,b}*

{101} → 101 ⇐ r.e

{abba} = abba

{01, 10} = 01 + 10

{^, ab} = ^ + ab

{^, 0, 00, 000, ---} = 0*

{1, 11, 111 ---} = 1(1)*

L = Set of all Strings of 0's & 1's ending in 00

{0,1}* → (0+1)*00

L = Set of " " " beginning with 0 & ending with 1 → 0(0+1)*1

L = {^, 11, 1111, 111111 ---} ⇒ (11)*

Defn

Two r.e P & Q are equivalent ($P \equiv Q$) if $P = Q$ represent the same set of string

Identities for R.e

1. $\wedge R = R \wedge = R$
2. $\wedge^* = \wedge$
3. $R + R = R$
4. $R^* R^* = R^*$
5. $RR^* = R^* R$
6. $(R^*)^* = R^*$
7. $\wedge + RR^* = R^* = \wedge + R^* R$
8. $(PQ)^* P = P(QP)^*$
9. $(P+Q)^* = (P^* Q^*)^* = (P^* + Q^*)^*$
10. $(P+Q)R = PR + QR$
11. $R(P+Q) = RP + RQ$

Arden's Theorm

→ Let P & Q be two r.e over Σ if P does not contain $\wedge$ then the eqn in R, namely

$$R = Q + RP \quad \text{---} (1)$$

has a unique solution given by $R = QP^*$

Proof :-

$$Q + RP = Q + QP^*P = Q(\wedge + P^*P)$$

$$= QP^* \quad (\text{by } 7)$$

$$\hookrightarrow = R$$

$$= \text{RHS}$$

$\therefore R = QP^*$ is a soln of eq (1)

For Uniqness

$R = Q + RP$

$= Q + (Q+RP)P$

$= Q + QP + RP^2$

$= Q + QP + QP^2 + RP^3$

$\vdots$

$Q + QP + QP^2 + QP^3 - - - + QP^i + RP^{i+1}$

$R = Q(\wedge + P + P^2 + P^3 - - - P^i) + RP^{i+1}$

$=$

let $w \in R$, $|w| = i$

$w \in Q(\wedge + P + P^2 - - - + P^i) + RP^{i+1}$

$w \notin RP^{i+1}$ ($\because |w| = i$ & P does not contain $\wedge$)

$\therefore w \in Q(\wedge + P + P^2 + - - - + P^i)$

$\therefore w \in QP^*$

Conversely

Let $w \in QP^*$

$\therefore w \in QP^k$ for some k

$w \in Q(\wedge + P + - - - + P^i) + RP^{i+1}$

So $w \in R$

that mean, unique

$R = QP^*$ is a Solution

Example Prove that

$(1+00^*1) + (1+00^*1)(1+0^*1)^* (0+10^*1) = 0^*1(0+10^*1)^*$

Sol $(1+00^*1)(\wedge + \underbrace{(0+10^*1)^*}_{P^*}\underbrace{(0+10^*1)}_{P})$

$(1+00^*1)(0+10^*1)^*$

$(\wedge + 00^*1)(0+10^*1)^* = 0^*1(0+10^*1)^* = RHS$

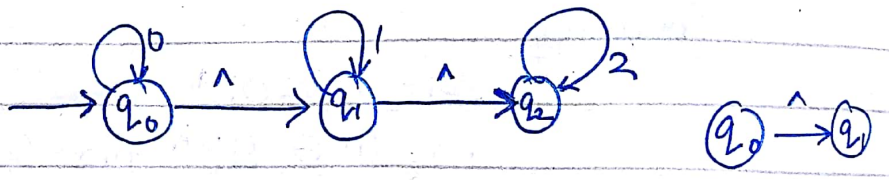
Date 29/Jan/18

→ Transition system containing ∧-moves

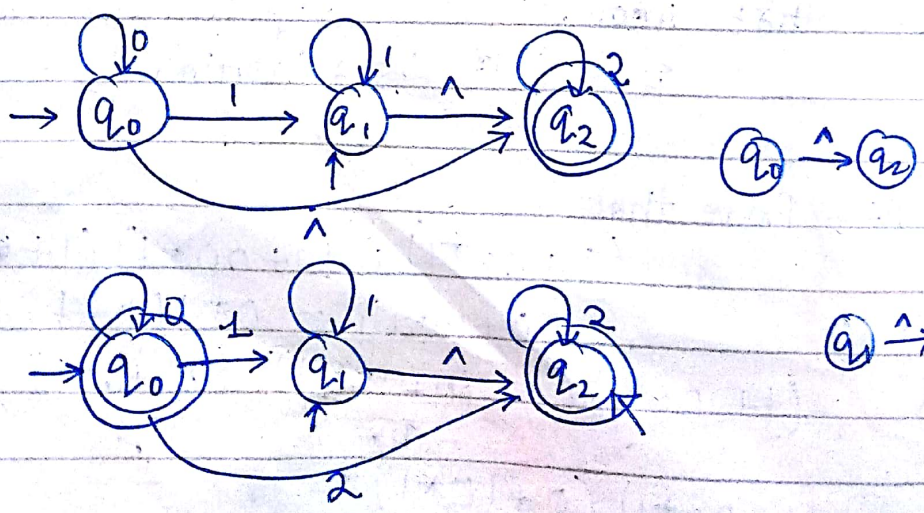
Suppose you want to replace a ∧-move from V_1 to another vertex V_2
we proceed as follows :

