Theory of computation.

TOC

3/Jan/2018.

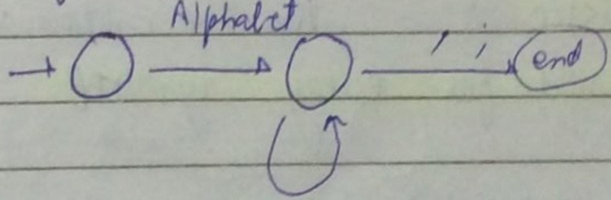
Formal Language and Autometa Theory
By Peter Linz.

One scheduled
Surprise Quize.

Theory of computation

High Level Language ⟶ Low Level Language.
conversion → check that Language Syntax correct.

Different Level of Automata.



Check Syntax.

drawing flow chart → Autometa.
conversion → compiler

Automata ⟶
1. finite Automata. — Least powerful.
2. push down Automata (FA + Stack)
3. Linear bounded Automata (FA + finite tape)
4. Turing Machine (FA + Infinite Tape).

↓

Capable of Processing any kind of language

Finite Automata

defined as 5 tuple, $Q, \Sigma, \delta, q_0, F$

$Q \rightarrow$ set of states

$q_0 \rightarrow$ initial state.

$F \rightarrow$ set of final states.

$\delta \rightarrow$ transition function.

$\Sigma \rightarrow$ input symbol.

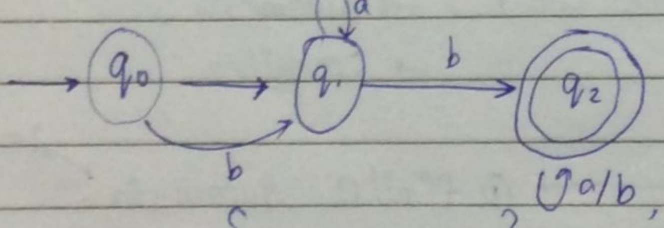
Finite Automata
- deterministic (DFA)
- Non Deterministic (NFA)

deterministic : Are the automata which have move defined for every symbol.

Move not defined are considered as dead move.

DFA.

$$\delta : Q \times \Sigma \longrightarrow Q$$



$Q = \{q_0, q_1, q_2\}$

$q_0 = \{q_0\}$

$F = \{q_2\}$

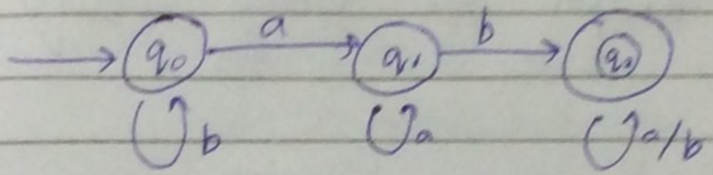
$\Sigma = \{a, b\}$

	a	b
$\rightarrow q_0$	q_1	q_1
q_1	q_1	q_2
q_2	q_2	q_2

$\delta(q_0, a) \rightarrow q_1$
$\delta(q_0, b) \rightarrow q_1$
$\delta(q_1, a) \rightarrow q_1$
$\delta(q_1, b) \rightarrow q_2$
$\delta(q_2, a) \rightarrow q_2$
$\delta(q_2, b) \rightarrow q_2$.

- Design a finite automata over $\{a, b\}$ that accepts string containing ab as substring.

bag ✗ aab ✓



baa ✗ babbab ✓

Acceptance by Final State.

If a string reaches final state then acceptable.

✗ baa

$(q_0, baa) \vdash (q_0, aa) \vdash (q_1, a) \vdash (q_1, a)$

Not accepted.

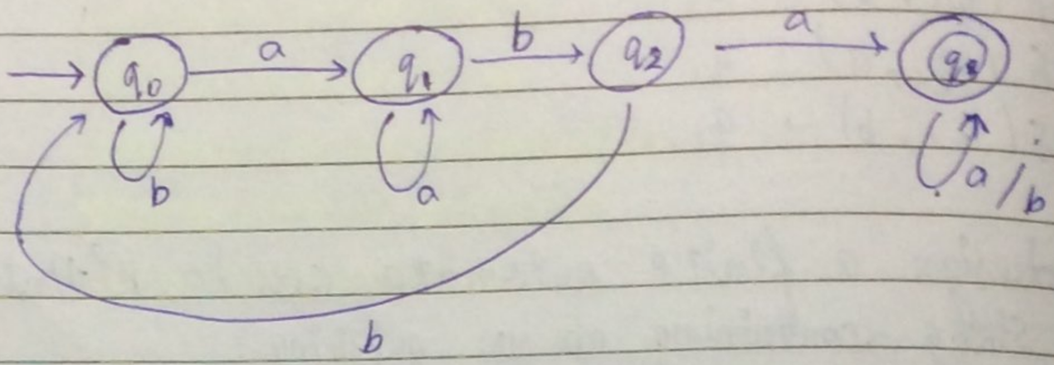
✓ abab a.

~~$(q_0, a) \vdash (q_1, b) \vdash (q_2, a) \vdash (q_2, b) \vdash (q_2, a)$~~

$(q_0, ababa) \vdash (q_1, baba) \vdash (q_2, aba) \vdash (q_2, ba) \vdash (q_2, a) \vdash (q_2, a)$

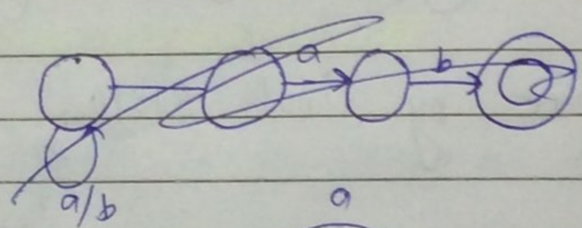
acceptable

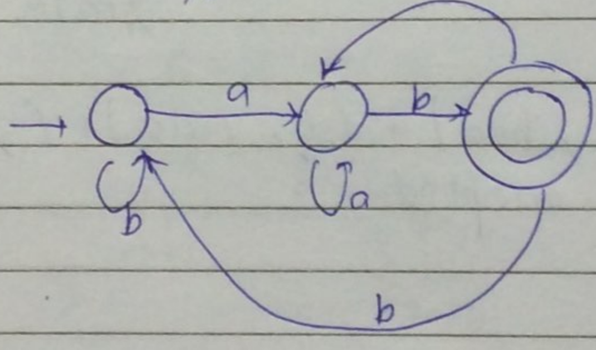
o Design with ab a as subscript



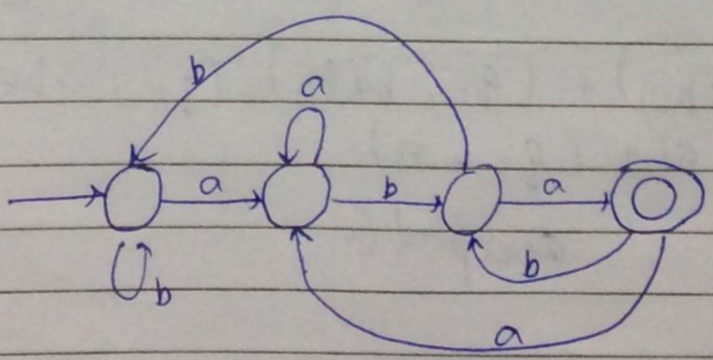
o FA that accepts all string ending with ab

*





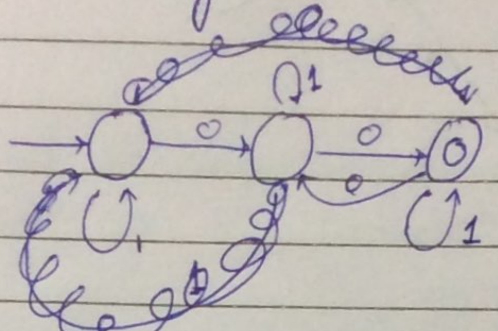
o Design end with aba



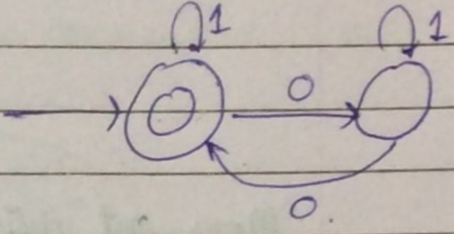
aba b

o $\Sigma = \{0, 1\}$ (no of 0 ≥ 1)

* Containing 0 in even number.



0 0



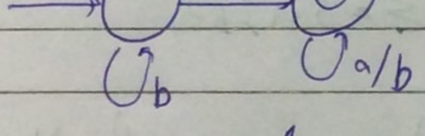
o $\Sigma = \{a, b\}$

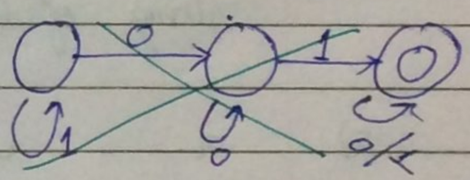
~~* Exactly one a~~ **

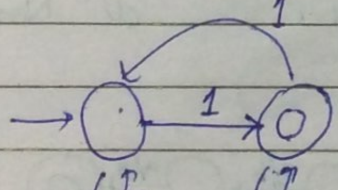
a) atleast one a.

b) 0 followed by 1. (*whenever 0 then 1)

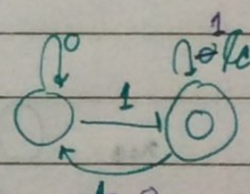
c) odd number of 1.

a) 

b) 

c) 

1 1 1

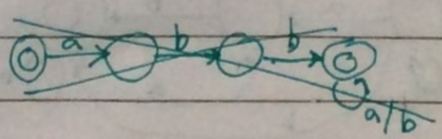


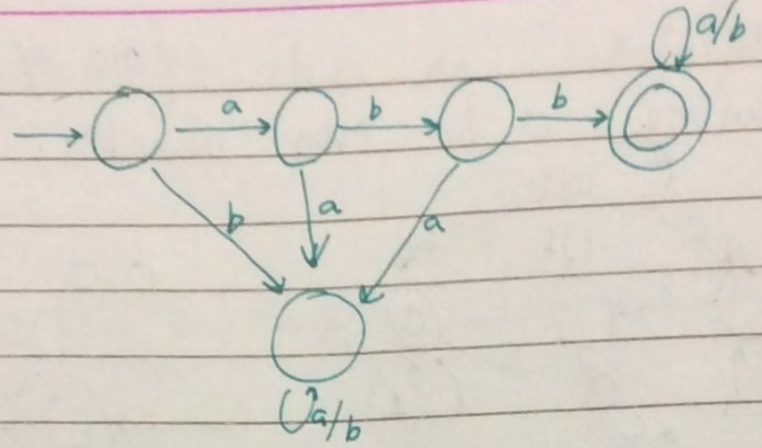
5/Jan/2018. Dead State : Is a state which has all outgoing moves defined to itself

i.e $\delta(q_d, a) \rightarrow q_d$

$\forall a \in \Sigma$

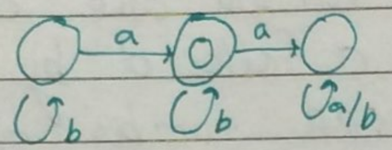
o Design a FA that accepts string starting with abb





Moves not defined are considered as dead move.

