```
int a;
a = 5;
```

C++

word → has an alphabet set, collectively called language

keyword ← int
identifier ← a
number ← 5

Language is a set which has collection of words. words dont depend on each other.
e.g {int, a, 5, ...}

Sentence / Syntax (different words merged together) + dependency is involved.

int a; // context free grammar

language ≠ sentence

collection of alphabets → word

collection of words → language

language = set → "distinct elements"

base condition — prior language — alphabet set.

$L1 = \{a, b\} \Rightarrow \Sigma = \{a, b\}$

$|L1| = n_x(L1) = 2$ — cardinality of set L1

$|x1| = 1$ — length of string x1

cardinality ≠ length

word can be of any length.

words { canonical order (ordered by word length i.e., 0, 1, 2 ...)
lexicographical order (same as dictionary)

$n^0 = 1$, similarly $x^0 = \Lambda$

? e.g $\Sigma = \{a, b\}$, $L = \{x \mid x \in \Sigma ; 0 \le |x| \le 2\}$ — words 0 1 2 — canonical order.

$L = \{\Lambda, a, b, aa, ab, ba, bb\}$

of c2dca

Theory of Automata :-

Instructor : Hamad Ul Qudous

→ Theory of Machines which help us in problem solving.

23/9/22

Traditional Central areas of theory of computation:

1. Automata
2. Computability → (classification as solvable/nonsolvable).
3. Complexity → e.g some problems are computationally hard (scheduling) & others are easy (sorting)
   (classification as easy/hard)

- Yes/No machine (valid or invalid)
  Application : compiler

→ finite automation (text processing, compilers, hardware design)

→ context free grammar (Prog. languages, AI)

mid I

→ regular language
- FSM (to identify it) — finite state machine (no memory attached)
- regular expression

→ context free language (rules based)
- syntax/sentence — (PDA - Push down automata OR finite automata with memory) — 1 stack

→ turing's machine (finite automata with 2 stacks).

Language / Set (group of welldefined distinct items) → elements/members

ID = {a, b ... z, A ... Z, _, 0 ... 9}

word req. ← 65

alphabet set ← $N = \{0, 1, 3, 4\}$ $M = \{5, 6\}$

language ← $L = \{x \mid x \in (N \cup M)^2\}$ → You've 2 choices Now.

NN NM MN MM — length is exactly 2 here.

65 ≠ 065

Intro to Computation by Michael Sipser

Domain : $\{0 \dots 6\} - \{2\}$

↓

set of possible inputs to function.

Range : outputs of function.

Onto function

↓

domain is fully utilized in range

subset ↓ every member of A is also a member of B. $A \subseteq B$	proper subset ↓ subset + A is not equal to B $A \subsetneq B$	multiset ↓ if we dont want to consider multiple repetitions of member, the set is multiset.

infinite set ↓ infinite elements $\{1,2,3,\dots\}$ $\{\dots,-1,0,1,\dots\}$	empty set ↓ $\emptyset$	singleton set ↓ 1 member	unordered pair ↓ set with 2 members

union ↓ $A \cup B$	intersection ↓ $A \cap B$	complement ↓ $\overline{A}$ $U - A$

25/8/22

alphabets are smallest component for making language.

representation for alphabet set

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

$L = \{x \mid x \in \Sigma^1 ;\ x \text{ ends with } b\} = \{b\}$

$|L| = 1$

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

$L = \{aab, bab, abb, bbb\}$

$\underset{\Sigma^2}{_}\ \underset{2^2 = 4}{_}\ \underset{}{b}$

$|L| = 4$

Q. $\Sigma = \{a, b, c\}$

$L = \{x \mid x \in \Sigma^3 ;\ x \text{ ends with } b\}$

$L = \{aab, abb, acb, bab, bbb, bcb, cab, cbb, ccb\}$

$\underset{\Sigma^2}{_}\ \underset{3^2 = 9}{_}\ \underset{}{b}$

$|L| = 9$

alphabet ^ locations

Q. $\Sigma = \{a, b, \dots, z\}$

$|L| = 26^2$

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

$L = \{x \mid x \in \Sigma^{0|1|2|3|4} ;\ x \text{ ends with } b\}$

$2 + 4 + 8 + 16$

30

lexographical order / Canonical order

↳ lengthwise

↗ dictionary

$\Sigma^0 b$	$\Sigma^0 = 0$	
$\Sigma^1 b$	$\Sigma^1 = 1$	$\{b\}$
$\Sigma^2 b$	$\Sigma^2 = 2$	$\{cb, bb\}$
$\Sigma^3 b$	$\Sigma^3 = 4$	$\{aab, abb, bcb, bbb\}$
$\Sigma^4 b$	$\Sigma^4 = 8$	$\{aaab, \dots, bbbb\}$

$|L| = 15$

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

00102030 4

$L = \{x \mid x \in \Sigma, x$ starts & ends with $b\}$

$\Sigma^0 = 0$

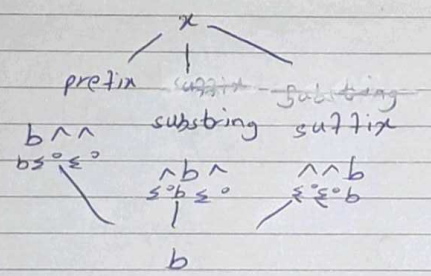
$\Sigma^1 = 1 \quad \{b\}$

$\Sigma^2 = 1 \quad \{bb\}$

$\Sigma^3 = 2 \quad \{bab, bbb\} \quad b\Sigma^1 b$

$\Sigma^4 = 4 \quad \{baab, babb, bbab, bbbb\} \quad b\Sigma^2 b$

$|L| = 8$



Q. $L = \{a, aa, aaa, \ldots\}$

$= \{x \mid x \in \Sigma^{0010203\infty}, x$ only has $a\}$

Q. $L = \{a, ab, aab, abb, aaaa\}$ This Q. seems wrong. Skip!

$= \{x \mid x \in \Sigma^{0010\ldots\infty}, x$ starts with $a\}$ ✗

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

$L = \{x \mid x \in \Sigma^{*}$ (kleene closure) ; $|L| = 12\}$

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots \Sigma^n$

✗ $L = \{a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab\}$

✓ $L = \{\wedge, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa\}$

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

$L = \{x \mid x \in \Sigma^*;\ |x| \le 4$ & $|L| \le 10$

x starts with a but doesnt ends with b$\}$

$L = L1 \cap L2$

$L = \{a, aa, aaa, aba, aaaa, aaba, abaa, abba\}$

$\Sigma^0 = 0$

$\Sigma^1 = 1 \quad \{a\}$

$\Sigma^2 = 1 \quad \{aa\}$

$\Sigma^3 = 2 \quad \{aaa, aba\}$

$\Sigma^4 = 4 \quad \{aaaa, aaba, abaa, abba\}$

$|x| > 4 \rightarrow \Sigma^5 = \quad \{aaaaa, aaaba\}$

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

$L = \{x \mid x \in \Sigma^*;\ x$ as a decimal digit & summation of alphabet is divisible by 2$\}$

$L = \{0, 00, 000, 110, 0000, 0110, 1100, 00000, 00110, 01100\}$

$\Sigma^0 = 0$

$\Sigma^1 = 1 \quad \{0\}$

$\Sigma^2 = 1 \quad \{00\}$

$\Sigma^3 = 2 \quad \{000, 110\}$

$\Sigma^4 = 3 \quad \{0000, 0110, 1100\}$

$\Sigma^5 = 3+ \quad \{00000, 00110, 01100$

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

$L = \{x \mid x \in \Sigma \; ; \; x \text{ is a palindrome} \; ; \; 0 \le |x| \le 3\}$

$= \{\wedge, a, b, aa, bb, aaa, aba, bab, bbb\}$

(palindrome + canonical) – valid words.