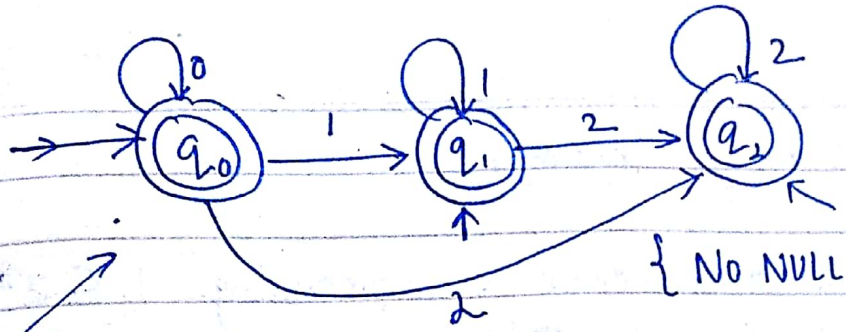
Step1 :- find all the edges starting from V_2
Step2 :- duplicate all these edges starting from V_1 without changing the edge label
Step3 :- if V_1 is the initial state, make V_2 also Initial state.
Step4 :- if V_2 is a final state, make V_1 also the as the final state.

For Example



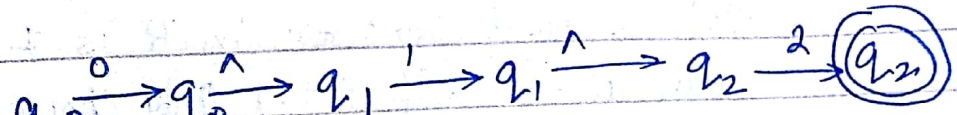
Suppose we want to replace ∧ move from q_0 to q_2





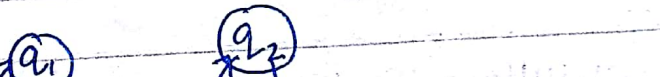
{ No NULL moves left }

Check $w = 012$ is accepted by system with null moves? from given diagram



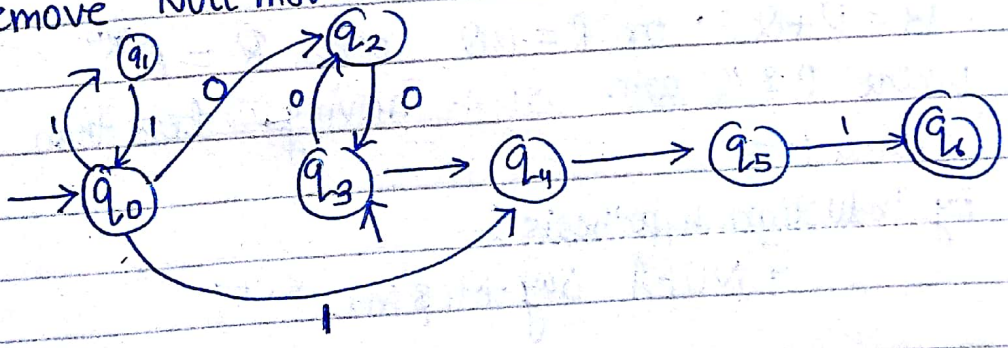
$0 \wedge 1 \wedge 2 \equiv 012$

→ Similarly $w = 012$ is accepted by it also!!!



for Example

→ Remove NULL move.



★ Kleene's Theorem

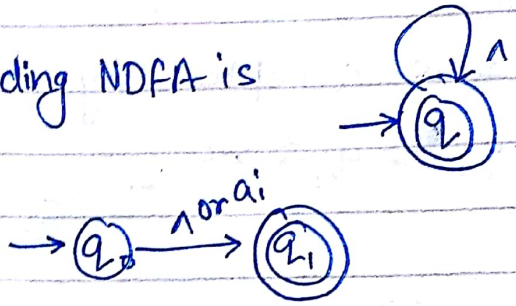
if R is a r.e. over Σ representing $L \subseteq \Sigma^*$ then
$\exists$ NDFA M with ■ $\wedge$-moves St. $L = T(M)$.