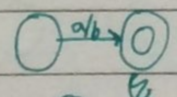
o $\Sigma = \{a, b\}$ accept all string containing exactly one a



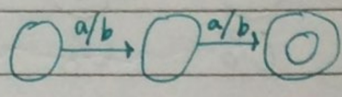
o What will be the min number of state required to accept string of length x over $\Sigma = \{a, b\}$

a) x. ~~b)~~ $x+1$ c) $x-1$ d) None of these

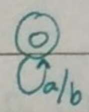
$x = 1$



$x = 2$



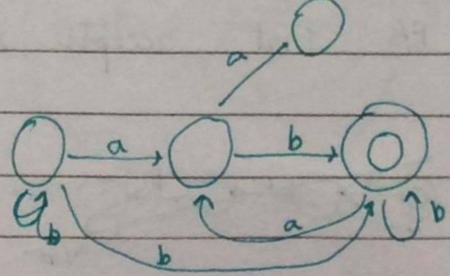
$x = 0$

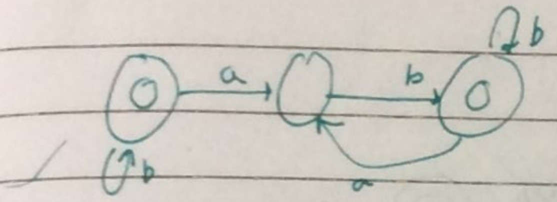


o $\Sigma = \{a, b\}$

* accept all string having a followed by b.

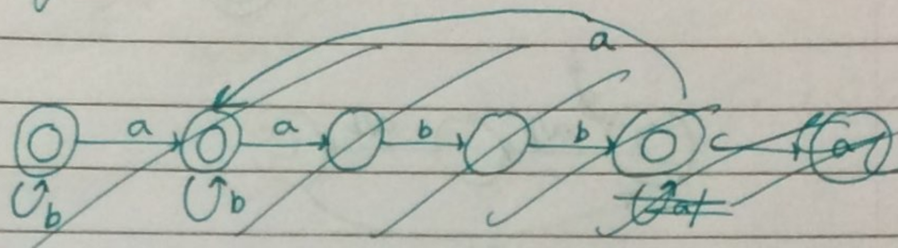
ab





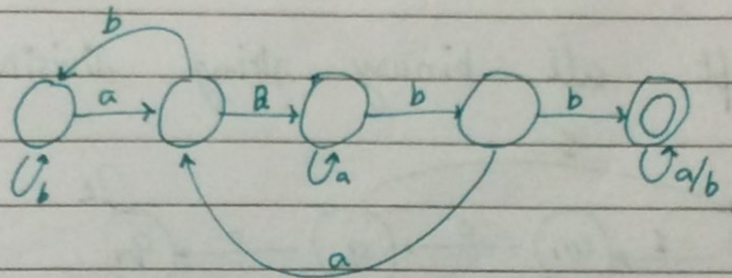
o Accept all the string containing two consecutive a followed by two consecutive b.

Atleast



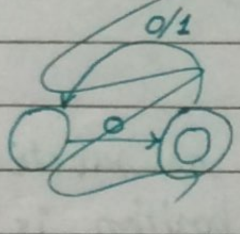
abaabb

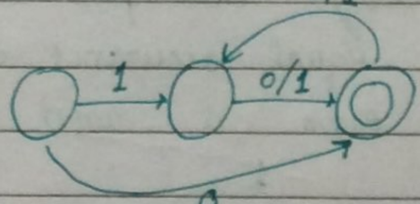
bb



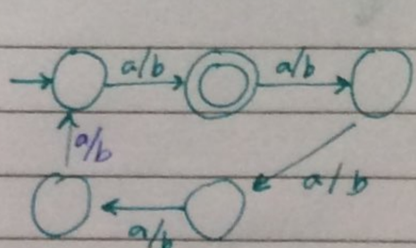
o All string ~~stra~~ start with 0 have odd length

* and strat with 1 have even length.



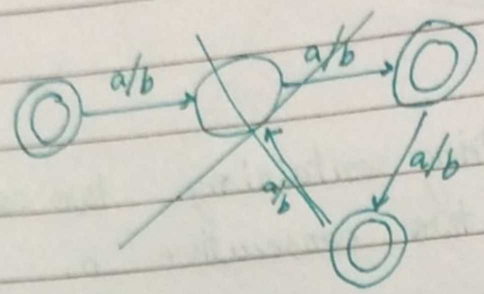


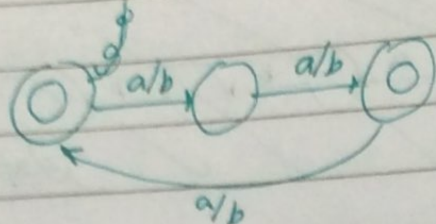
o Draw a FA over $\Sigma = \{a, b\}$ where $|\omega| \bmod 5 = 1$



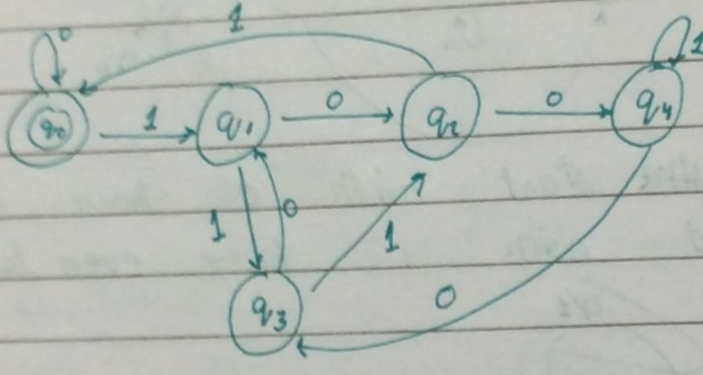
Q $|w| \bmod 3 \neq 1$

x 7





Q Accept all binary string divisible by 5.

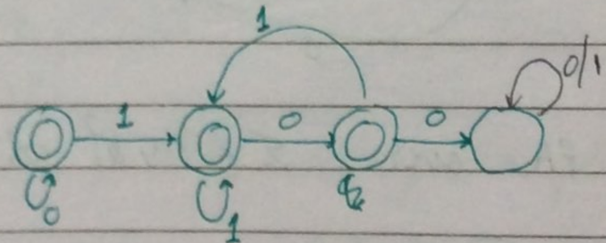


Q1. All string not containing 100 as substring

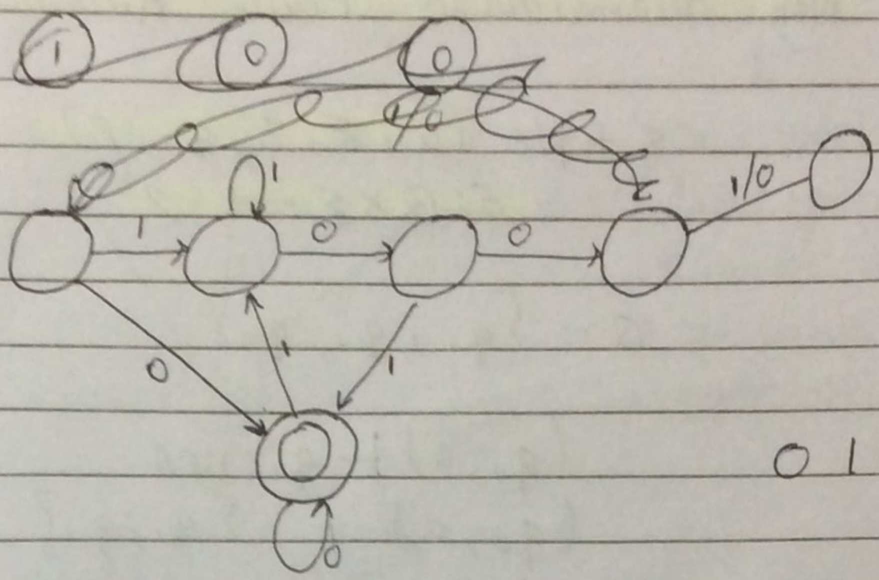
Q All string where every odd position is 1.

Q All string contain equal occurance of 01 or 10.

Q. " " " even a and odd b.

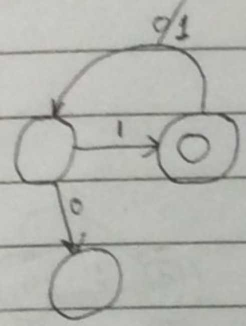


1.

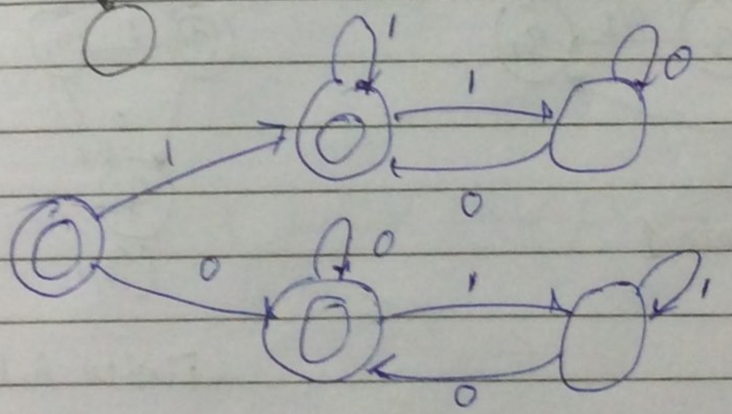


0 1 0 0 1

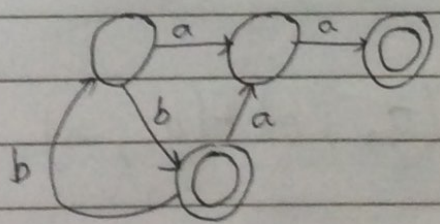
2.



3.



4.



15/Jan/2018

Non-deterministic Finite Automata

5 tuple $(Q, \Sigma, \delta, q_0, F)$

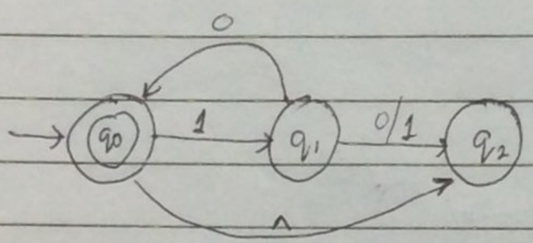
$\delta : Q \times \Sigma \longrightarrow 2^Q$

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

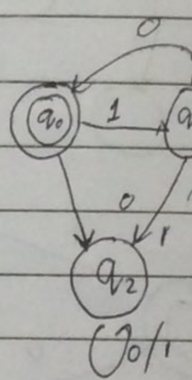
$(q_0, a) \vdash q_1$ DFA

$(q_0, a) \vdash \{q_1, q_2\}$

Acceptance of string.



Finite Automata 1.