$|L| = 9$

Language

Base — prior Language

(above question) — us of set operations.

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

$L = \{x \mid x \in \Sigma \; ; \; x \text{ is an even palindrome} \; ; \; 0 \le |x| \le 3\}$

$L1 = \{x \mid x \in \Sigma \; ; \; x \text{ is even} \; ; \; 0 \le |x| \le 3\}$

$= \{\underbrace{\wedge}_{0}, \underbrace{aa, ab, ba, bb}_{2}\}$

$L2 = \{x \mid x \in \Sigma \; ; \; x \text{ is palindrome} \; ; \; 0 \le |x| \le 3\}$

$= \{\wedge, a, b, aa, bb, aaa, aba, bab, bbb\}$

$L = L1 \cap L2$

$= \{\wedge, aa, bb\}$

$|L| = 3$

Homework # 1.

Q1. $\Sigma = \{a, b\}$

alphabet ^ locations

a. $L_1 = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3$, $|L| = ?$

$= 2^0 + 2^1 + 2^2 + 2^3$

$= 15$

b. $L2 = \{w \text{ over } \Sigma \mid |w| > 5 \;\&\; |w| \le 10 \;\&\; |w| \text{ is odd}\}$

What is the cardinality of L2.

$|L_2| = 2^7 + 2^9$

$= 640$

c. $L3 = \{w \text{ over } \Sigma \mid |w| > 5 \;\&\; |w| \le 10 \;\&\; |w| \text{ is even}\}$

$|L_3| = ?$

$= 2^6 + 2^8 + 2^{10}$

$= 1344$

d. $L_4 = \{w \text{ over } \Sigma \mid |w| > 5 \;\&\; |w| \le 10\}$

$|L_4| = ?$

$= 2^6 + 2^7 + 2^8 + 2^9 + 2^{10}$

$= 1984$

e. $L_2 \cap L_3$ & $|L_2 \cap L_3| = ?$

$L_2 \cap L_3 = \{\}$

$|L_2 \cap L_3| = 0$

f. $L_2 \cap L_4$ & $|L_2 \cap L_4| = ?$

$L_2 \cap L_4 = L_2$

$|L_2 \cap L_4| = |L_2| = 640$

g. $L_3 \cap L_4$ & $|L_3 \cap L_4| = ?$

$= L_3$ $\quad = |L_3|$

$= 1344$

h. $L_2 - L_4$ & $|L_2 - L_4| = ?$

$= \{\}$ $\quad = 0$

i. $L_3 - L_4$ & $|L_3 - L_4| = ?$

$= \{\}$ $\quad = 0$

Q2. $\Sigma = \{0, 1, 2\}$

a. $L_1 = \{ s$ over $\Sigma \mid |s| \le 10$ & sum of digits in s are ≥ 5 & $\le 10 \}$

$L_{11} = \{ s$ over $\Sigma \mid |s| \le 10 \}$

$L_{12} = \{ s$ over $\Sigma \mid$ sum of digits in s are $\ge 5 \}$

$L_{13} = \{ s$ over $\Sigma \mid$ sum of digits in s are $\le 10 \}$

$L_1 = L_{11} \cap L_{12} \cap L_{13}$

b. $L_2 = \{ s$ over $\Sigma \mid |s| \le 15$ & sum of digits in s is divisible by $2 \}$

$L_{21} = \{ s$ over $\Sigma \mid |s| \le 15 \}$

$L_{22} = \{ s$ over $\Sigma \mid$ sum of digits in s is divisible by $2 \}$

$L_2 = L_{21} \cap L_{22}$

c. $L_3 = \{ s$ over $\Sigma \mid |s| \le 15$ & sum of digits in s is divisible by $3 \}$

$L_{31} = \{ s$ over $\Sigma \mid |s| \le 15 \}$

$L_{32} = \{ s$ over $\Sigma \mid$ sum of digits in s is divisible by $3 \}$

$L_3 = L_{31} \cap L_{32}$

d. $L_4 = \{ s$ over $\Sigma \mid |s| \le 15$ & sum of digits in s is divisible by 2 or divisible by $3 \}$

$= L_2 \cup L_3$

e. $L_5 = \{ s$ over $\Sigma \mid |s| \le 15$ & sum of digits in s is divisible by 2 and divisible by $3 \}$

$= L_2 \cap L_3$

f. $L_6 = \{ s$ over $\Sigma \mid |s| \le 15$ & sum of digits in s is divisible by 2 but not divisible by $3 \}$

$= L_2 - L_3$,

$= L_2 \cap L_3'$

g. $L_7 = \{ s$ over $\Sigma \mid |s| \le 15$ and sum of digits in s is divisible by 3 but not divisible by $2 \}$

$= L_3 - L_2$,

$= L_3 \cap L_2'$

h. $L_8 = \{ s$ over $\Sigma - \{2\} \mid |s| \le 15$ and sum of digits in s is divisible by $3 \}$

$L_{81} = \{ s$ over $\{2\} \mid |s| \le 15 \}$

$L_{82} = \{ s$ over $\{2\} \mid$ sum of digits in s is divisible by $3 \}$

$L_8 = L_3 - (L_{81} \cap L_{82})$

i. $L_9 = \{ s$ over $\Sigma - \{2\} \mid |s| \le 15$ and sum of digits in s is divisible $2 \}$

$L_{91} = \{ s$ over $\{2\} \mid |s| \le 15 \}$

$L_{92} = \{ s$ over $\{2\} \mid$ sum of digits in s is divisible by $2 \}$

$L_9 = L_2 - (L_{91} \cap L_{92})$

$\longleftrightarrow$

Ch2 John C Martin

Introduction to finite automata: FA → DFA, NFA, NFA^

(directed graph

Representation :

- alphabet set (Σ)
- states (Q)
- start state (s or q_0) symbol:- → O
- set of final states (f_n) – 0 to n
- transition function (δ_{DFA}) – action

$\underbrace{\Sigma \times Q}_{\text{input/domain}} \rightarrow \underbrace{Q}_{\text{output/range}}$

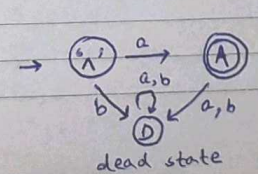
(cartesian product)

cartesian product of alphabet set with states = DFA (deterministic finite automata)

e.g/ $\Sigma = \{a, b\}$ $\quad Q = \{\wedge, A\}$, $s = \text{'}\wedge\text{'}$

$L = \{a\}$ $\quad \Sigma \times Q \rightarrow Q$

Accept/ Reject



dead state

$a, \wedge \rightarrow A$ $\quad a, A \rightarrow D$ $\quad a, D \rightarrow D$

$b, \wedge \rightarrow D$ $\quad b, A \rightarrow D$ $\quad b, D \rightarrow D$

$\boxed{\text{double circle = final state}}$

30/8/22

x over Σ

canonical ~~lexicographical~~ order.