Proof by PMI on the no. of Character in R

basis Step

Suppose no. of character in R is 1
Then $R = \wedge$ (if $R = 1$) Only Choice
or $R = a_i$ where $a_i \in \Sigma$

→ Corresponding NDFA is



Induction Step :-

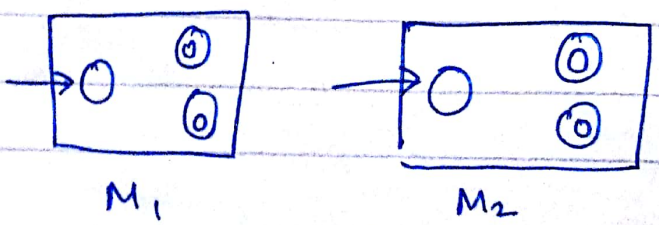
Suppose the result is True for any R with no. of Character $\leq n$

→ Now Suppose the no. of character in R is $(n+1)$

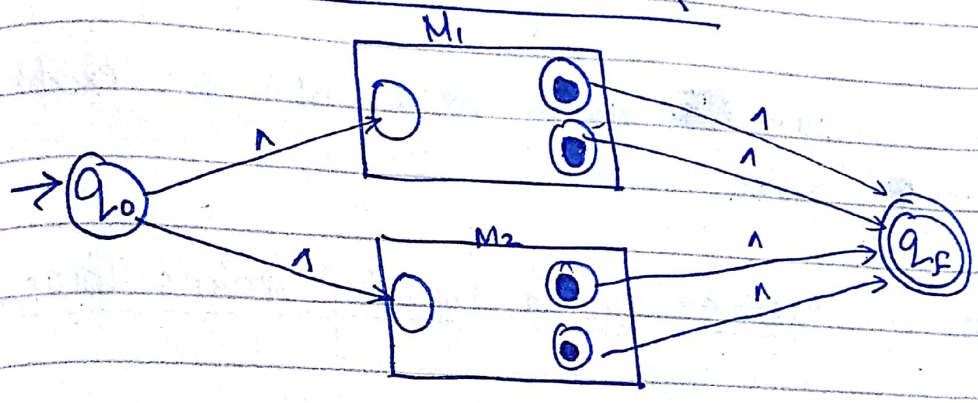
$R = P+Q$ or $R = PQ$ or $R = P^*$
where P & Q are r.e. having less than n Characters

By Induction hypothesis :-
$\exists$ NDFA Say M_1 & M_2 S.t

$L(P) = T(M_1)$ & $L(Q) = T(M_2)$

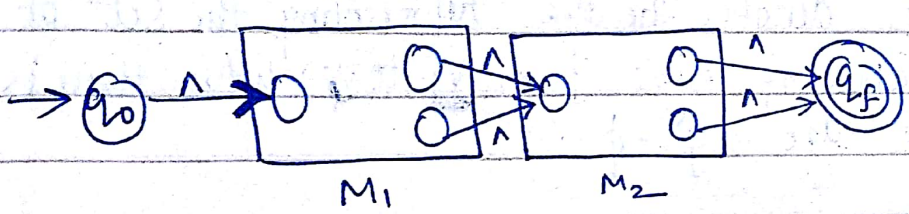


Case 1. When $R = P + Q$

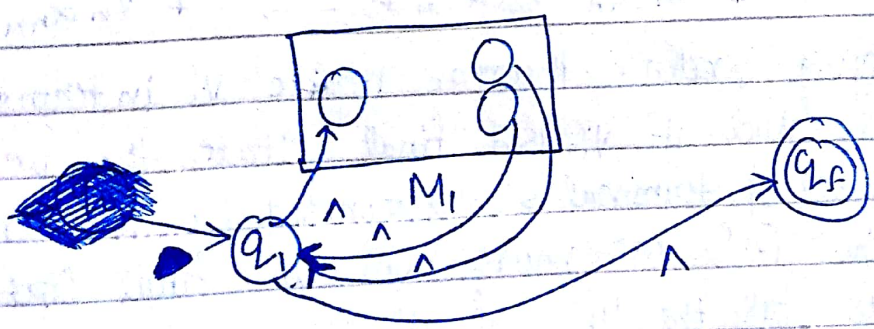


$L(R) = T(M)$

Case 2: When $R = PQ$



Case 3: when $R = P^*$



Hence we proved Kleene's Theorem

Date 31/Jan/18

→ Given a Transition System we can find a r.e excepted by given T.System

How??

We use Algebraic System by using Arden's Theorem.

Following assumption are there.

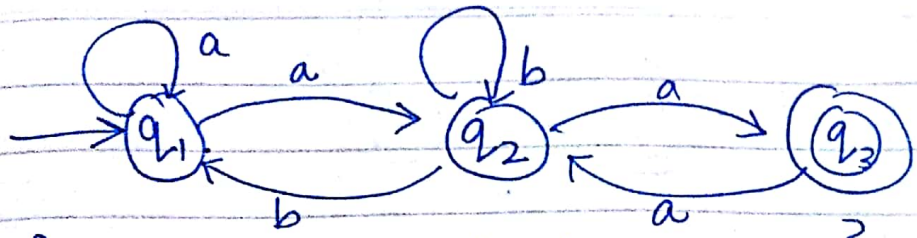
1) Transition graph does not have any null moves
2) It has only One Initial State, say V_1
3) Its Vertices are $V_1 V_2 - - - V_n$
4) α_{ij} denotes the r.e representing the set of labels of edges from V_i to V_j when there is no edge $\alpha_{ij} = \phi$

$$V_1 = V_1\alpha_{11} + V_2\alpha_{21} + \cdots + V_n\alpha_{n1} + \Lambda$$
$$V_2 = V_1\alpha_{12} + V_2\alpha_{22} + \cdots + V_n\alpha_{n2}$$
$$\vdots$$
$$V_n = V_1\alpha_{1n} + V_2\alpha_{2n} + \cdots + V_n\alpha_{nn}$$

By Applying Arden's theorem write V_i in terms of α_{ij} and if V_i is a final State then we get a string of terminals recognized by the Transn System if the T.S. has more than One final States then take the Union of all these.