Finite Automata 2

1010

$(q_0, 1010)$

$\vdash (q_1, 010)$

$\vdash (q_2, 10) \vdash (q_2, 0) \vdash (q_2, \wedge)$

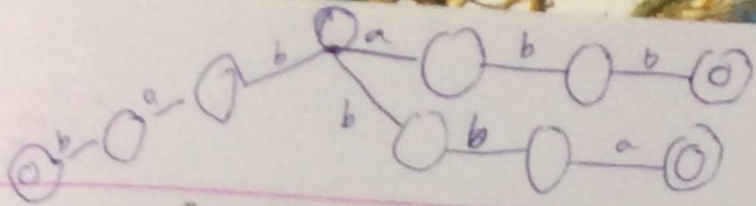
$(q_0, 1010)$

$\vdash (q_1, 010)$

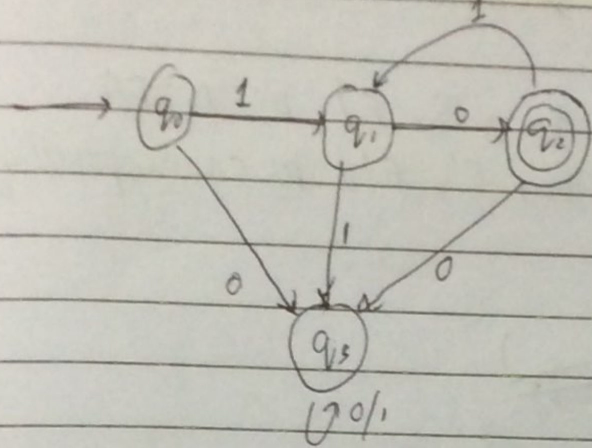
$\vdash (q_0, 10)$

$\vdash (q_1, 0)$

$\vdash (q_0, \wedge)$



$(10)^n$, $n > 0$

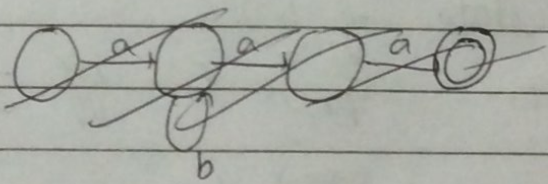


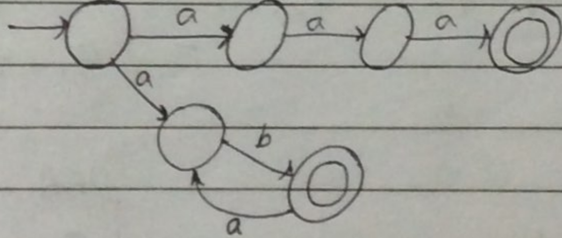
NFA

→ There exists many ~~input~~ path for a specific input

→ Use empty string transition.

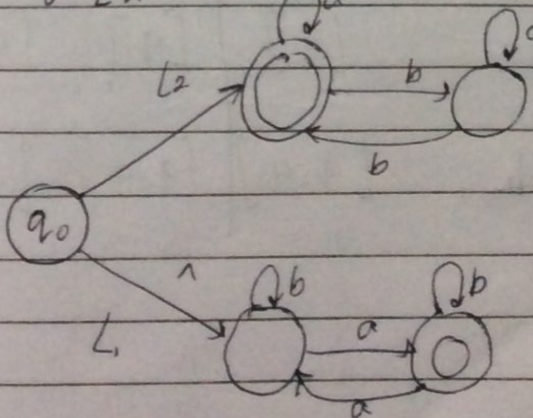
$L = (a^3 \cup (ab)^n \quad n \geq 0)$





2) L_1 : a in odd no. L_2 : b in even no.

$L_1 \cup L_2$



NDFA to DFA

$M = (Q, \Sigma, q_0, \delta, F)$ be NDFA
$M' = (Q', \Sigma, q_0', \delta', F')$ be corresponding DFA

~~$\delta'([q_1, q_2 \ldots q_i$~~

$Q' = [q_0, q_1, \ldots q_i]$ where $q_0, q_1 \ldots q_i \in Q$
$q_0' = [q_0]$
$\Sigma' = \Sigma$
F' = it is state that contains at least one final state of NDFA.
$[q_0, q_1 \ldots q_i] \in F'$ iff $\exists\, q_i \in F$

$\delta'([q_1, q_2 \ldots q_i], a) + (q_1, q_2, \ldots q_k)$
where $\delta(q_1, a) \cup \delta(q_2, a) \ldots \cup (q_i, a)$
$= \{q_1, q_2, \ldots q_n\}$

NDFA

	a	b
$\rightarrow q_0$	q_0	q_1
q_1	q_1	$q_0 q_1$

DFA

	a	b
$\rightarrow [q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1]$	$[q_0 q_1]$
$[q_0 q_1]$	$[q_0 q_1]$	$[q_1, q_0]$

~~$a^* \cup (ab)^* b$~~

NDFA

	a	b
$\rightarrow q_0$	$q_0 q_1$	q_2
q_1	q_0	q_1
q_2		$q_0 q_1$

NDFA

DFA

	a	b
$\rightarrow$ [q_0]	[$q_0 q_1$]	[q_2]
~~[q_1]~~	~~[q_0]~~	~~[q_1]~~
[q_2]		[$q_0 q_1$]
[$q_0 q_1$]	[$q_0 q_1$]	[$q_1 q_2$]
[$q_1 q_2$]	[q_0]	[$q_0 q_1$]

NDFA

	0	1	2
$\rightarrow q_0$	$q_1 q_4$	q_4	$q_2 q_3$
q_1		q_4	
q_2	q_2		q_3
q_3		$q_4 q_1$	
q_4	q_1	q_2	$q_1 q_2$

DFA

	0	1	2
$\rightarrow$ [q_0]	[q_1, q_4]	[q_4]	[q_2, q_3]
[q_4]	[q_1]	[q_2]	[$q_1 q_3$]
[q_1]		[q_4]	
[q_2]	[q_2]		[q_3]
[q_3]		[$q_4 q_1$]	
[$q_1 q_4$]	[q_1]	[$q_2 q_4$]	[$q_1 q_3$]
[q_2, q_4]	[$q_1 q_2$]	[q_2]	[$q_1 q_2 q_3$]
[$q_2 q_3$]	[q_2]	[$q_1 q_4$]	[q_3]
[$q_1 q_2 q_3$]	[q_2]	[$q_1 q_4$]	[q_3]

NDFA

	a	b
$\rightarrow q_0$	$q_1 q_2$	q_2
q_1	q_2	$q_1 q_3$
q_2	q_0	$q_1 q_0$
q_3	q_2	$q_1 q_2$

17/Jun/2018

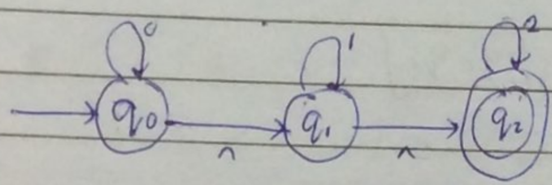
NFA with ^ transition

NFA to DFA with empty string transition.

1. ^ - moves
2. NFA to DFA
3. Minimize it
4. Implement it

$q_1 \xrightarrow{\wedge} q_2$

1. Duplicate all moves from q_2 to q_1 itself
2. If q_2 is FS then make q_1 also as final state.



Final

	0	1	2	^
→ q_0	q_0	q_1	q_2	-
q_1		q_1	q_2	
q_2			q_2	

Step 1

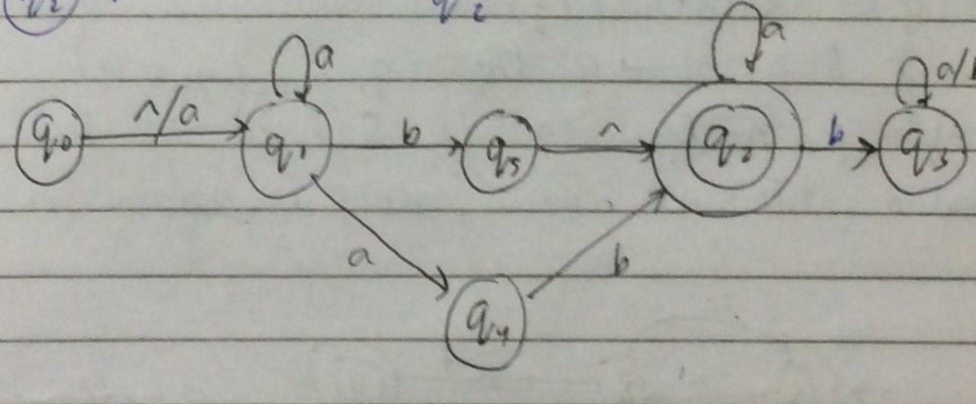
	0	1	2	^
q_0	q_0			q_1
q_1		q_1		q_2
q_2			q_2	

q

	0	1	2	Λ
$\rightarrow q_0$	q_0	q_1		q_2
q_1		q_1	q_2	q_2
q_2			q_2	

	0	1	2	Λ
$\rightarrow q_0$	q_0	q_1		q_2
q_1		q_1	q_2	
q_2			q_2	

	0	1	2	Λ
$\rightarrow q_0$	q_0	q_1	q_2	
q_1		q_1	q_2	
q_2			q_2	



	a	b	Λ
q_0	q_1		q_1
q_1	$q_1 q_4$	q_5	
q_2	q_2	q_3	
q_3	q_3	q_3	
q_4	q_2		
q_5			q_2

		a	b	^
	$\rightarrow q_0$	q_1, q_4	q_5	
	q_1	q_1, q_4	q_5	
	q_2	q_2	q_3	
Final	q_3	q_3	q_3	
	q_4		q_2	
	q_5	q_2	q_3	

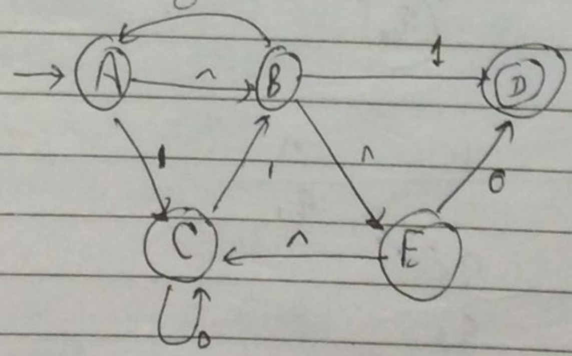
ϵ - closure (q)

* Is the set of state possible to reach from q with $\wedge$- moves

ϵ - closure (q)

1. $q \in$ ϵ-closure (q)
2. if $\delta(q, \wedge) \rightarrow q_i$ then $q_i \in \epsilon$-closure (q)
3. $\forall$ $p \in \wedge$-closure (q) if $\delta(p, \wedge) \rightarrow r$ then $r \in \wedge$-closure (q)

Repeat 3 till no new state is added in $\wedge$-closure (q)



ϵ - closure (A) $= \{A, B, E, C\}$

$\delta(q, \omega)$

$\epsilon\text{-closure}(q) \xrightarrow{\delta(\epsilon\text{-closure}(q),\, a)} P$

$\downarrow$

$\epsilon\text{-closure}(P) = R$

$\downarrow \delta(R, a)$

S

$\downarrow$

$\epsilon\text{ closure}(S)$

$\delta(B, 101)$

$\{BEC\}, 1$

$= (D, 01)$ $\{DBEC\}, 0$

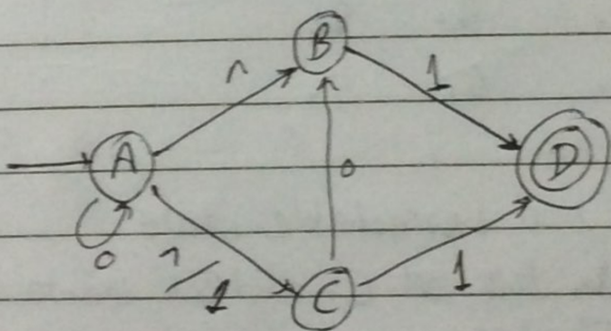
$= \{D\}$ $\{ADCBE\}, 1$

$\{CBD1\}$

$\delta(A, 01)$

$(\{A, B, E, C\}, 01)$

$\rightarrow \{ADC\} \rightarrow (\{ABECD\}, 1) \rightarrow \{CDB\} \rightarrow \{CDEB\}$



Task 1 $\delta(A, 01)$

Task 2 Remove Λ Move

Task 3 Convert NFA to DFA.

a b

22/Jan/2018

Minimization of finite Automata.