Q. $L = \{x \mid x \in \Sigma^* ;\ x$ starts with 'a'$\}$

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

✗ $L = \{a, aa, aaa, aaaa, aaaaa, aaaaaa, aaaaaaa, aaaaaaaa, aaaaaaaaa, aaaaaaaaaa\}$

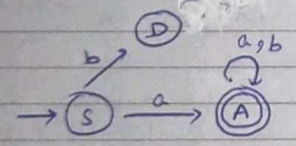
✓ $L = \{a, aa, ab, aaa, aab, aba, abb, aaaa, \ldots\}$

Q. $L = \{x \mid x \in \Sigma^* ;\ x$ contains at most one 'a'$\}$

$L = \{\wedge, a, b, ab, ba, bb, abb, bab, bba, bbb, \ldots\}$

$\longleftrightarrow$

$L = \{a, aa, ab, aaa, aab, aba, abb, \ldots\}$



finite automata processes string character by character.

Q	Σ	Qnew
S	a	A
A	b	A
	b	
	b	(A) ∈ f_n

$Q = \{S, A\}$

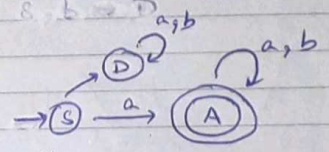
$\Sigma = \{a, b\}$

$Q \times \Sigma \rightarrow Q$

Domain		Range
S, a	→	A
S, b	→	D (trap/dead state)
A, a	→	A
A, b	→	A

Its FA. Not DFA because there's a transition which doesn't hold. New state is added i.e., D.

DFA Conversion :-



DFA is checked via cartesian product.

Additionally

D, a → D

D, b → D

$\longleftrightarrow$

$L = \{x \mid x \in \Sigma^* ;\ x$ contains atmost 1 'a'$\}$

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

$L = \{\wedge, a, b, ab, ba, bb, abb, bab, bba, bbb, abbb, babb, bbab, bbba, bbbb, \ldots\}$

S, a → A

S, b → B

A, a → D

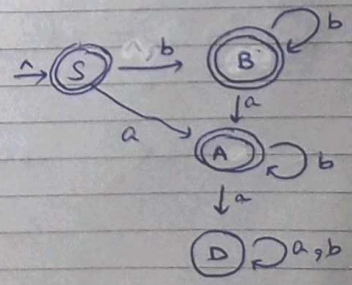
A, b → A

B, a → A

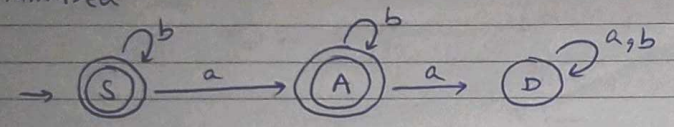
B, b → B

D, a → D

D, b → D



Minimized



Trace Tree

root is made ~~x~~ via start state

S	S
|a	|a
A	A
|b	|b
A	A
|b	|b
$\boxed{A} \in f_n$	$A \xrightarrow{a} D \notin f_n$
accepted	not accepted

"DFA" (ends with case)

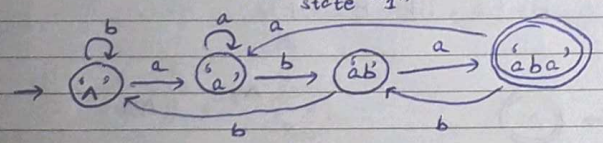
Q. $\Sigma = \{a,b\}$

$L = \{x \mid x \in \Sigma;\ x \text{ ends with 'aba'}\}$

$= \{\underbrace{aba}_{3}, \underbrace{aaba, baba}_{4}, \underbrace{aaaba, ababa, baaba, bbaba}_{5}\}$

$\delta_{DFA} = Q \times \Sigma \to Q$

state "1"



$\wedge \xrightarrow[b]{a}$ $\quad a \xrightarrow[b]{a}$ $\quad ab \xrightarrow[b]{a}$ $\quad aba \xrightarrow[b]{a}$

Prefix (sequence)	suffix (strings at end)
aba	a
^	^
a	a
ab	
aba	

Prefix	suffix
	b
^ — stay here	^
a	b
ab	
aba	

Prefix	Suffix	
	aa	ab
^	^	^
a	a	b
ab	aa	ab
aba		

Prefix	Suffix
	aba
^	^
a	a
ab	ab
aba	aba

Prefix	Suffix
	abb
^	^
a	b
ab	bb
aba	abb

Suffix starts from right side.

Prefix	Suffix	
	bab	baa
^	^	^ → assuming that it ends with '^'
a	b	a
ab	ab	aa
aba	bab	baa

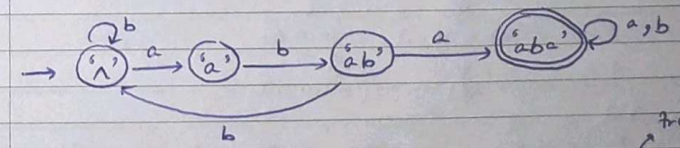
String processing in finite automata

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

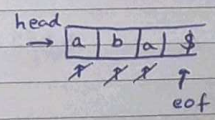
$L = \{x \mid x \in \Sigma;\ x \text{ contains 'aba'}\}$

$aba.. \Sigma^*$
$= \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots$

$\Sigma = (a \cup b)$



Input tape : (Unidirectional)



from input tape

Q	Σ	Action
'^'	a	'a'
'a'	b	'ab'
'ab'	a	'aba' => final

Accepted ✓

end goal : when the tape ends, the finite automata is in acceptance state.
else, it's rejected state.

DFA for language starting with 'aba'

$L = \{x \mid x$ belongs to S ; x starts with 'aba' $\}$

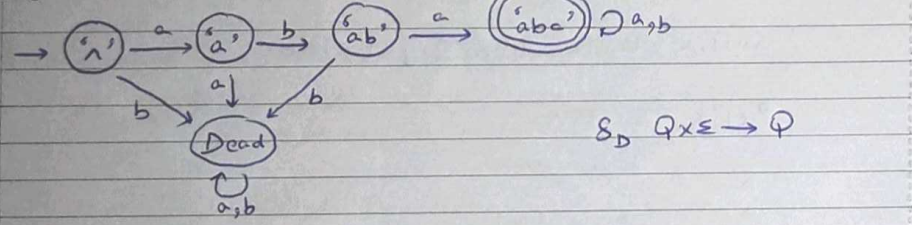
$\Sigma / S = \{a, b\}$

$L = \{\underset{3}{aba}, \underset{(aba)\Sigma'}{abaa, abab}\}$

x — prefix, substring, suffix

prefix → aba

sequence transition ✓



$\delta_D \quad Q \times \Sigma \rightarrow Q$

Closure property of DFA

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

$L = \{x \mid x \in \Sigma^* ;\ x$ & Σ of x is divisible by 2$\}$

x → as a digit → 'M1'

Σ of x → sum of # of elements → 'M2'

$= \{0, 10$

↳ as # ✓

↳ as sum ✗

$1 \rightarrow 0 \Rightarrow 1$ ✗

Closure property of FA

$M1 \times M2 \rightarrow M$

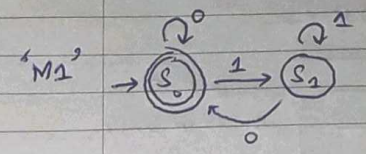
$= M1 \cap M2$

digits

$L1 = \{0, 00, 10, \ldots\}$ ends with '0' here.