Example

→



→ find the string recognized by system ?

Sol first 3 assumption holds.

4th assumpⁿ

↳

$q_1 = q_1 a + q_2 b + \wedge$

$q_2 = q_1 a + q_2 b + q_3 a$

$q_3 = q_2 a$

↓

$q_2 = q_1 a + q_2 b + q_2 aa$

$q_2 = q_1 a + q_2 (b + aa)$

$R \quad Q \quad R \quad P$

(Arden Theorm: $R = Q + RP$ then $R = QP^*$)

$q_2 = q_1 a (b + aa)^*$

$q_1 = q_1 a + q_1 a (b+aa)^* b + \wedge$

$q_1 = q_1 (a + a(b+aa)^* b) + \wedge$

$R \quad R \quad P \quad Q$

$= \wedge (a + a(b+aa)^* b)^*$

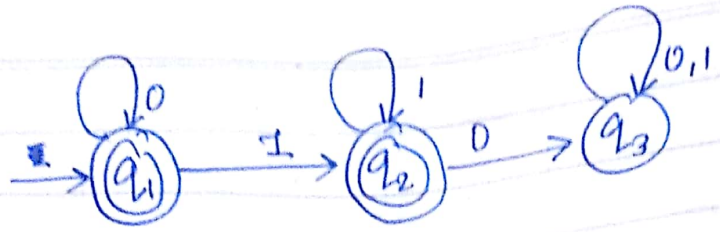
$q_1 = (a + a(b+aa)^* b)^*$

Ans

Now. $q_2 = (a + a(b+aa)^* b)^* a (b+aa)^*$

Now $q_3 = (a + a(b+aa)^* b)^* a (b+aa)^* a$

Q



4th Assump.

$q_1 = q_1 0 + \wedge$

$q_2 = q_1 1 + q_2 1$

$q_3 = q_2 0 + q_3 (0+1)$

Now.

~~$q_2 = q_1 01 + q_2 1$~~

[$R = Q + R$
$R = QP^*$

$q_1 = q_1 0 + \wedge$

$R \quad R\,P \quad Q$

$q_1 = \wedge 0^* = 0^*$ [By Arden's Theorem] — ①

$q_2 = 0^* 1 + q_2 1$

$R \quad Q \quad R\,P$

$q_2 = 0^* 1\,1^*$ — ②

$q_3 = 0^* 1\,1^* 0 + q_3 (0+1)$

$R \quad Q \quad R \quad P$

$q_3 = 0^* 1\,1^* 0\,(0+1)^*$ — ③

X Since q_3 is not final state

Take the union of q_1 & q_2 Since both are final.

$\$$ not being used in q_1 & q_2.

$q_1 + q_2 = 0^* + 0^* 1\,1^*$

$= 0^* (\wedge + 1\,1^*)$ [$\wedge + PP^* = P^*$]

$= 0^* 1^*$

Construction DFA equivalent R.E.

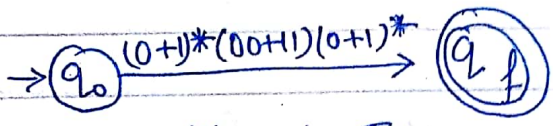
Step 1: Construct a transition graph equivalent to the given r.e using Kleene's Theorem.

Step 2: Construct the transition table for the system obtained in step 1. Construct the equivalent DFA.