✓ State Equivalence

q_1 and q_2 are equivalent if

$(q_1, x) \rightarrow q_f$

than

$(q_2, x) \rightarrow q_f$

$\forall\ x \in \Sigma$

$(q_1, x) \vdash q_5 \qquad q_5 \notin F$

$(q_2, x) \vdash q_k \qquad q_k \in K$

	a	b	c
$\rightarrow q_1$	q_1	q_2	q_4
q_2	q_1	q_3	q_4
q_3	q_4	q_2	q_4
q_4	q_1	q_1	q_4

1. Remove all dead and unreachable state
2. Divide set Q into two set Q_1 and Q_2 containing set of NF and Final State.
3. Check equivalence for every state in Q_1 and Q_2 if the state in set are not equivalent split the set
   Repeat above till all states in set are equivalent

	$q = 0$	$q = 1$
$\rightarrow q_0$	q_1	q_5
q_1	q_6	q_2
q_2	q_0	q_2
~~q_3~~	~~q_2~~	~~q_6~~
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2
q_8	q_8	q_8

~~$q_0 q_1 q_5 q_6 q_2 q_4 q_7$~~

~~$Q_1(q_0, q$~~

$P_1(q_0, q_1, q_4, q_5, q_6, q_7)$ $P_2(q_2)$

q_0 a

$P_1(q_0, q_1, q_4, q_5, q_6, q_7, q_8)$ $P_2(q_2)$

$P_3(q_0, q_4, q_6)$ $P_4(q_1, q_5, q_7)$

(q_0, q_4) (q_6) (q_1, q_7) q_5

P5 P6 P7 P8

q_0 a → P_1
q_0 b → P_1
q_1 a → P_1
q_1 b → P_2
q_4 a → P_1
q_4 b → P_1
q_5 a → P_2
q_5 b → P_3
q_6 a → P_3
q_6 b → P_3
q_7 a → P_3
q_7 b → P_2

q_0, a → P4
q_0, b → P4
q_4, a → P4
q_4, b → P4
q_6, a → P3
q_6, b → P3.

q_1, a → P6
q_1, b → P2
q_5, a → P2
q_5, b → P6
q_7, a → P6
q_7, b → P2.

q_0, a → P7
q_0, b → P8
q_4, a → P7
q_4, b → P8

q_1, a P6
q_1, b P2
q_7, a P6
q_7, b P2.

	a	b
→ P5	P7	P8
P6	P6	P5
P7	P6	P2
P8	P2	P6
P2	P5	P2

	a	b	
$\rightarrow q_0$	q_1	q_2	
q_1	q_0	q_2	
q_2	q_3	q_1	
q_3	q_3	q_0	
q_4	q_3	q_5	x
q_6	q_5	q_6	x
q_7	q_6	q_3	x
q_5	q_6	q_4	x

$P_1(q_0\ q_1\ q_2\ \cancel{q_3})$ $P_2(q_3)$

q_2

$P_3(q_0, q_1)$ $P_4(q_2)$

$P_5(q_0)$ $P_6(q_1)$

q_0	a	P_1
q_0	b	P_1
q_1	a	P_1
q_1	b	P_1
q_2	a	P_2
q_2	b	P_1
$\cancel{q_3}$	$\cancel{a}$	
$\cancel{q_3}$	$\cancel{b}$	

q_0 a
q_0 b
q_1 a
q_1 b

^

	a	b
$\rightarrow q_0$	q_0	$q_0 q_2$
~~q_1~~	~~q_3~~	~~$q_2 q_3$~~
q_2	q_4	–
~~q_3~~	~~q_4~~	~~–~~
q_4	–	–

$q_0 q_2 q_4$

	a	b
$[q_0]$	$[q_0]$	$[q_0 q_2]$
$[q_0 q_2]$	$[q_0 q_4]$	$[q_0 q_2]$
$[q_0 q_4]$	$[q_0]$	$[q_0 q_2]$

29/Jan/2018 **Regular Expression**

→ It is an algebra notation for the language accepted by FA

→ it includes three symbol +, ·, *

+ union
· concatenation
* kleene closure

$a^* =$ 0 or more occurance of a

$L = a^*$

$= \{ \wedge, a, aaa \_\_\_\_ \}$

$(a+b) =$ Either a or b

$a \cdot b$

$\Sigma = \{a, b\}$

① L = all string containing abb as substring abb

~~abb~~

$(a+b)^* = \{ \wedge, a, b, ab, aa, ba, bb \ldots$

Ans $(a+b)^* abb (a+b)^*$

② All string ending with abb

$(a+b)^* abb$

③ Even no of a.

$(b^* a b^* a b^*)^*$

$b^* (a b^* a)^* b^*$

④ All string containing at least 3 occurance of a.

$\cancel{b^* a b^* a b^* a b^* a^*}$

$(b^* a b^* a b^* a)(a+b)^*$

⑤ Exactly 3

$b^* a b^* a b^* a b^*$

→ L = a.b+c

a.(b+c) ✗ (ab+c) ✓

(ab, ac) (ab, c)

~~ab~~

~~abc~~

~~c~~

Precedence → Star $\quad * \quad *$
→ Concatenation $\quad \cdot$
→ Union $\quad +$

→ $L = a^{2n} b^{2m+1} ab \quad m>0 \quad n>0$

$R_1: (aa)^*(bb)^* b\ ab$ ✗
$R_2: aa(aa)^* b(bb)^* bab$ ✓
$(aa)^+ (bb)^+ bab$

$aa^* = a^+$ = +ve kleene closure
$RR^* = R^+$

⑥ At least one pair of consecutive zero

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

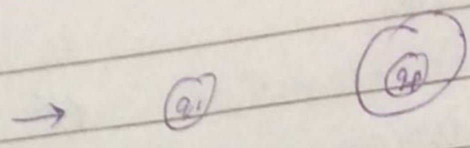
⑦ No pair of consecutive 0.

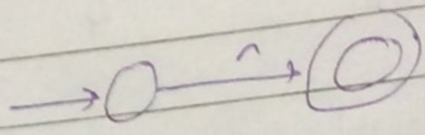
~~$1^*(010)^*1^*$~~
$(1+01)^*(0+\wedge)$

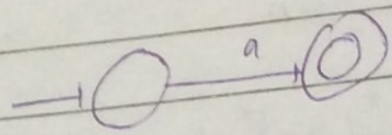
Kleene Theorem
→ For every R.E. there exists a FA to accept it

Lang
↓
RE
↓
ϵ -NFA
↓
NFA
↓
DFA
↓
minimize

Proof.

$R = \phi$ 

$R = \wedge$ 

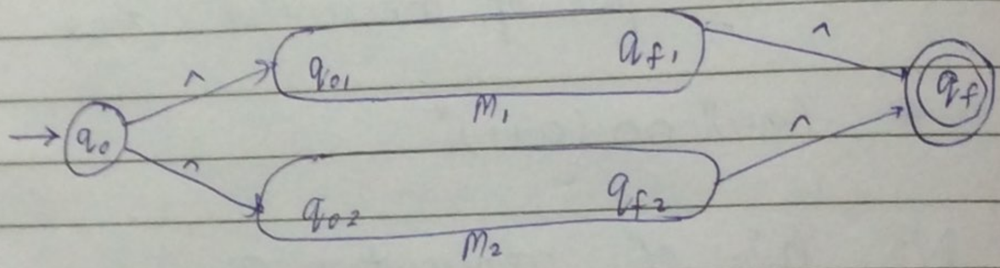
$R = a$ 

There exist FA for RE R_1 and R_2

R_1 : it is $M1(Q_1, q_{01}, \Sigma_1, \delta_1, q_{f_1})$

R_2 : " " $M2(Q_2, q_{02}, \Sigma_2, \delta_2, q_f)$

$R = R_1 + R_2$



$M(Q, \Sigma, \delta, q_0, q_f)$

$Q = Q_1 \cup Q_2 \cup \{q_0, q_f\}$

$\Sigma = \Sigma_1 \cup \Sigma_2$

$\delta = \delta_1 \cup \delta_2 \cup q_0, \wedge \rightarrow \{q_{01}, q_{02}\}$

$\cup \quad q_{f1}, \wedge \rightarrow q_f$

$\cup \quad q_{f_2} \wedge \rightarrow q_f$

30/Jan/2018.

Finite Automata to Regular Expression.

Identies

1. $\phi + R = R$
2. $\phi \cdot R = \phi$
3. $\wedge \cdot R = R$
4. $\wedge^* = \wedge$
5. $R^* R = RR^* = R^+$
6. $R^* R^* = R^*$
7. $R + R = R$
8. $\wedge + RR^* = R^*$
9. $(PQ)^* P = PQPQ \_\_ PQP = P(QP)^*$
10. $(P+Q)^* = (P^* Q^*)^* = (P^* + Q^*)^*$
11. $(P+Q)R = PR + QR$
12. $\phi^* = \epsilon \quad \phi^* = \epsilon$

8 Feb
17 Feb
Quiz

Arden Theorem

$R = Q + RP$ and $P \neq \wedge$
then $R = QP^*$

$R = QP^*$

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

$R = Q + RP = Q + (Q+RP)P = Q + QP + RP^2$
$= Q + QP + (Q+RP)P^2 = Q + QP + QP^2 + RP^3$

$$R = \underbrace{(Q + QP + QP^2 + \_\_ QP^i)}_{A} + \underbrace{RP^{i+1}}_{B}$$

Let $\omega \in R$ and $|\omega| = i$

$\therefore R \in (Q + QP + \quad QP^i)$
or
$R \in RP^{i+1}$

$P \neq \wedge$ So $|RP^{i+1}| \geq i+1$

therefore $w \notin RP^{i+1}$

$w \in Q + QP + \dots QP^i$

So
$$R = Q + QP + \dots QP^i$$
$$= Q(\wedge + P + \dots P^i)$$
$$R = QP^*$$

Change Automata to Regular Expression.

Write equation for each state $q_i \in Q$ describing its reachability from other states.

$$q_1 = q_{11}q_1 + q_{12}q_2 + q_{13}q_3 + \dots q_{1n}q_n$$
$$q_2 = q_{21}q_1 + q_{22}q_2 + q_{23}q_3 + \dots + q_{2n}q_n$$
$$q_n = q_{n1}q_1 + q_{n2}q_2 + \dots + q_{nn}q_n$$

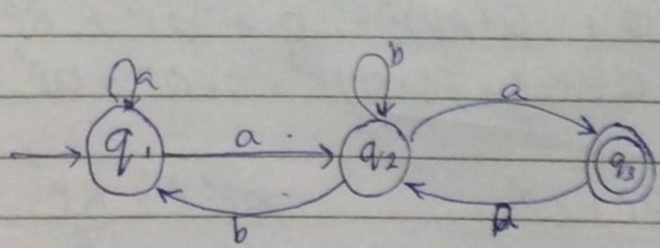
Apply identities and Arden Theorem and generate

$$q_f = q_0 q_{ij}$$

Remove q_0.

This is the final automata.

Q.