$0 \% 2 = 0$ ✓

'M1'



$S_0 \rightarrow$ even

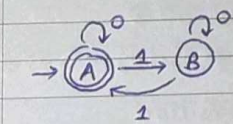
$S_1 \rightarrow$ odd

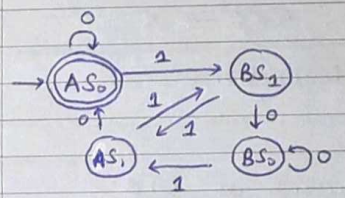
$S_0 < {}^{0}_{1}$ $\quad S_1 < {}^{0}_{1}$

Summation

$L2 = \{0, 00, 11, \ldots\}$ $\quad 1+1 = 2 \% 2 = 0$ ✓

'M2'





$A < {}^{0}_{1} {}^{A}_{B}$ $\quad B < {}^{0}_{1} {}^{B}_{A}$

$S_0 < {}^{0}_{1} {}^{S_0}_{S_1}$ $\quad S_1 < {}^{0}_{1} {}^{S_0}_{S_1}$

final state ?

$M_1 \cap M_2 = M$

AS_0 ✓ $\quad$ '110' ✓ e.g $\quad$ '11' ✗

$M_1 \cup M_2 = M$

AS_0, AS_1, BS_0 ✓

Types of FA

DFA, NFA, NFA λ

FA representation : 5 Tuples

$\{\Sigma, Q, S, f_n, \delta\}$

$f_n \leq Q$

doesnt change

1) δ_{DFA} $\quad Q \times \Sigma \rightarrow Q$

$Q = \{A, B\}$
$\Sigma = \{0, 1\}$

$A, 0 \rightarrow$ ___
$A, 1 \rightarrow$ ___
$B, 0 \rightarrow$ ___
$B, 1 \rightarrow$ ___

} state $\in Q$, 1 state only

$B, 0 \rightarrow A, B$ ✗
Multiple output states not accepted in DFA.

2) δ_{NFA} $\quad Q \times \Sigma = 2^Q$

2^Q → power set of Q

Power set of Q
$= \{ \{\}, \{A\}, \{B\}, \{A, B\} \}$ $\quad 2^2 = 4$

$\{\}$ → no transition
i.e.,
$A \times 0 \rightarrow$ ✗

3) $\delta_{NFA\wedge}$ You can view null transitions even $\wedge \notin$ "Σ"

Taking complement of DFA

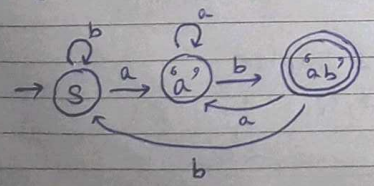
first, create DFA. Then,
Using closure property
Final states = nonfinal states
nonfinal states = final states

1/9 Q. Construct a DFA for following language

$L = \{ \underbrace{x \mid x \in \Sigma^*}_{x \text{ over } \Sigma} ;\ x \text{ ends with 'ab'} \}$ $\quad \Sigma = \{a, b\}$

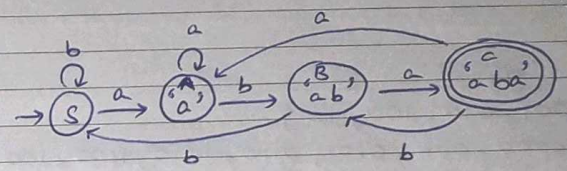
$|x_{pattern}| = 2$
optimal machine $= 2+1 = 3$.

$= \{ ab, aab, bab, aaab, abab, baab, bbab, \ldots \}$



Q. Construct a DFA for following language.

$L = \{ x \mid x \in \Sigma^* ;\ x \text{ ends with 'aba'} \}$ $\quad \Sigma = \{a, b\}$



$\Sigma^* aba$

$= \{ aba, aaba, baba, aaaba, abaaba, baaba, bbaaba, \ldots \}$

pattern (prefix)	aba
S	^
A	a
B	ab
C	aba

string (suffix)	a	b	aa	ab	aba	abb	abaa	abab
	^	^	^	^	^	^	^	^
	a	b	a	b	a	b	a	b
			aa	ab	ba	bb	aa	ab
					aba	abb	baa	bab
							abaa	abab

S: ^ → a, b
A: a → aa, ab
B: ab → aba, abb
C: aba → abaa, abab

Consider state with bigger length.

Quiz 1

Q. $\Sigma = \{0, 1\}$
$L = \{ w \text{ over } \Sigma ;\ w \text{ doesnt end on 1 and number of zeroes is divisible by 2}, |w| < 5 \}$

$L1 = \{ w \text{ over } \Sigma ;\ w \text{ doesnt end on 1}, |w| < 5 \}$
$= \{ \wedge, 0, 00, 10, 000, 010, 100, 110, 0000, 0010, 0100, 0110, 1000, 1010, 1100, 1110 \}$

$\Sigma^0 = 1$ $\quad \{\wedge\}$ $\quad \Sigma^2 = 2$ $\quad \{00, 10\}$
$\Sigma^1 = 1$ $\quad \{0\}$ $\quad \Sigma^3 = 4$ $\quad \{000, 010, 100, 110\}$
$\Sigma^4 = 8$ $\quad \{0000, 0010, 0100, 0110, 1000, 1010, 1100, 1110\}$

$L2 = \{w \text{ over } \Sigma;$ number of zeroes is divisible by $2; |w| < 5\}$

$= \{\wedge, 1, 00, 11, 001, 010, 100, 111, 0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111\}$

$\Sigma^0 = 1 \quad \{\wedge\}$

$\Sigma^1 = 1 \quad \{1\}$

$\Sigma^2 = 2 \quad \{00, 11\}$

$\Sigma^3 = 4 \quad \{001, 010, 100, 111\}$

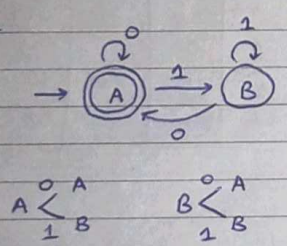
$\Sigma^4 = 8 \quad \{0000, 0011, 0101, 0110, 1010, 1100, 1001, 1111\}$

$L = L1 \cap L2$

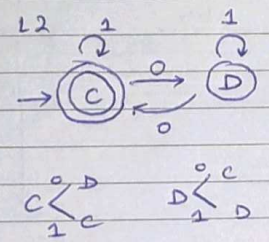
$= \{\wedge, 00, 010, 100, 0000, 0110, 1010, 1100\}$

$|L| = 8$

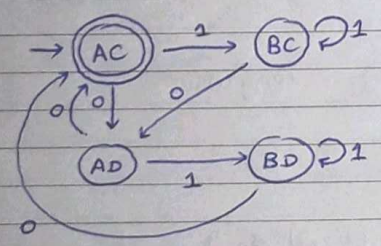
L1



L2



L



Closure property of DFA.

automaton simulator.com (without spaces).