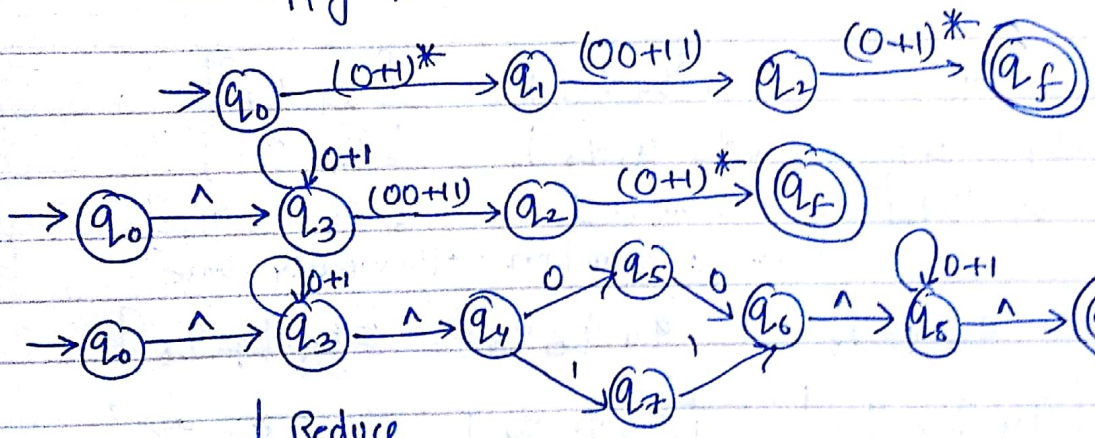
Suppose string is :—

$$(0+1)^*\ (00+11)\ (0+1)^*$$

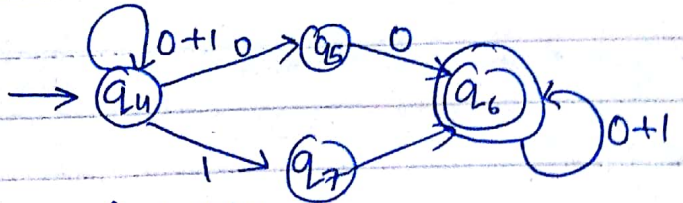
Step 1 :-



Now apply Kleene's Theorm.



↓ Reduce States



Now we find DFA

State/Σ	0	1
q_4	q_4, q_5	q_4, q_7
q_5	q_6	—
q_6	q_6	q_6
q_7	—	q_6

→ But it is still NDFA

Now to construct DFA

Q → NDFA

2Q → DFA

$Q = \{q_4, q_5, q_6, q_7\}$

2Q

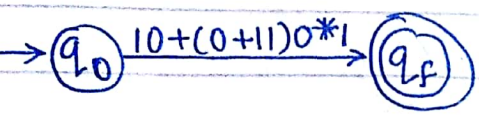
State/Σ	0	1
$[q_4]$	$[q_4 q_5]$	$[q_4 q_7]$
$[q_4 q_5]$	$[q_4 q_5 q_6]$	$[q_4 q_7]$
$[q_4 q_7]$	$[q_4 q_5]$	$[q_4 q_5 q_6]$
$[q_4 q_5 q_6]$	$[q_4 q_5 q_6]$	$[q_4 q_7 q_6]$
$[q_4 q_7 q_6]$	$[q_4 q_5 q_6]$	$[q_4 q_7 q_6]$

Use δ function. → Previously done.

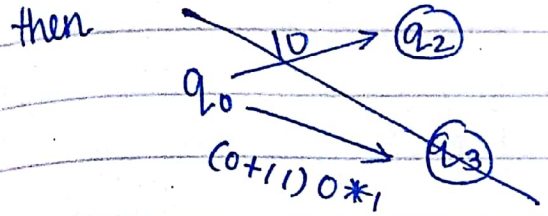
Date : 5/Feb/18

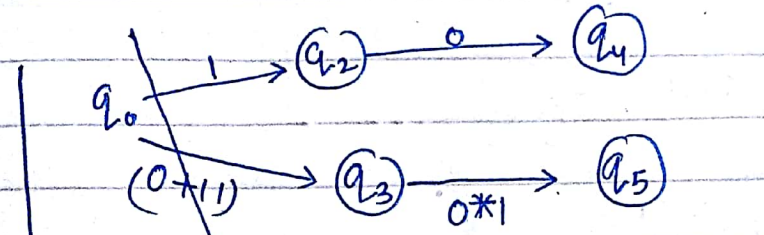
Example : Construct DFA equivalent to the r.e.
$10+(0+11)0^*1$