$q_1 = q_1 a + b q_2$ — (1)

$q_2 = a q_1 + b q_2 + a q_3$ — (2)

$q_3 = a q_2$ — (3)

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

$\underset{R}{q_2} = \underset{Q}{q_1 a} + \underset{R}{q_2}\underset{P}{(b+aa)}$

$q_2 = q_1 a(b+aa)^* \quad — (4)$

(1) and (4)

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

$\underset{R}{q_1} = \underset{R}{q_1}\underset{P}{(a + a(b+aa)^* b)} + \underset{Q}{\wedge} \qquad R = RP + Q$

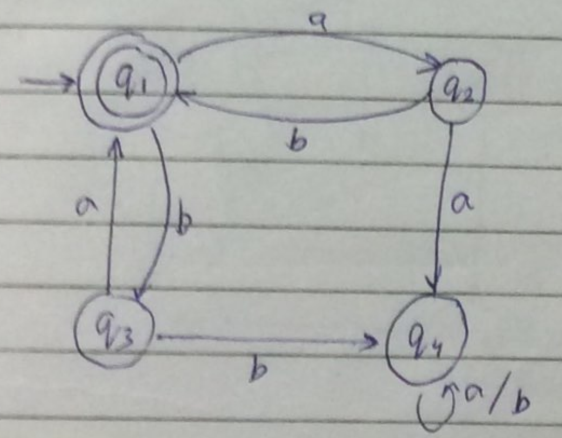
$q_1 = \wedge(a + a(b+aa)^* b)^* \qquad R = QP^*$

$q_1 = (a + a(b+aa)^* b)^* \quad — (5)$

eq (3) $q_3 = q_1 a(b+aa)^* a$

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

Q



$q_1 = bq_2 + aq_3 \qquad = baq_1 + abq_1 = q_1$

$q_2 = aq_1 \qquad = abq_2 + aaq_3 \qquad = q_1(ab+ba) + \wedge$

$q_3 = bq_1 \qquad = bbq_2 + baq_3 \qquad = (ab+ba)^*$

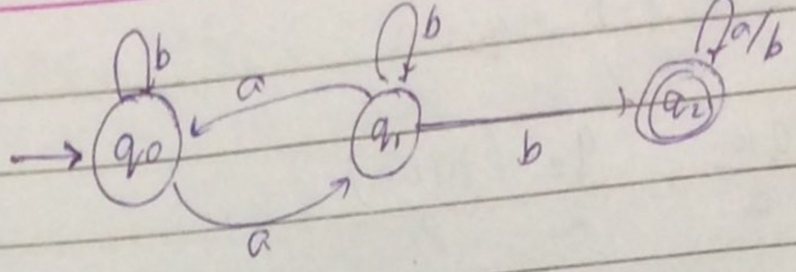
$q_4 = aq_4 + bq_4 + bq_3 + aq_2$

$\cancel{q_4 = aq_4 + bq_4 + bbq_1 + aaq_1}$

$\cancel{q_2 = abq_2 + aaq_3}$

$\cancel{q_3 = bbq_2 + baq_3 = bb(abq_2 + aaq_3) + baq_3}$

Q.



$$q_0 = q_0 b + q_1 a \cancel{+ q_1 b} \quad -①$$

$$q_1 = q_1 b + q_0 a \quad -②$$

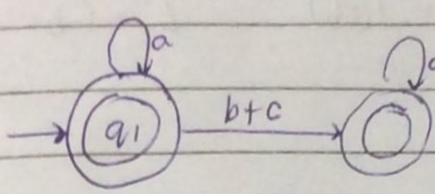
$$q_2 = a q_2 + b q_2 + b q_1 \quad -③$$

$$q_1 = q_1 b + ab q_0 + aa q_1 + ab q_1$$

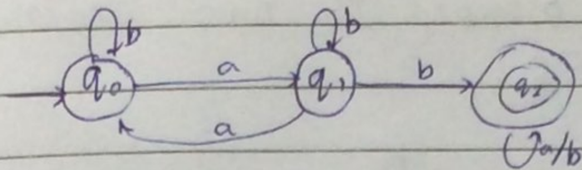
$$q_2 = (a+b) q_2 + b q_1$$

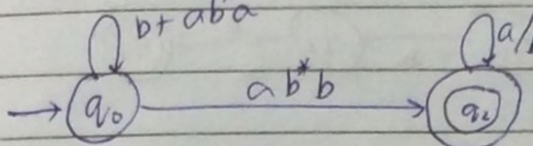
$$q_2 = (a+b) q_2 + bb q_1 + ba q_0$$

$$(q_0 + q_1) = (q_0 + q_1)(a+b)$$

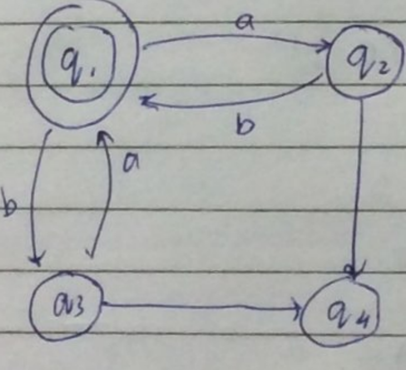


$(a^* + (b+c)a^*)^*$

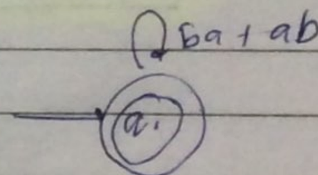




$(b+ab^*a)^* \, ab^*b \, (a+b)^*$



$(ab + ba)^*$



Closure propertie on regular language

L_1 and L_2 are Regular language

Union $L = L_1 \cup L_2$

Concatenation $L = L_1 \cdot L_2$

kleeve closure

$L = L_1^*$

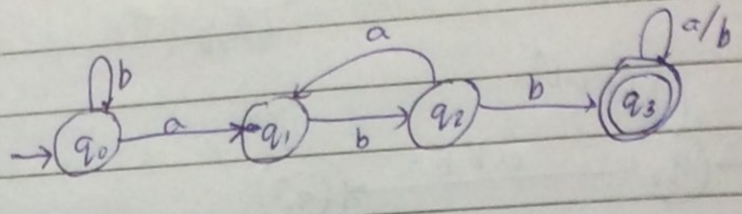
$L^R = (ba)^*$

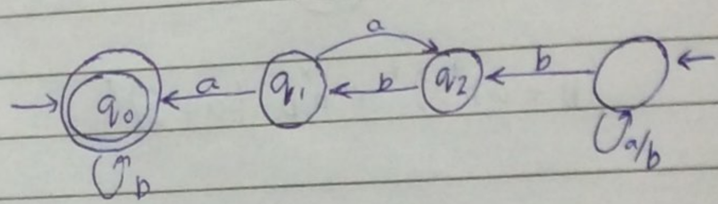
Reversal $L = L_1^R$ $\quad L_1 = (ab)^*$

Procedure

1. Reverse the move
2. Make IS ~~as~~ as FS (Final State).
3. Make FS as IS if there are more than one FS then create a new IS and $\wedge$- move to all the FS.

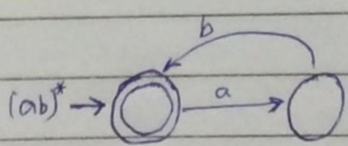
$\boxed{L_1} \Rightarrow$

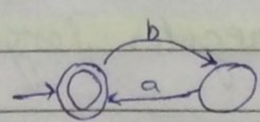




$(a+b)^* \, bba \, (a+b)^*$

Complement

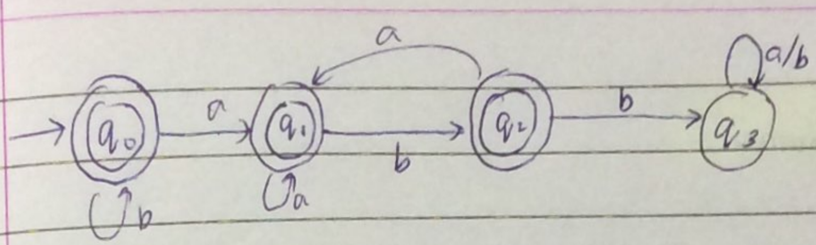




$L = \overline{L_1}$

Procedure

1. Make FS as NFS (Non final state)
2. Make all non-FS as FS



Intersection

$L = L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ $\quad L_1 = \{q_{01}, q_{f1}, \Sigma, \delta_1, Q_1\}$

$L_2 = \{q_{02}, q_{f2}, \Sigma, \delta_2, Q_2\}$

Even 0 and odd 1

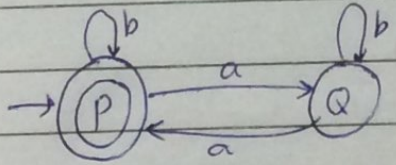
Procedure

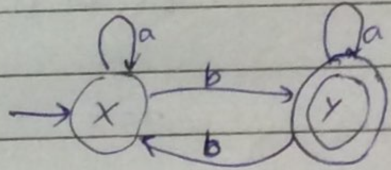
→ Make $[q_{01}, q_{02}]$ as initial state for FAL

→ $\delta[q_{12}, q_{26}], a = \{\delta_1(q_{1i}, a), \delta_2(q_{2i}, a)\}$

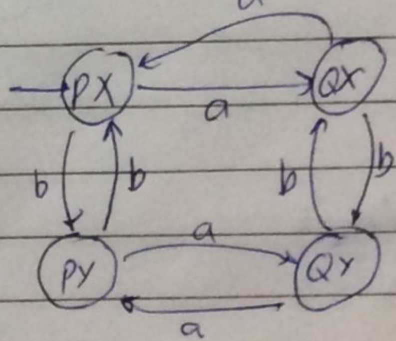
→ $[q_{f1}, q_{f2}]$ is FS

$Q = [q_i, q_j]$ where $q_i \in Q_1$, $q_j \in Q_2$.





	a	b
→ PX	QX	PY
QX	PX	QY
PY	QY	PX
QY	PY	QX



$L = L_1 - L_2$
$= L_1 \cap \overline{L_2}$

1/Feb/2018

Pumping Lemma.

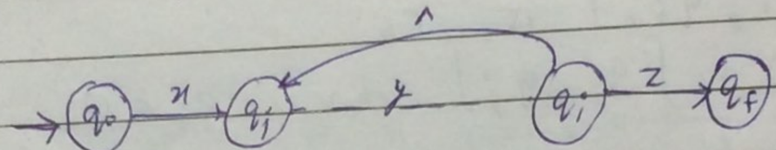
$Q: \Sigma, \delta, q_0, F$

if L is regular then each string $w \in L$ can be written $x\ y\ z$ i.e $w \in xyz$

where $|y| \geq 1$ $\quad |w| \geq m$

and $|xy| \leq M$ — No of stage.

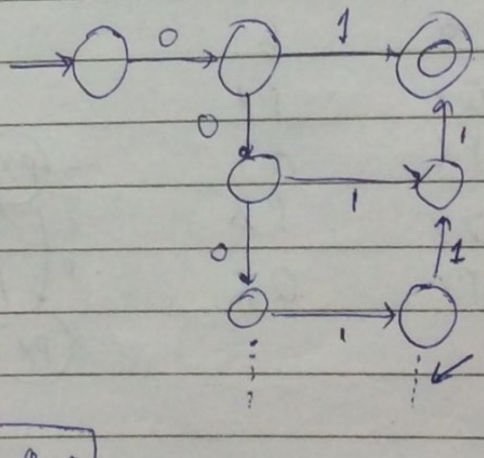
then $\forall i \geq 0$ $\ xy^i z \in L$



$L = 0^n 1^n$ $\quad n > 0$

$n = 1$ $\quad w = 01$

$n = 2$ $\quad w = 0011$



~~Any string is~~

Not finite.

k = no of states.

~~Let~~ $\boxed{L = 0^n 1^n}$ $\ n > 0$ $\quad |Q| = k$ $\quad K \geq M$

$w = xyz = 0^m 1^m$ $\quad |w| = 2m > k$

$\boxed{\text{Let } y = 0}$ $\quad |y| \geq 1$ $\quad |xy| \leq k$

$w = \underset{x}{0^{m-1}}\ \underset{y}{0}\ \underset{z}{1^m}$

Non Regular.

$i = 0$ $\quad w = 0^{m-1} 1^{m} \notin L$

$i = 1$ $\quad w = 0^{m} 1^{m}$

$i = 2$ $\quad w = 0^{m-1} 0^{2} 1^{m} = 0^{m+1} 1^{m} \notin L$

$\boxed{\text{Let } y = 1}$ $\quad 0^{m} 1 1^{m-1}$

Non regular. $\quad i = 2 \quad 0^{m} 1^{2} 1^{m-1} \notin L$

$\boxed{\text{Let } y = 01}$

$$w = \underbrace{0^{m-1}}_{x} \underbrace{01}_{y} \underbrace{1^{m-1}}_{z} \in L$$

Non Regular.