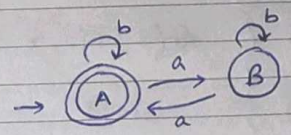
6/9

$L = \{x \mid x \in \Sigma^*;$ x contain even number of a's$\}$

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

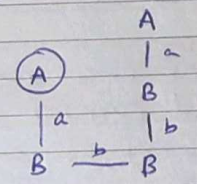
$L1 = \{\underbrace{\wedge}_{\Sigma^0}, \underbrace{b}_{\Sigma^1}, \underbrace{aa, bb}_{\Sigma^2}, \underbrace{aab, aba, baa, bbb}_{\Sigma^3}, \ldots$

M1



$x = abba$

a) Trace tree



Accepted!

b) Extended Transition Function

$\delta^* \quad Q \times \Sigma^* \rightarrow Q$

string

we know that, $\delta \quad Q \times \Sigma \rightarrow Q$

character

$\delta^*(q_0, x_a) <$ chop $\boxed{\delta(\delta^*(q_0, x), a)}$

$\delta(\delta(\delta^*(q_0, \wedge), x), a)$

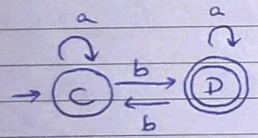
$\delta^*(\underline{start, abab})$	$\delta(\delta(\delta(\delta(start,a),b),a),b)$
$\delta(\delta^*(\underline{start, aba}),b)$	$\delta(\delta(\delta(S_0,b),a),b)$
$\delta(\delta(\delta^*(\underline{start, ab}),a),b)$	$\delta(\delta(S_0,a),b)$
$\delta(\delta(\delta(\delta^*(\underline{start, a}),b),a),b)$	$\delta(start,b)$
$\delta(\delta(\delta(\delta(\delta^*(\underline{start, \wedge}),a),b),a),b)$	start

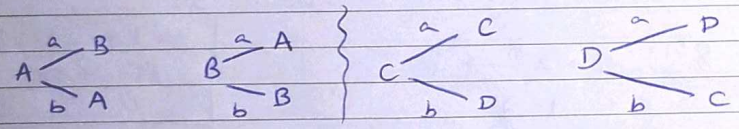
$\longleftrightarrow$

$L2 = \{x \mid x \in \Sigma^*$; x contains even number of a's & odd number of b's$\}$

$= \{b, ab, ba, aab, aba, baa, bbb, \ldots\}$

M2





Combining M1 & M2 : (closure property)

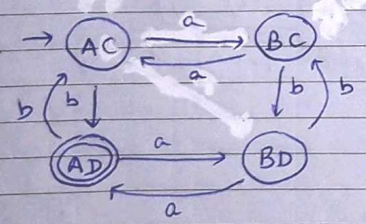
AND case

$f_n = \{AD\} = f_{n1} \times f_{n2}$

$f_n = \{AC, AD, BD\}$

OR case

State Digram



$L = L1 \cap L2 = \{b, aab, aba, baa, \ldots\}$

8/9

minus = start state
plus = final state

Transition Table

M1

Q	a	b
$\mp A$	B	A
B	A	B

M2

Q	a	b
$-C$	C	D
$+D$	D	C

δ

$Q \times \Sigma \rightarrow Q$

$M : M1 \cap M2$

$(Q_1 \times Q_2) \times \Sigma \longrightarrow Q$

$Q_1 = \{A, B\}$

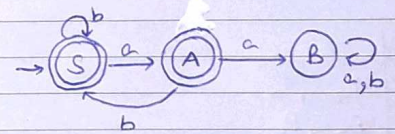
$Q_2 \{C, D\}$

$Q_1 \times Q_2 = \{AC, AD, BC, BD\}$

	a	b
$-AC$	BC	AD
BC	AC	BD
$+AD$	BD	AC
BD	AD	BC

Q. $L = \{x \mid x \in \Sigma^*$; x doesnt contains 'aa'$\}$, $\Sigma = \{a, b\}$

$= \{\wedge, a, b, ab, ba, bb, aba, bba, bab, abb, \ldots\}$

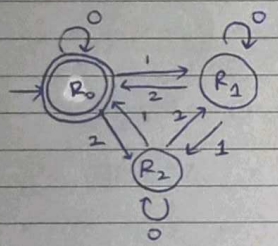


Q. $L = \{ x \mid x \in \Sigma^* ;$ x contains decimal numbers divisible by 3 & 5 & sum of alphabets are also divisible by 2 $\}$

e.g. 12210

$\Sigma = \{0, 1, 2\}$

M1 $L1 = \{ 0, 12, 21, 012, 021, 120, 210, \ldots \}$

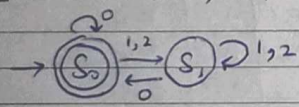


$R_0 + 0 = R_0$
$R_0 + 1 = R_1$
$R_0 + 2 = R_2$

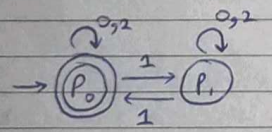
$R_1 + 0 = R_1$
$R_1 + 1 = R_2$
$R_1 + 2 = R_3$

$R_2 + 0 = R_2$
$R_2 + 1 = R_0$
$R_2 + 2 = R_1$

M2 $L2 = \{ 0, 00, 10, 20, \ldots \}$



M3



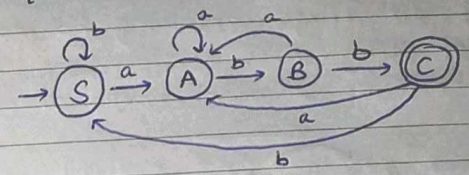
$f_n = \{R_0 S_0 P_0\}$

	0	1	2
$R_0 S_0 P_0$	$R_0 S_0 P_0$	$R_1 S_1 P_1$	$R_2 S_1 P_0$
$R_1 S_1 P_1$	$R_1 S_0 P_1$	$R_2 S_1 P_0$	$R_0 S_1 P_1$
$R_2 S_1 P_0$	$R_2 S_0 P_0$	$R_0 S_1 P_1$	$R_1 S_1 P_0$
$R_1 S_0 P_1$	$R_1 S_0 P_1$	$R_2 S_1 P_0$	$R_0 S_1 P_1$
$R_0 S_1 P_1$	$R_0 S_0 P_1$	$R_1 S_1 P_0$	$R_2 S_1 P_1$
$R_2 S_0 P_0$	$R_2 S_0 P_0$	$R_0 S_1 P_1$	$R_1 S_1 P_0$
$R_1 S_1 P_0$	$R_1 S_0 P_0$	$R_2 S_1 P_1$	$R_0 S_1 P_0$
$R_0 S_0 P_1$	$R_0 S_0 P_1$	$R_1 S_1 P_0$	$R_2 S_1 P_1$
$R_1 S_0 P_0$	$R_1 S_0 P_0$	$R_2 S_1 P_1$	$R_0 S_1 P_0$
$R_2 S_1 P_1$	$R_2 S_0 P_1$	$R_0 S_1 P_0$	$R_1 S_1 P_1$
$R_0 S_1 P_0$	$R_0 S_0 P_0$	$R_1 S_1 P_1$	$R_2 S_1 P_0$
$R_2 S_0 P_1$	$R_2 S_0 P_1$	$R_0 S_1 P_0$	$R_1 S_1 P_1$

Homework 2 - Part1

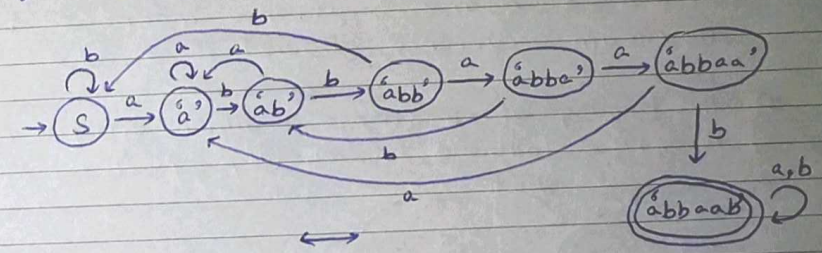
Draw DFA for following languages.