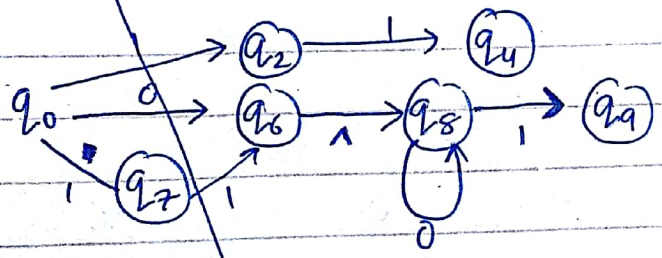
Step 1

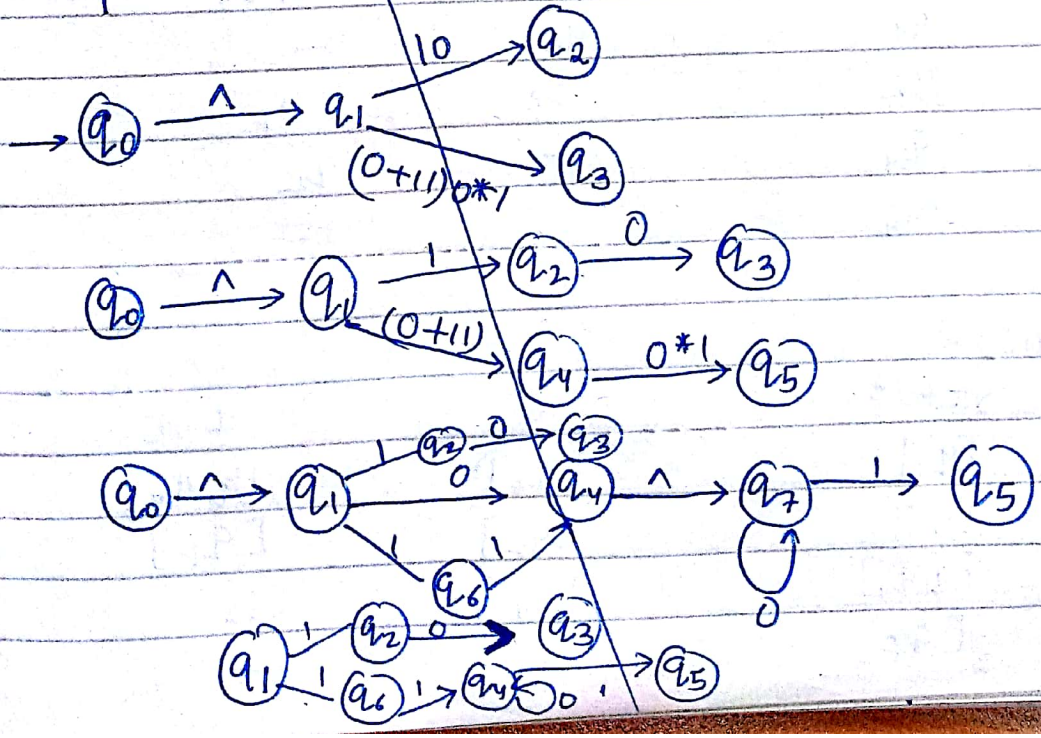


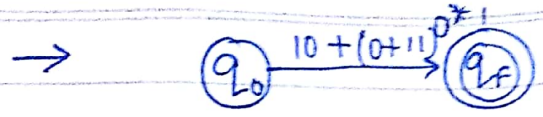
then

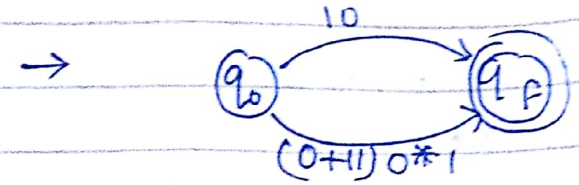


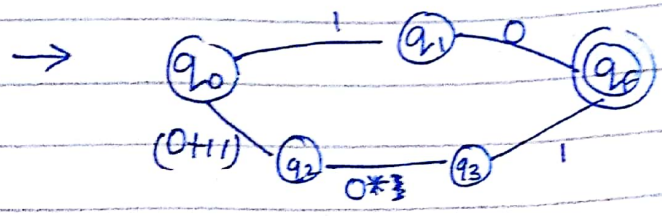


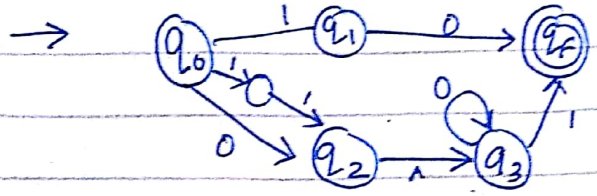


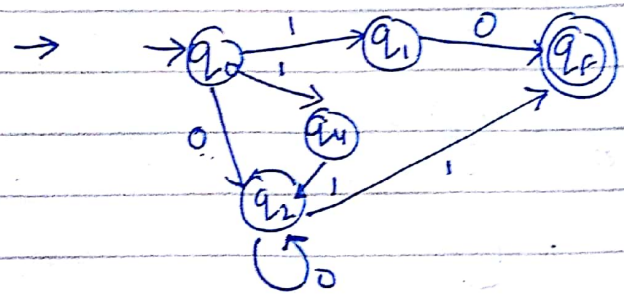












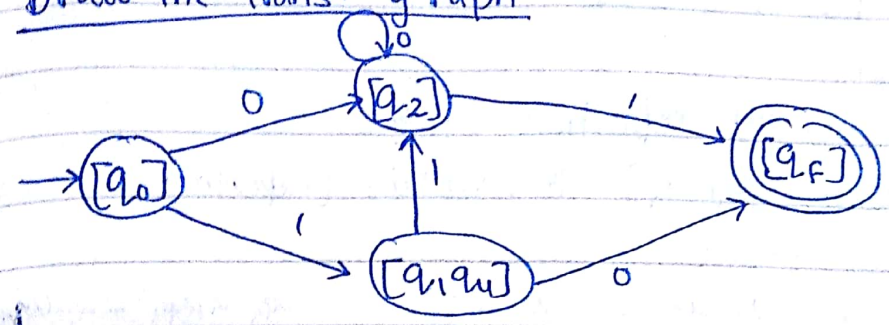
NDFA

State/Σ	0	1
q_0	q_2	q_1, q_4
q_1	q_f	—
q_2	q_2	q_f
q_4	—	q_2
q_f	—	—

DFA

State/Σ	0	1
$[q_0]$	$[q_2]$	$[q_1 q_4]$
$[q_2]$	$[q_2]$	$[q_f]$
$[q_1 q_4]$	$[q_f]$	$[q_2]$
$[q_f]$	—	—

Draw the Transn graph



Imp

Theorem: Pumping Lemma for regular Set

Statement:

Let $M = (Q, \Sigma, \delta, q_0, f)$ be a finite automate with n States

Let L be the regular set accepted by M. let $w \in L$, $|w| \geq n$ Then $\exists$ string x, y, z s.t $w = xyz$, $y \neq \Lambda$ & $xy^i z \in L$ $\forall i \geq 01$

Proof :- $L = T(M)$ $w \in L$

let $w = a_1 a_2 a_3 \dots a_m$, $m \geq n$

$|w| = m$

$$q_0 \xrightarrow[a_1 a_2 \dots a_m]{w} q_f$$

$\delta(q_0, a_1 a_2 \dots a_i) = q_i$ $\forall i = 1, 2 \dots m$

if $i=1$ $\delta(q_0, a_1) = q_1$

if $i=2$ $\delta(q_0 a_1 a_2) = \delta(\delta(q_0, a_1), a_2) = \delta(q_1, q_2)$
$= q_2$

Similarly

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots \xrightarrow{a_m} (q_m)$$

[n = no. of states.
$m \geq n$

We would get repeating states.

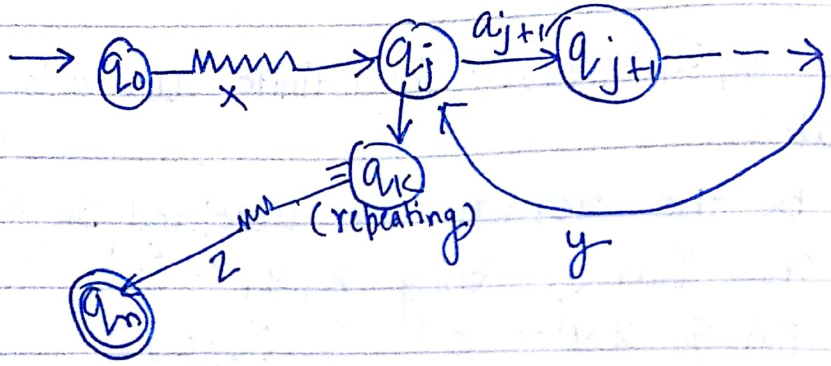
Suppose $(q_j\ q_k)$ is first pair [Repeating] $(j<k)$

So we break

$w = \underbrace{a_1 a_2 \ldots a_j}_{x}\ \underbrace{a_{j+1} \ldots a_k}_{y}\ \underbrace{a_{k+1} \ldots a_n}_{z}$

$w = xyz$

$\therefore |xy| \leq n$ (Since $|y| > 0$) ($y \neq \wedge$) given



xy^2z

xy^3z

$\vdots$

xy^iz always reaches to $q_n \in L$ (xy^iz)

→ which proves the theorem $\boxed{xy^iz \in L \ \forall i \geq 0}$

Date 21/Feb/18

Application of pumping lemma

$L \subseteq \Sigma^*$

① Show that L is not regular

Sol We proove it by Contradiction

Step1 Suppose L is regular Suppose the corresponding NDFA is $M = (Q, \Sigma, \delta, q_0, F)$, $|Q| = n$

$\exists\ x, y, z$ St. $w = xyz$, $|xy| \leq n$, $y \neq \wedge$

$xy^i z \in L$

Using one Example

Ex. Show that $L = \{ a^{i^2} \mid i \geq 1 \}$ is not regular

Soln Suppose L is regular then $\exists$ an NDFA

Say $M(Q, \Sigma, \delta, q_0, F)$ St. $L = T(M)$

Where $|Q| = n \rightarrow$ no. of State [must]

let $w \in L$ S.t $|w| \geq n$ Say $w = a^{n^2}$, $|w| = n^2 > n$

By Pumping lemma,

$\exists\ x, y, z$ S.t $w = xyz$, $|xy| \leq n$, $|y| \leq n$ ①

Consider the string xy^2z

$|xy^2z| = |x| + 2|y| + |z| > |x| + |y| + |z|$

$= n^2$

therefore $|xy^2z| > n^2$

Again $|xy^2z| = (|x| + |y| + |z|) + |y|$

$= n^2 + |y| \leq n^2 + n$ [since $|y| \leq n$]

$< n^2 + n + n + 1 = (n+1)^2$

therefore

$n^2 < |xy^2z| < (n+1)^2$

Which is not possible

$xy^2z \notin L$

(Length should be perfect Square)

Another Example

Ex Show that $L=\{a^p \mid p \text{ is a prime}\}$ is not regular? till

Soln Same as previous ①

$|xy^{p+1}z| = (|x|+|y|+|z|+p|y|)$ $\quad w = xyz$, $w = a^p$

$= p + p|y|$ $\quad |w| = |x|+|y|+|z| = p$

Let $y = a^m$ for some $m \geq 1$, $m \leq n$

$p + pm = p(m+1)$ → not a prime no: [it is product of 2 numbers]

hence $xy^{p+1}z \notin L$

hence it is not regular set.