$i = 2 \quad 0^{m-1} (01)^{2} y^{m-1}$

$0^{m-1} \, 0101 \, y^{m-1} = 0^{m} 10 1^{m} \notin L.$

$\boxed{L = 0^{2n}}$ $\quad n \geq 0$

$y = 00$ $\quad (00)^{i} 0^{2(n-1)} \in L$

$x = \wedge$

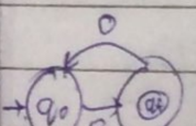
$z = 0^{2(n-1)}$

Regular.

$x = 0^{2(n-1)}$ $\quad K = 2n.$

$y = 00$

$z = \wedge$



a^{*}

5/Feb/2018

Regular Grammar

$S \to NVN$
$N \to$ Ram / apple / Book.
$V \to$ ate / Read.

$G(V, \Sigma, P, S)$

$V = \{S, N, V\}$

$V \to$ is set of NT
$\Sigma \to$ set of terminal
$P \to$ production
$S \to$ Start Symbol

$\Sigma = \{$Ram, apple, book, ate, read$\}$

P:

$S \Rightarrow NVN$
$N \Rightarrow$ Ram
$V \Rightarrow$ ate
$V \to$ read
$N \to$ apple
$N \to$ book.

Ram ate apple.

$S \to NVN$
$\to$ Ram V N.
$\to$ Ram ate N.
$\to$ Ram ate apple.

A grammer G is regular if all the production are of form

$A \to a \mid aB$ (Right linear)

or

$A \to a \mid Ba$ (left linear)

$S \to abA$
$A \to aA \mid a$

$S \to abA$
$S \to aba$

$S \to abA$
$\to abaA$
$\to abaaA$

$L = abaa^*$ (Right Linear)

$S \rightarrow Aba$
$A \rightarrow a | Ba$
$B \rightarrow Ba | a$

~~$L = aa^*ba$~~ or $L = a^*aba$ (Left Linear)

$S \rightarrow aA$
$A \rightarrow Bb | b$
$B \rightarrow aB | a$

ab
$aaBb$ $\quad a^*ab$

$S \rightarrow aA$
$A \rightarrow Bb | b$
$B \rightarrow aA$

aBb $\quad ab$
$aaAb$ $\quad aabb$ $\quad aaBbb$
$aaaAbb$
$a^n b^n \ \forall \ n > 0$ $\quad aaabbb$

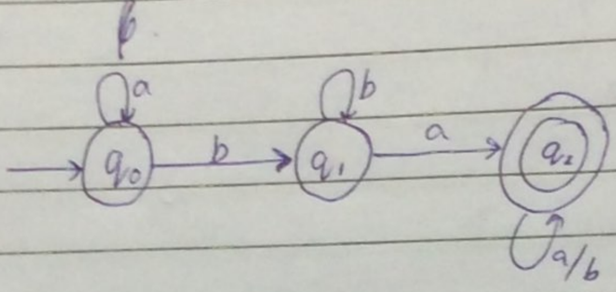
If all ~~Left~~ Left Linear or Right Linear grammar regular ~~or not~~

Otherwise (mixed) can be regular or not

FA to regular ~~re~~ grammar

$\delta(q_i, a) \longrightarrow q_j$ when $q_j \notin F$ $\qquad$ $P: A_i \longrightarrow a A_j$

$\delta(q_i, a) \rightarrow q_j$, $q_j \in F$ $\qquad$ $P: A_i \longrightarrow a A_j | a$



$(q_0, a) \vdash (q_0, a)$

$P: A_0 \longrightarrow a A_0$

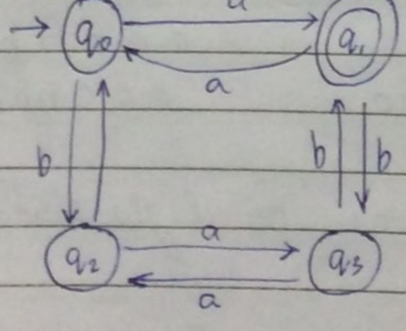
$A_0 \rightarrow b A_1 \quad (q_0, b) \rightarrow q_1$

$A_1 \rightarrow b A_1$

$A_1 \rightarrow a A_2 | a$

$A_2 \rightarrow a A_2 | a$

$A_2 \rightarrow b A_2 | b$



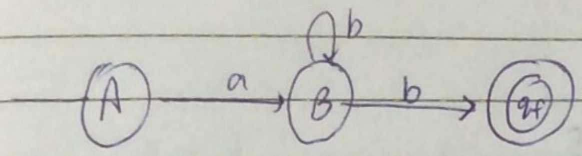
$P: \quad A_0 \longrightarrow a A_1 | a \qquad A_2 \longrightarrow b A_0 | b$

$A_0 \longrightarrow b A_2 \qquad A_3 \rightarrow a A_2$

$A_2 \longrightarrow a A_3 \qquad A_3 \rightarrow b A_1$

$A_1 \rightarrow aA_2$ then $\delta(q_1, a) \rightarrow q_2$
$A_1 \rightarrow a$ then $\delta(q_1, a) \rightarrow q_f$
$q_f \in F$

Eg
$A \rightarrow aB$
$B \rightarrow bB/b$.



Eg.
$A \rightarrow aB/bC$
$B \rightarrow bB/b$
$C \rightarrow aA/a$.

6/Feb/2018.

Context free grammar

→ A grammar $G(V_N, \Sigma, P, S)$ is CFG if all the production are of form

$A \rightarrow \alpha$ where $\alpha \in (V_N \cup \Sigma)^*$

$S \rightarrow AB$
$A \rightarrow aaA$
$A \rightarrow \wedge$
$B \rightarrow Bb$
$B \rightarrow \wedge$

Eg aaab

$S \rightarrow AB$
$S \rightarrow aaAB$
$S \rightarrow aaaaAB$

does not belong to this automata.

Eg aab

$S \rightarrow AB$

$\rightarrow aaAB \quad (A \rightarrow aaA)$

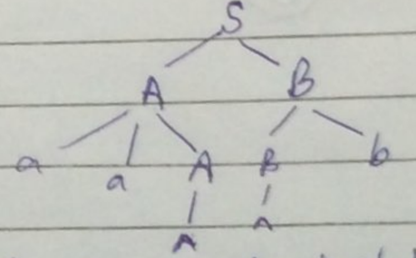
$\rightarrow aaB \quad (A \rightarrow \wedge)$

$\rightarrow aaBb \quad (B \rightarrow Bb)$

$\rightarrow aab \quad (B \rightarrow \wedge)$

belong to grammar.

derivation tree



⇒ string: ababcbcbb. ⟶ belong to the grammar

$S \rightarrow abScB$

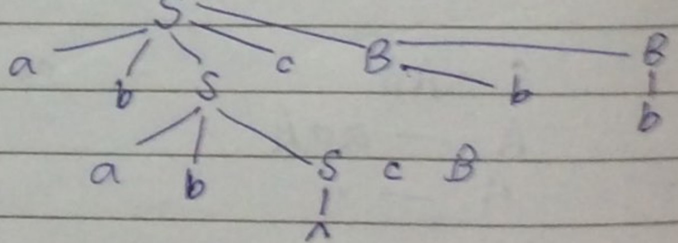
$S \rightarrow \wedge$

$B \rightarrow bB \mid b$

~~ababScB~~

~~ababcB~~

derivation tree.



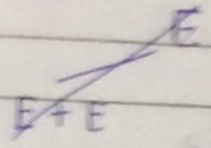
Ambiguous grammar

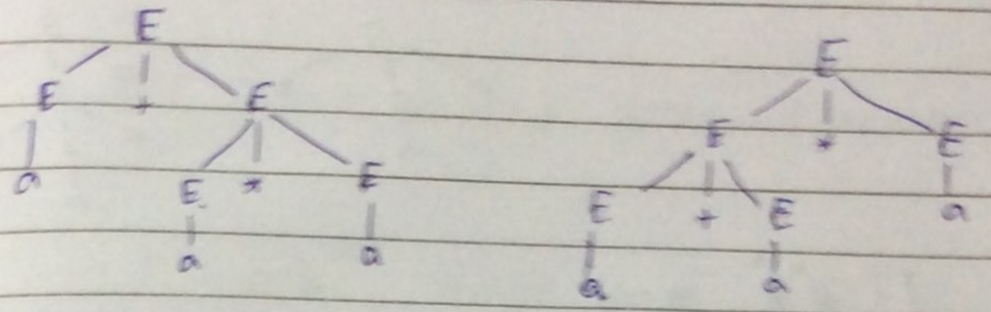
⇒ $E \rightarrow E+E \mid E*E \mid a$

string a+a*a.