1) $L = \{ w \in \{a,b\}^* \mid w$ ends with abb $\}$

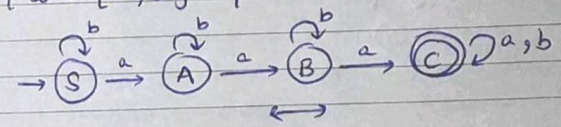


$= \{ abb, aabb, babb, aaabb, ababb, baabb, bbabb, \ldots \}$

2) $L = \{ w \in \{a,b\}^* \mid w$ contain substring abbaab

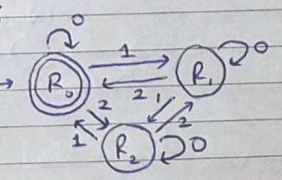


3) $L = \{ w \in \{a,b\}^* \mid w$ contains atleast three a's $\}$



4) $L = \{ w \in \{0,1,2\}^* \mid$ sum of digits in w are divisible by 3 $\}$

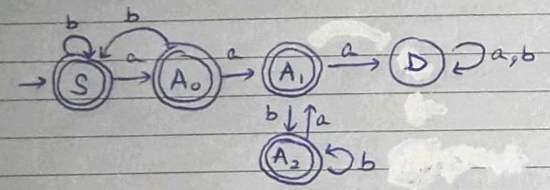
(R : remainder)



5) $L = \{w \in \{0,1,2\}^* \mid$ The number is divisible by 3 $\}$

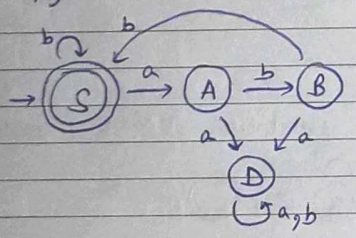
Same as Q4

6) The language of all strings containing no more than one occurrence of string aa.
(The string aaa contains 2 occurrences of aa)



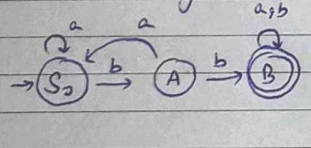
No more than one occurrence means that maximum occurrence allowed is 1.

7) The language of all strings in which every 'a' (if there is any) is followed immediately by 'bb'.



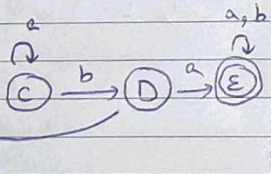
8) The language of all strings containing both bb & aba as substrings.

M1



	a	b
S_0	S_0	A
A	S_0	B
B	B	B

M2



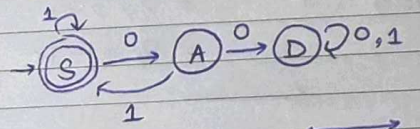
	a	b
S_1	C	S_1
C	C	D
D	E	S_1
E	E	E

$M = M1 \cap M2$
$= 3 \times 4 = 12$ states.

9) The language of all strings containing both aba and bab as substrings.

M1 = containing aba (4 states)
M2 = containing bab (4 states)
$M = M1 \cap M2$
$= 4 \times 4 = 16$ states.

10) $L = \{w \in \{0,1\}^* \mid$ every 0 in w is followed by atleast one 1$\}$



Homework 2 - Part 2

Draw DFA using closure properties of regular languages.

1) $L = \{w \in \{0,1,2,3\}^* \mid$ sum of digits in w are either divisible by 4 or 6$\}$

4 states, 6 states

Total $= 4 \times 6 = 24$ states.

Final states = M_1 or M_2 or both.

2) $L = \{w \in \{0,1,2,3,4,5,6,7,8,9\}^* \mid$ w is divisible by 5 and 6$\}$

2 states, 6 states

Total $= 2 \times 6 = 12$ states

$M = M1 \cap M2$

3) $L = \{ w \in \{a,b\}^* \mid w$ contains odd number of a's and even number b's $\}$

$\downarrow$ M1 — 2 states

$\downarrow$ M2 — 2 states

$M = M1 \cap M2$
$= 2 \times 2 = 4$ states

4) $L = \{ w \in \{a,b,c\}^* \mid n_a(w)$ and $n_b(w)$ are even but $n_c(w)$ are odd. $\}$

2 $M1 = n_a(w)$ is even
2 $M2 = n_b(w)$ is even
2 $M3 = n_c(w)$ is odd

$M = M1 \cap M2 \cap M3$
$= 2 \times 2 \times 2$
$= 8$ states.

5) $L = \{ w \in \Sigma^* \mid w$ contains the substring 010, but does not contain the substring 0101$\}$

$M1 =$ contains 010 $\rightarrow$ 4 states
$M2 = !$ (contains 0101) $\rightarrow$ 5 states
$\downarrow$
take complement of states

$M = M_1 \cap M_2 = 4 \times 5 = 20$ states.

DFA
$Q \times \Sigma \rightarrow Q$

13/9

NFA
$\downarrow$
Nondeterministic finite Automata.

Tuples (same as DFA other than transition function)

5
- Σ
- Q
- q_0
- $\delta_n = \{\}$
- δ_{NFA}

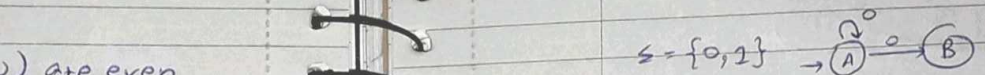
2^Q = Maximum DFA states

$Q \times \Sigma \rightarrow P(Q)$
power set of Q

State Diagram

$Q = \{A, B\}$
$P(Q) = \{ \phi, A, B, \{A,B\} \}$
$\downarrow$
- can skip transition (skip link)
- can have multiple transition (multi state)
- can have normal transition (single state)

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



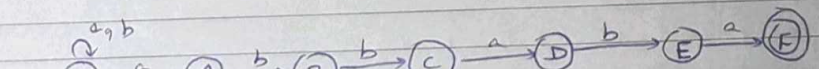
NFA is easy to design.

Today's lecture:
1) NFA
2) Subset Construction Method
NFA-DFA

$L = \{ x \mid x \in \Sigma^*;\ x$ ends with 'abbaba' $\}$

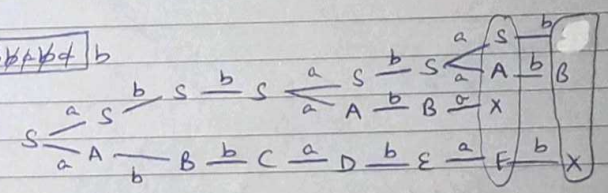
$= \{$ abbaba, a abbaba, b —, aa —, ab —, ba —, bb —, ... $\}$
Σ^0 Σ^1 Σ^2 ...