Theorem If L is regular over Σ then $(\Sigma^* - L)$ is also regular

Pf L is regular over Σ ∃ an NDFA $M=(Q,\Sigma,\delta,q_0,f)$ s.t $L = T(M)$

Construct an NDFA $M' = (Q, \Sigma, \delta, q_0, f')$ where $f' = Q - f$

let $w \in T(M') \Rightarrow \delta(q_0, w) \in f' = Q - f$

i.e $\delta(q_0, w) \notin f$ i.e $w \notin L \Rightarrow w \in \Sigma^* - L$

$\therefore \Sigma^* - L$ is regular

Theorem if L_1 & L_2 be regular sets overs Σ then $L_1 \cap L_2$ is also regular over Σ

Pf $L_1 \cap L_2 = (L_1' \cup L_2')'$ [By de morgan's law]

$= \Sigma^* - (L_1' \cup L_2')$

$= \Sigma^* - [(\Sigma^* - L_1) \cup (\Sigma^* - L_2)]$

$= \Sigma^* - L$ Since L_1 & L_2 are regular, $\Sigma^* - L_1$ & $\Sigma^* - L_2$ is also regular

Since $\Sigma^* - L$ is regular as already proved

hence $L_1 \cap L_2$ is also regular over Σ

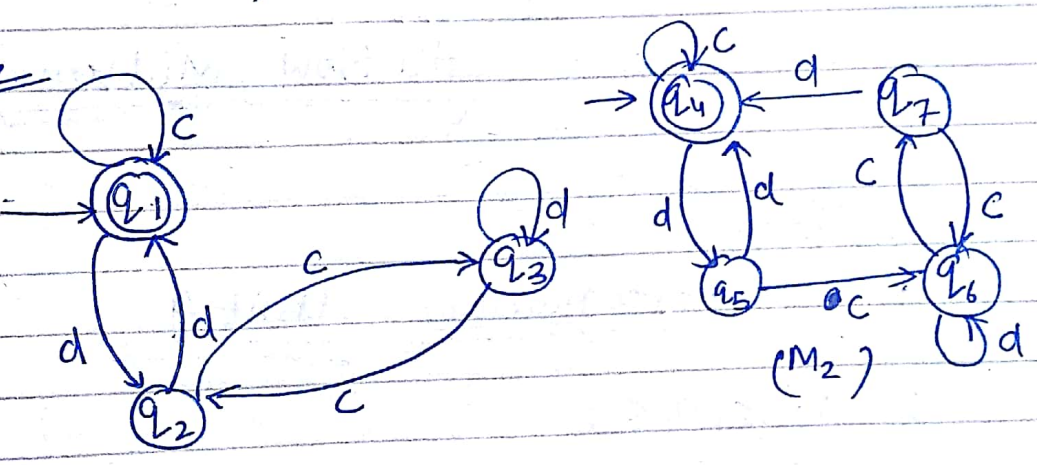
→ If 2 automaton say M & M' accept same set of strings over Σ then they are equivalent

→ Comparison Table

→ Consists of (n+1) cols, n = |Σ|

(q_0, q_0') (q_{a_1}, q_{a_1}') (q_{a_2}, q_{a_2}') — — — (q_{a_n}, q_{a_n}')

Example

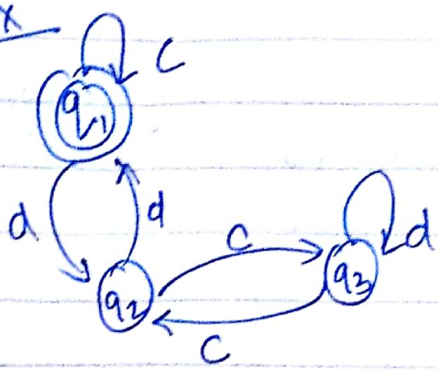


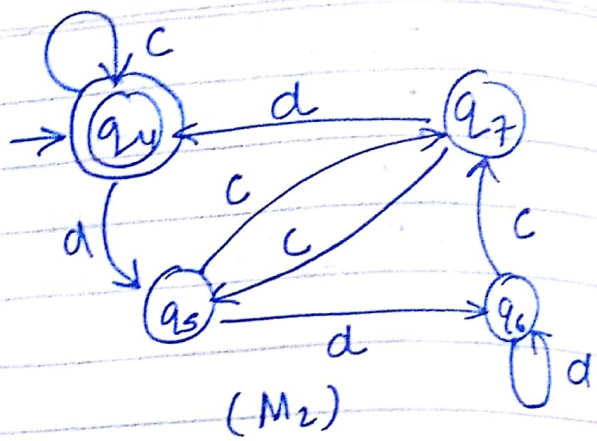
(q_1, q_1')	(q_c, q_c')	(q_d, q_d')
(q_1, q_4)	(q_1, q_4)	(q_2, q_5)
(q_2, q_5)	(q_3, q_6)	(q_1, q_4)
(q_3, q_6)	(q_2, q_7)	(q_3, q_6)
(q_2, q_7)	(q_3, q_6)	(q_1, q_4)

No new entry, we have one final & non final entry.

both machine are equivalent

Ex





(M_2)

M_1	c	d
$(q_1 q_4)$	$(q_1 q_4)$	$(q_2 q_5)$
$(q_2 q_5)$	$(q_3 q_2)$	$(q_1 q_6)$ → final, non final Stop here also both are non equivalent

till Now Mid Sem