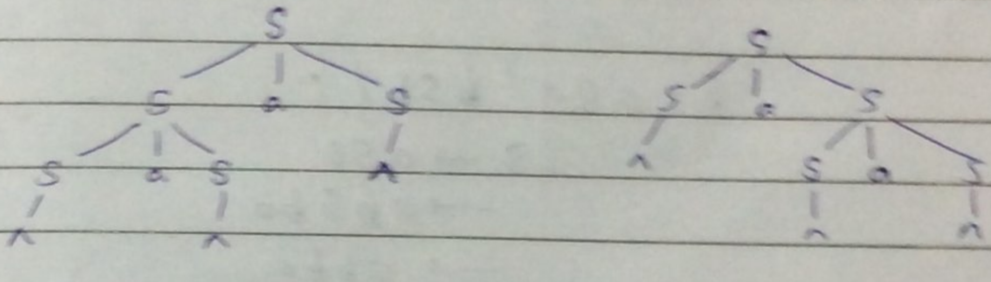
E + E

E + E * E





⇒ S → SaS | ^



⇒ S → 0A | 1B

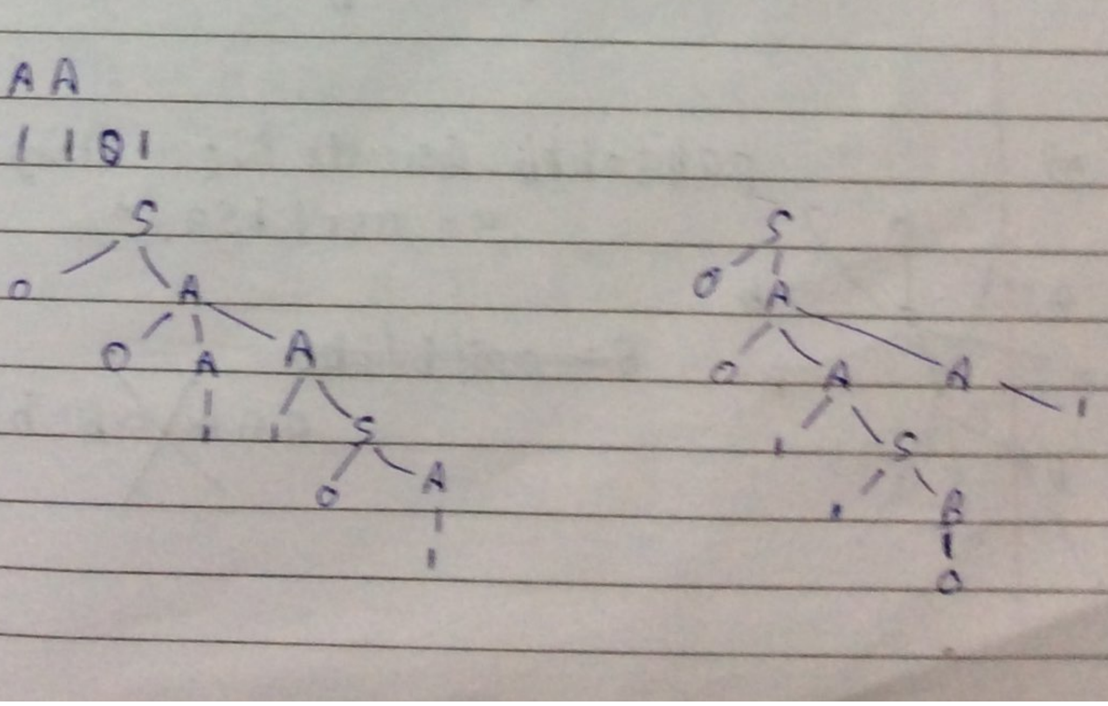
A → 0AA | 1S | 1

B → 1BB | 0S | 0

001101

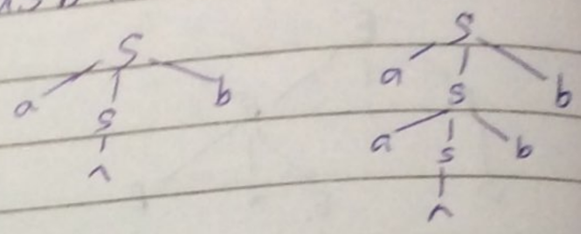
00AA

001101



$\Rightarrow$ $L = a^n b^n \quad n \geq 0$

$S \rightarrow aSb \mid \wedge$



$L = a^n b^n \quad n > 0$

$S \rightarrow aSb \mid ab$

$\Rightarrow$ $L = ww^R \quad w \in \{a, b\}^*$

$S = aSa \mid bSb \mid \wedge$

$S \rightarrow aSa$
$\rightarrow abSba$
$\rightarrow abba$

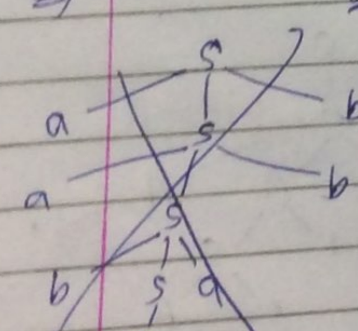
$\Rightarrow$ $L = wcw^R \quad w \in \{a, b, c\}^*$

$S = aSa \mid bSb \mid c$

L = all string having equal number of a & b

$\Rightarrow$ aabaabbb. Generate this String from gramm

$S = aSb \mid bSa \mid \wedge$



~~S = aaS / b / bbS~~

aabaab bb

109

$\Rightarrow$ baab ~~for~~ from $S \rightarrow aSb / bSa / \wedge / SS$

o L = all string having unequal number of a & b

7/Feb/2018.

$L = a^n b^m \quad n \geq m$

$S \rightarrow aSb / A$
$A \rightarrow aA / a$

$\Rightarrow L_1 = a^n b^{2m} \qquad L_2 = a^n b^{2m}$

$a^*(bb)^* \qquad S \rightarrow aSbb / \wedge$

$\Rightarrow L = a^n b^n c^m d^m$

$S \rightarrow AS \qquad m\, n > 0$
$A \rightarrow aAb / ab$
$B \rightarrow cBd / cd$

$\Rightarrow L = a^m b^{2m} c^n \qquad m, n \geq 0$

$S \rightarrow AC$
$A \rightarrow aAbb / \wedge$
$C \rightarrow cC / \wedge$

$\Rightarrow L = a^n b^{m+n} c^m = a^n b^n b^m c^m$

$L = a^n b^m c^k$ $\quad n = m \quad m \le k$

$L_1 = n = m \quad a^n b^n c^*$ $\quad L = L_1 \cup L_2$

$L_2 = m \le k \quad a^* b^m c^{m+n}$

$S = S_1 / S_2$

$S_1 \longrightarrow AC$

$A \rightarrow aAb / \wedge$

$C \longrightarrow cC / \wedge$

$S_2 \rightarrow BD$

$B \rightarrow aB / \wedge$

$D \longrightarrow aB / \wedge$

$L = \omega(a, b)$ where

no. of as in ω ≠ no. of bs in ω

$S \rightarrow aSb / bSA / SS / A / B / abS / Sab / Sb$

$A \rightarrow aA / a$

$B \rightarrow bB / b$

9/Feb/2018.

Closure properties of CFL

Let L_1 and L_2 be CFL

$L_1 : G_1 (S_1, \Sigma_1, V_1, P_1)$

$L_2 : G_2 (S_2, \Sigma, V_2, P_2)$

$L = L_1 \cup L_2$

$L_1 = a^n b^n c^m \quad m, n \ge 0$

$L_2 = a^n b^m c^m \quad m, n \ge 0$

$S \rightarrow S_1 / S_2$

$S_1 \longrightarrow AC$

$A \longrightarrow aAb / \wedge$

$C \longrightarrow cC / \wedge$

$S_2 \longrightarrow PB$

$P \longrightarrow aP / \wedge$

$B \longrightarrow bBc / \wedge$

Concatenation

Doubt
73 → 0 followed by 1
→ all string exactly one a. Pg 74
75 → aabb ans

$L = L_1 \cdot L_2$

$ab\,a^2b^2c\,a^3b^3c \quad b^4c^4\,bc\,abc^2$

Kleene Closure

If L_1 is context free, L_1^* will also be context free.

$L = L_1^*$ $\quad S \to SS_1 / \wedge$

$abc\,a^2b^2\,ab\,a^3b^3c$

Reversal (transpose)

$L = L_1^R$

$L_1 = a^n b^m c^m \quad n, m \geq 0$

$L = c^m b^m a^n \quad n, m \geq 0$

$S \to CA$

$A \to bAa / \wedge$

$C \to Cc / \wedge$

$G(V, \Sigma, P, S)$

$P = P^T$

⇒ $L = L_1 \cap L_2$

$L_1 = a^n b^m c^m d^k \quad n, m, k \geq 0$

$L_2 = a^n b^p c^q d^n \quad n, p, q \geq 0$

follow CFL

$a = d.$

$L = a^n b^m c^m d^n \quad m, n \geq 0$

⇒ $L = L_1 \cap L_2$

$L_1 = a^n b^n c^r$

$L_2 = a^m b^n c^n$

$L = L_1 \cap L_2$

not follow CFL

$\overline{\overline{L_1} \cup \overline{L_2}}$

Complement

$\overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2$

CFL are closed under complementation

$L_1 / \overline{L_1} \rightarrow CFL$

$L_2 / \overline{L_1} \rightarrow CFL$

$$\frac{\overline{L_1} \cup \overline{L_2} \rightarrow CFL}{\overline{L_1 \cup L_2}}$$

Simplification of CFL.

Elimination of useless symbol

Eg

$S \rightarrow BD$

$A \rightarrow aAB / c$	$B \rightarrow bB / L$	$S \rightarrow AB / \cancel{A}BC$
$C \rightarrow BC$	$E \rightarrow a/b$	$A \rightarrow aA / SC$
$D \rightarrow a$	Non Reachable	$D \rightarrow c / \wedge$
$B \rightarrow a$		$B \rightarrow bC$
		$A \rightarrow b/c$

Non Generating Non Terminal

Non Reachable

W - set of All non terminal that are generating.

$W_1 = \{A / A \rightarrow \alpha \text{ and } \alpha \in \Sigma^*\}$

$W_{i+1} = W_i \cup \{A / A \rightarrow \alpha \in P \text{ and } \alpha \in (\Sigma \cup W_i)^*\}$

$W_i = W_{i+1}$ $\quad V_N' = W_i$

P' = all production from W_i

Q. $C \to d$
$S \to AB/BC$
$A \to aA$
$C \to dC$
$B \to bC/SA$
$E \to b/C$

Q. $S \to AB/a$
$A \to BC/b$ $\quad W = \{S\ A\}$
$B \to ab/C \to$ Remove
$C \to aC/B \to$ Remove
Remove all containing A B & C
$\left.\begin{array}{l} S \to a \\ A \to b \end{array}\right\} \Rightarrow \boxed{S \to a}$

W. $\longrightarrow$ One who derives only terminal.

$W_1 = \{C, E\}$
$W_2 = \{E, C\} \cup \{B\} = \{ECB\}$
$W_3 = \{ECB\} \cup \{S\} = \{ECBS\}$

$C \to d$ ✓
$C \to dC$ ✓
$S \to BC$
$B \to bC$
$E \to b/C$

$\Rightarrow$ $S \to AB/CaD$
$A \to BG/Aa/b$
$B \to CB$
$C \to Ac/d$
$D \to SA/Hb$
$G \to a$
$H \to aG$

$W_1 = \{G\ A\ C\}$.

$W_2 = \{G\ A\ C\ H\}$
$W_3 = \{A, C, G, H, D, S\}$
$S \Rightarrow$
$A \to Aa$
$A \to b$
$C \to d$
$C \to Ac$
$G \to a$
$H = aG$.

* If S not included Grammar is empty *

12/feb/2018.

Eliminating Useless Symbol
→ Not Reachable

Method.

$W_1 = \{S\}$

$W_{i+1} = W_i \cup \{A \mid B \to A \text{ and } B \in W_i\}$

Repeat till $W_i = W_{i+1}$.

$V_N' = V_N \cap W_i$

$\Sigma' = \Sigma \cap W_i$

P' = include all production which have T and NT only from W_i

Q $S \to aAa$

$A \to Sb \mid bCC$

$C \to aC \mid b$

$E \to a \mid b$

$W_1 = \{S\}$

$W_2 = \{S\} \cup \{a, A\} = \{S, a, A\}$

$W_3 = \{S, a, A\} \cup \{b, C\} = \{S, a, A, b, C\}$

$W_4 = \{S, a, A, b, C\} \cup \{a, C, b\}$

$= \{S, a, A, b, C\}$

$V_N' = \{S, A, C, E\} \cap W_i$

$= \{S, A, C\}$

$\Sigma' = \{a, b\} \cap W_i = \{a, b\}$

P' = $S \to aAa$ $\quad C \to aC$

$A \to Sb$ $\quad C \to b$

$A \to bCC$

Q. $S \to Aa / f$ $\quad W_1 = \{S\}$

$A \to SbB$ $\quad W_2 = \{S\} \cup \{A a f\}$

$B \to bCC$ $\quad W_3 = \{S\} \cup \{Aaf\} \cup \{SbB\}$

$C \to eD / Ec / d$ $\quad W_4 = \{S\} \cup \{Aaf\} \cup \{SbB\} \cup \{bCC\}$

$E \to aA / bB$

$F \to bE / c$ $\quad \{S, C.$

Elimination of Null production

$\cancel{S \to aS / \wedge}$

$\cancel{S \to aS / a}$

$A \to aAb / \wedge$

W_i = find set of all nullable variable.

$A \to \wedge$ then add A to W_i

$W_1 = \{A / A \to \wedge \in P\}$

$W_{i+1} = W_i \cup \{B / B \to A$ and $A \in W_i\}$

$A \to \wedge$

$\times\ B \to aA$

$\times\ B \to AC$

$C \to b$

$B \to A$

$P' =$ if $A_1 \in W_i$

and $B \to A_1 A_2 \ldots A_n \in P$

$P' = B \to A_2 \ldots A_n / A_1 A_2 \ldots A_n$

= write new P including & excluding nullable variable

$S \to aS / AB$

• $A \to \wedge / aD$ $\quad W_1 = \{A, B\}$

$B \to \wedge / bDa.$ $\quad W_2 = \{A, B\} \cup \{S\} = \{ABS\}$

$D \to b$

$S \to aS / a$

$S \to AB / A / AB$

$A \to aD$

$B \to bDa$

$D \to b.$

$\Rightarrow$ $S \rightarrow ABC$

$A \rightarrow aAb / \wedge$

$B \rightarrow bC / CA$

$C \rightarrow aC / \wedge$

$W_1 = \{A\ C\}$

$W_2 = \{AC\} \cup \{B\}$

$W_3 = \{A\ C\} \cup \{B\} \cup \{S\}$

$S = ABC / AB / BC / AC / A / B / C$

$A \rightarrow aAb / ab$

$B \rightarrow bC / b / C / A / CA$

$C \rightarrow aC / a$

$\Rightarrow$ **Elimination of unit Production**

$$\left.\begin{array}{l} A \rightarrow B \\ B \rightarrow C \\ C \rightarrow a \end{array}\right\} \begin{array}{l} A \rightarrow a \\ B \rightarrow a \\ C \rightarrow a \end{array}$$

<u>Method</u>

construction of $W_i(A)$ set defining at NT derived from A. unit production form.