Σ^* abbaba



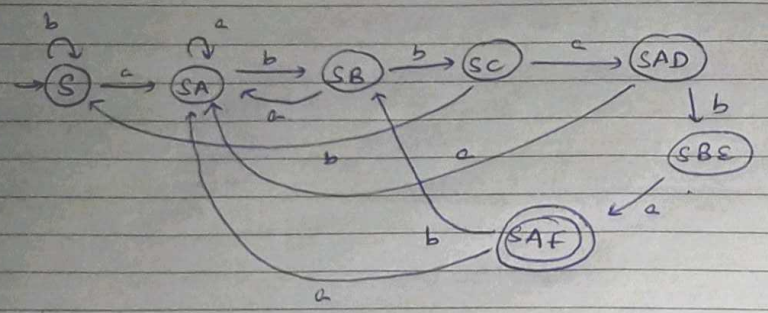
Input $\boxed{\cancel{a}\cancel{b}\cancel{b}\cancel{a}\cancel{b}\cancel{a}}$ b

Trace Tree



If any 1 state of longest link is final, then the string is acceptable

$Q = \{S, A, B, C, D, E, F\}$



		a	b
$P(Q)$	S	SA	S
	SA	SA	SB
	SB	SA	SC
	SC	SAD	S
	SAD	SA	SBE
	SBE	SAF	SC
	SAF	SA	SB

$S \xrightarrow{a} SA$, $S \xrightarrow{b} S$

$A \xrightarrow{a} \times$, $A \xrightarrow{b} B$

$B \xrightarrow{a} \times$, $B \xrightarrow{b} C$

$C \xrightarrow{a} D$, $C \xrightarrow{b} \times$

$D \xrightarrow{a} \times$, $D \xrightarrow{b} E$

$E \xrightarrow{a} F$, $E \xrightarrow{b} \times$

$F \xrightarrow{a} \times$, $F \xrightarrow{b} \times$

every state which has 'f' is final state.

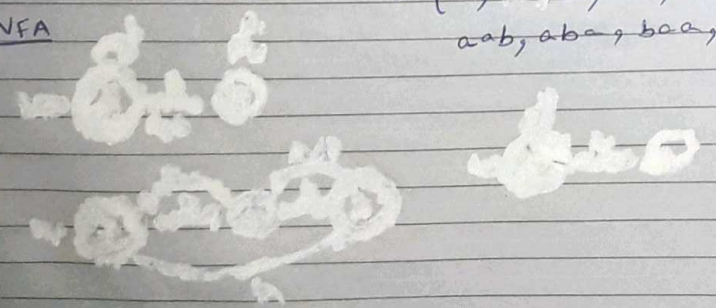
Maximum combinations $= 2^7$

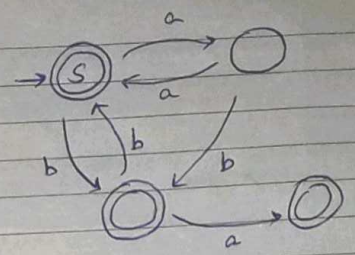
$\longleftrightarrow$

Q. $L = \{x \mid x \in \{a, b\}^*;\ x$ contains even # of a's or odd number of b's$\}$

$= \{\wedge, b, aa, ab, ba, bb, aab, aba, baa, bbb, \ldots\}$

NFA





In the end, you'd end up with DFA. Hence this diagram is discontinued.

$Q \subseteq P(Q)$

every DFA is also a NFA

(DFA is a subset of NFA)

$\longleftrightarrow$

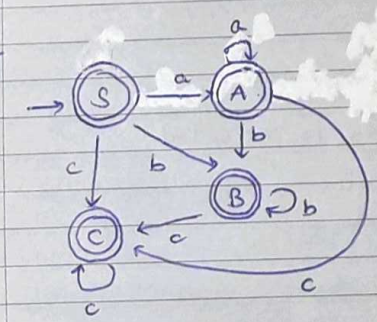
15/9

Construct NFA for the following language. $(a^* b^* c^*)$

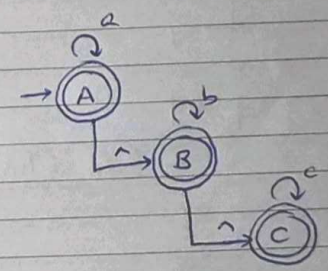
$L = \{a^n b^m c^p;\ n, m, p \geq 0\}$ $\quad \Sigma = \{a, b, c\}$

$L = \{\wedge,$

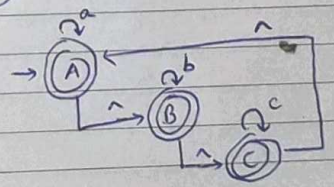
NFA



NFA $\wedge$



$(a^* b^* c^*)^* \longrightarrow \Sigma^*$



δ_{NFA} : $Q \times \Sigma \rightarrow P(Q)$

$\delta_{NFA\wedge}$: $Q \times (\Sigma \cup \wedge) \rightarrow P(Q)$

$\wedge \notin \Sigma$ Here it is NFA $\wedge$

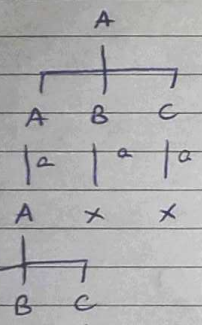
0 length

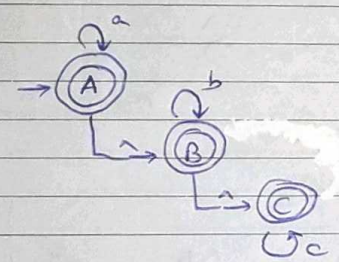
if $\wedge \in \Sigma$, then it will be NFA.

1 length

$\wedge$-closure

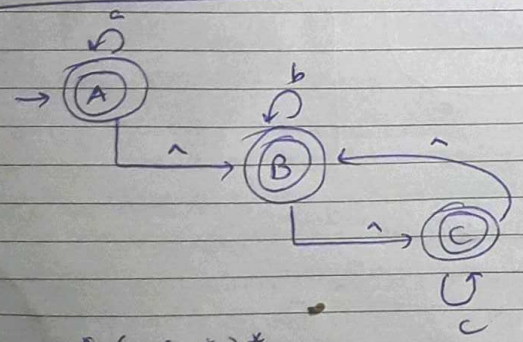
Trace Tree (ac)





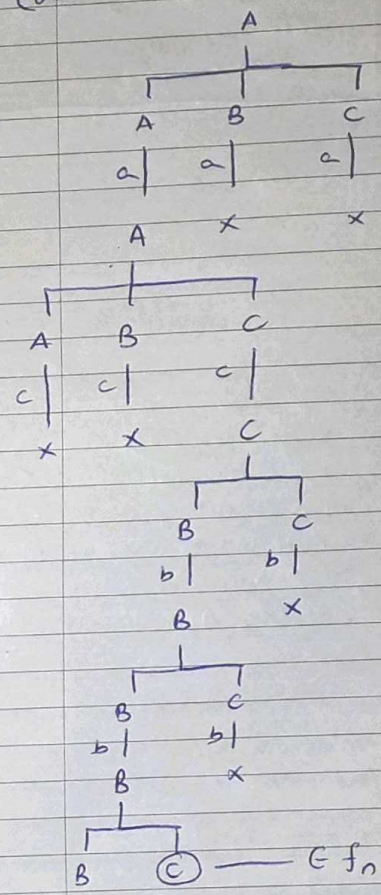
$(a^*b^*c^*)$

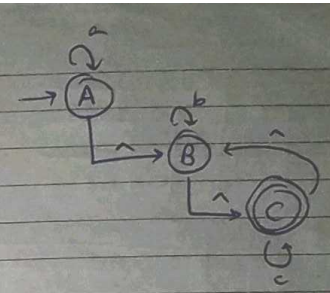
$C \in f_n$; hence ac is accepted.