$W_1(A) = A$

$W_{i+1}(A) = W_i(A) \cup \{B / A \rightarrow B \ \& \ B \in W_i(A)\}$

if $B \in W_i(A)$ and $B \rightarrow q \in P$ then

$A \rightarrow q \in P'$

$$\left.\begin{array}{l} A \rightarrow B \\ B \rightarrow q \end{array}\right\} \begin{array}{l} A \rightarrow q \\ B \rightarrow q \end{array}$$

$S \rightarrow AB$ $W_i(S) = \{S\}$
$A \rightarrow a$ $W_i(A) = \{A\}$
$B \rightarrow C / b$ $W_i(B) = \{B, C, D, E\}$
$C \rightarrow D$ $W_i(C) = \{C, D, E\}$
$D \rightarrow E / bc$ $W_i(D) = \{D, E\}$
$E \rightarrow a.$ $W_i(E) = \{E\}$

Start Writing from down.

$S \rightarrow AB$
$A \rightarrow a$
$B \rightarrow b$
$E \rightarrow a$
$D \rightarrow a / bC$
$C \rightarrow a / bC$
$B \rightarrow a / bC$

$A \rightarrow BC / B$
$B \rightarrow e / bd$
$C \rightarrow aC / d$

$A \rightarrow BC / e / bd$
$B \rightarrow e / bd$
$C \rightarrow eC / d$

Null
↓
Unit
↓
Useless — Non Generating
Useless — Non Reachable.

12/Feb/2018

Non-Reachable

$W_1 = \{S\}$

$W_{i+1} = W_i \cup \{A \mid B \to A \text{ and } B \in W_i\}$

$V_N' = V_N \cap W_i \quad \Sigma' = \Sigma \cap W_i$

P' = all production P that include $\{V_i \in W\}$

$S \to aA \mid b$
$A \to aCb \mid e$
$C \to Be$
$B \to cB \mid d$
$E \to a \mid b$

$S \to aA \mid b$
$A \to aCb \mid e$
$C \to Be$
$B \to cB \mid d$

$W_1 = \{S\}$
$W_2 = \{S\} \cup \{A, a, b\}$
$W_3 = \{S, A, a, b, e, C\}$
$W_4 = \{S, A, a, b, e, C, B\}$
$W_5 = \{S, A, a, b, e, C, B, c, d\}$

$V_N' = V_N \cap W_i$
$= \{S, A, B, C, E\} \cap W_i$
$= \{S, A, B, C\}$

$\Sigma' = \Sigma \cap W_i$
$= \{a, b, c, d, e\} \cap W_i$
$= \{a, b, c, d, e\}$

Date
Page

12/Feb/2018.

Non-Reachable

$W_1 = \{S\}$

$W_{i+1} = W_i \cup \{A \mid B \to A \text{ and } B \in W_i\}$

$V_N' = V_N \cap W_i \quad \Sigma' = \Sigma \cap W_i$

P' = all production P that include $\{V_n \in W$

$S \to aA/b$

$A \to aCb/e$

$C \to Be$

$B \to cB/d$

$E \to a/b.$

$S \to aA/b$

$A \to aCb/e$

$C \to Be$

$B \to cB/d.$

$W_1 = \{S\}$

$W_2 = \{S\} \cup \{A, a, b\}$

$W_3 = \{S, A, a, b, e, C\}$

$W_4 = \{S, A, a, b, e, c, B\}$

$W_5 = \{S, A, a, b, e, C, B, c, d\}$

$V_N' = V_N \cap W_i$

$= \{S, A, B, C, E\} \cap W_i$

$= \{S, A, B, C\}$

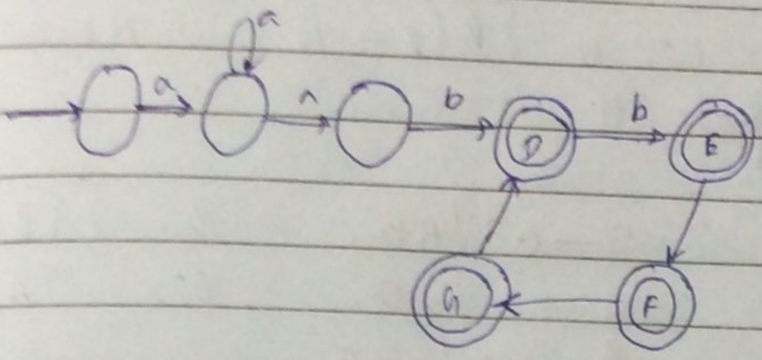
$\Sigma' = \Sigma' \cap W_i$

$= \{a, b, c, d, e\} \cap W_i$

$= \{a, b, c, d, e\}$

Make context free grammar

→ $L = a^m b^n$ $m > 0$ n is not multiple of 5.



$A \to aA \,/\, aC$

$C \to bD \,/\, b$

$D \to bE \,/\, b$

$E \to bF \,/\, b$

$F \to bG \,/\, b$

$G \to bC$

→ Every opening bracket have a closing bracket

$S \to e \,/\, SS \,/\, [S]$

$G = (V, \Sigma, R, S)$ with set of variable $V = \{S\}$, where S is the start variable, set of terminals $\Sigma = \{[,]\}$; and rules

→ Find CFG that generate the language

$L(G) = \{ a^n b^m \mid 0 \le n \le m \le 2n \}$

$S \to aSb \,/\, aSbb \,/\, \wedge$

Elimination of NULL Production

W_i = Set of all nullable variable

$A \to \wedge \longrightarrow$ add A to W_i

$W_{i+1} = W_i \cup \{B \mid B \to A \text{ and } A \in W_i\}$

Repeat till $W_i = W_{i+1}$

①

$S \to aS / AB$
$A \to \wedge / aD$
$B \to \wedge / bDa$
$D \to b$

$W_1 = \{A B\}$
$W_2 = \{A, B, S\}$

$S \longrightarrow aS / A / AB / B$
$A \longrightarrow aD$
$B \longrightarrow bDa$
$D \longrightarrow b$

$S \to ABC$
$A \to aAb / \wedge$
$B \to bc / CA$
$C \to aC / \wedge$

$W_1 = \{A, C\}$
$W_2 = \{A, C\} \cup \{B\}$
$W_3 = \{A, C, B, S\}$

$S \to ABC / AB / BC / CA / A / B / C$
$A \to aAB / ab$
$B \to bC / b$
$B \to CA / C / A$
$C \longrightarrow aC / a.$

~~Normal Form~~.

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

$Q \rightarrow$ Set of states
$q_0 \rightarrow$ initial state
$F \rightarrow$ set of final states
$\delta \rightarrow$ transition function
$\Sigma \rightarrow$ input symbol.

DFA → moves defined for every symbol.

Chomsky Normal Form (CNF)

Grammar G is in CNF if all the production are in following form:

2 non terminal

$A \to BC$ or

$A \to a$ 1 terminal.

Procedure

~~Simply~~

Simplify grammar 112 116 120 118

Removing intermediate terminals for each terminal.

Q

$S \to AB/a$

$A \to b/cD/eD/d$

$B \to cD/eD/dC$

$C \to eDdB/a$

$D \to g/c/b$

$X \to c$ $Y \to e$ $Z \to d$

$S \to AB/a$

$A \to b$

~~$A \to cD$~~

$A \to XD$

$A \to YD$

$A \to d$

$B \to XD$

$B \to YD$

$B \to ZC$

$C \to YDZB$

$C \to a$

$D \to g/c/b.$

Gribeach Normal Form (GNF)

A grammar G is in GNF of all the production are of form

$$A \rightarrow a q_1 q_2 \ldots q_n / a$$
$$a \in \Sigma$$
$$q_1, q_2, \ldots q_n \in NT$$

Start with T followed by any no. NT.

if $A \rightarrow Aq/\beta$

$$A \rightarrow \beta A'/\beta$$
$$A' \rightarrow qA'/q$$

Q $S \rightarrow CA/BB$
$B \rightarrow b/SB$
$C \rightarrow b$
$A \rightarrow a$

↓

Already in chomsky normal form.

STEP 1 Change the names of the Non-terminal Symbol into some A_i in ascending order of i.

$S \longrightarrow A_1$
$C \longrightarrow A_2$
$A \longrightarrow A_3$
$B \longrightarrow A_4$.

$A_1 \longrightarrow A_2A_3 / A_4A_4$
$A_4 \longrightarrow b / A_1A_4$
$A_2 \longrightarrow b$
$A_3 \longrightarrow a$

STEP 4. Alter the rules so that the Non-Terminals are in ascending order, s.t. if the production is of the form

$A_i \longrightarrow A_jX$, then
$i < j$ & should never be $\underbrace{i = j}_{\Downarrow \text{ removed later}}$

$A_1 \longrightarrow A_2A_3 / A_4A_4$ ✓ since $i < j$ Here.
$A_4 \longrightarrow b / A_1A_4$ $i = 4 \longrightarrow j \longrightarrow$ X
$A_2 \longrightarrow b$
$A_3 \longrightarrow a$.

$A_4 \longrightarrow b / A_2A_3A_4 / A_4A_4A_4 \longrightarrow$ X
$A_4 \longrightarrow b / bA_3A_4 / A_4A_4A_4$.
✓ ✓ X

~~A_4~~ This condition is left recurrsion.

Step Remove left Recursion

$A_4 \rightarrow b / bA_3A_4 / A_4A_4A_4$

~~A_4~~ $Z \rightarrow A_4A_4Z / A_4A_4$

$A_4 \rightarrow b / bA_3A_4 / bZ / bA_3A_4Z$

$A_1 \rightarrow bA_3 / bA_4 / bA_3A_4A_4 / bZA_4 / bA_3A_4ZA_4$
$A_4 \rightarrow b / bA_3A_4 / bZ / bA_3A_4Z$
$Z \rightarrow bA_4 / bA_3A_4A_4 / bZA_4 / bA_3A_4 / qZA_4 /$
$bA_4Z / bA_3A_4A_4Z / bZA_4Z / bA_3A_4ZA_4Z$

eg

$S \rightarrow AB / b$
$A \rightarrow CBx / gF$
$F \rightarrow a$
$B \rightarrow bB / C$

(have not removed C because we'r to again add it later on).

$S \rightarrow ABa$

$S \rightarrow b$

$A \rightarrow aaA$

$A \rightarrow B$

$B \rightarrow bBb$

$B \rightarrow \wedge$.

$X \rightarrow AB$

$Z \rightarrow a$.

$S \rightarrow XZ$

$X \rightarrow AB$

$Z \rightarrow a$

$S \rightarrow b$

$A \rightarrow \wedge$

b^{2n}.

~~$A \rightarrow BBb$~~ $A \rightarrow bQ$.

~~$B \rightarrow$~~ $\wedge$.

~~$B \rightarrow bBB$~~

$P \rightarrow bQ / \wedge$

$Q \rightarrow bP$

bbbb

$B \rightarrow \wedge$

$B \rightarrow bQ$

$Q \rightarrow bB$

$S \rightarrow XZ / b$

$X \rightarrow AB$

X

Z

A

B