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

Trace Tree (aabb)





Null Elimination Method

↳ NFA $\wedge$ to NFA

States will remain same.

(for both NFA $\wedge$ & NFA).

Transition table for NFA^

	a	b	c	^
A	A	x	x	B
B	x	B	x	C
C	x	x	C	B

Note

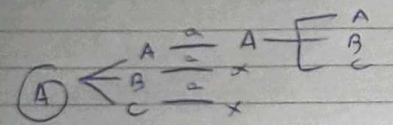
if $\Sigma = \{a, b, c, \wedge\}$ then given diagram is NFA.

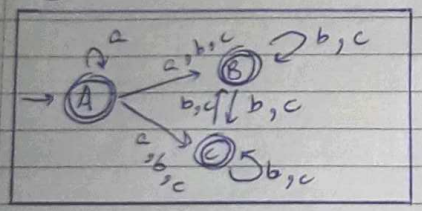
becz $|\wedge| = 1$

Transition table for NFA

	a	b	c
A	A, B, C	B, C	B, C
B	x	B, C	B, C
C	x	B, C	B, C

^ closure of A for 'a'.





State Diagram of NFA

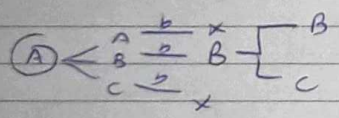
Final state for NFA:

① final state for NFA ^ (C).

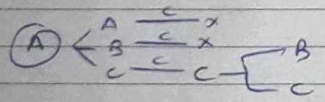
② all ^ accepting states (previous) which led to final state for NFA ^.

(B, then A in this case).

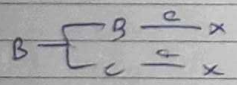
^ closure of A for 'b'.



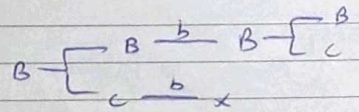
^ closure of A for 'c'.



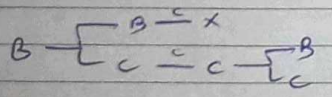
^ closure of B for 'a'.



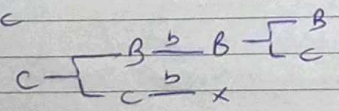
^ closure of B for 'b'.



^ closure of B for 'c'.

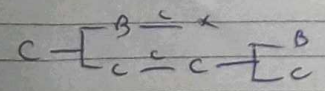


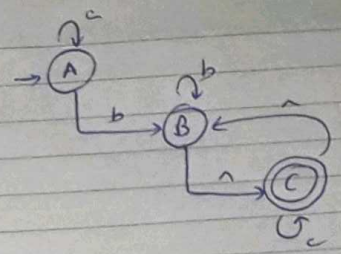
^ closure of C for 'a'.



^ closure of C for 'b'.

^ closure of C for 'c'.





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

bb^*

b^+ means that b can be accepted atleast once.

Extra (self study)

Regular Expression:

Machine (FA)
- DFA
- NFA

} purpose was validity of words/strings which exist inside language

The language which is accepted in finite Automata is "Regular Language".
Regular Expression is used to define or express a Regular Language.

Arithmetic Expression: operator, operand

Regular Expression: operator, alphabet

Operator:
- Union (+)
- Concat (.)
- Star (Kleene) (*)
- Positive Star (+)
- Parentheses ( )

& there's precedence too

Language Construction → Expression Construction

Examples :-

language

1) $L1 = \{\wedge\} \Rightarrow R.E = \wedge$

2) $L2 = \{a\} \Rightarrow R.E = a$

3) $L3 = \{b\} \Rightarrow R.E = b$

4) $L = \{\wedge, a, b\} \Rightarrow R.E = (\wedge + a + b)$

↓

Either $\wedge$ or a or b

5) $L = \{aa, ab, ba, bb\} \Rightarrow R.E = (aa + ab + ba + bb)$

length = 2

find pattern!

$R.E = (a+b) \cdot (a+b)$

$= aa \quad ba$

6) $L = \{a, b, aa, ab, ba, bb\}$ ⟷

$\Sigma^1 - \Sigma^2$

$R.E = (\wedge + a + b) \cdot (a + b)$

$\wedge a \quad = a$

$\wedge b \quad = b$

7) $L = \{\wedge, a, b, aa, ab, ba, bb\}$

$R.E = (\wedge + a + b) \cdot (\wedge + a + b)$

8) $L = \{\wedge, a, b, aa, ab, ba, bb \ldots\}$

$\Sigma^0, \Sigma^1, \Sigma^2, \ldots$

$R.E = (a+b)^* \longrightarrow$ Kleene star

$(a+b)^0 \mid (a+b)^1 \mid (a+b)^2 \mid \ldots$

↓ ↓ ↓

$\wedge$ — — —

9) $L = \{a, b, aa, ab, ba, bb, \ldots\}$

$\Sigma^1, \Sigma^2, \ldots$

$R.E = (a+b)^+ \longrightarrow$ positive star

$(a+b) \mid (a+b)^2 \mid \ldots$

↓ ↓

— — —

⟷

10) $L = \{x \mid x \in \{0, 1\}^*;\ x$ as a number is divisible by $2\}$

words

$L = \{0, 00, 10, 000, 010, 100, 110, \ldots$

$\Sigma^0 \quad \Sigma^1 \quad \Sigma^2$

$\Sigma^0 = \wedge \times$

$\Sigma^1 = 0, 1 \qquad x \% 2 = 0$

$\Sigma^2 = 00, 01, 10, 11$

$\Sigma^3 = 000, 001, 010, 011, 100, 101, 110, 111$

↓ ↓ ↓ ↓

0 10 100 110

$R.E = (0+1)^* \cdot 0$

11) $L = \{x \mid x \in \Sigma^*;\ x$ ⟷ as sum of digits divisible by $2\}$

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

$L = \{0, 00, 11, 000, 011, 101, 110$

$\Sigma^0 = \wedge \times$ (for now, not included)

$\Sigma^1 = 0, 1$

$\Sigma^2 = 00, 01, 10, 11 \qquad (0+0)\%2=0 \quad (1+1)\%2=0$

$\Sigma^3 = 000, 001, 010, 011, 100, 101, 110, 111$

$R.E = (0^* 1 0^* 1 0^*)^* + 0^*$

↓ ↓

or 1s (with zeroes) which are in multiple of 2. | Either only 0's

RE to NFA ∧ using Kleene's Algorithm:-

Q. R.E = $(a+b)^*a$

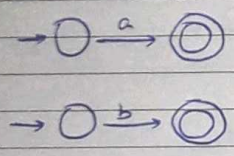
M_1 M_2

M_3

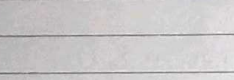
M_4

M_5

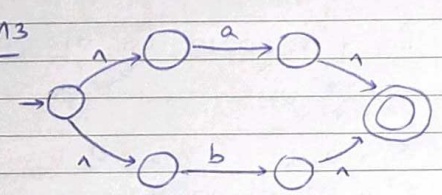
M1



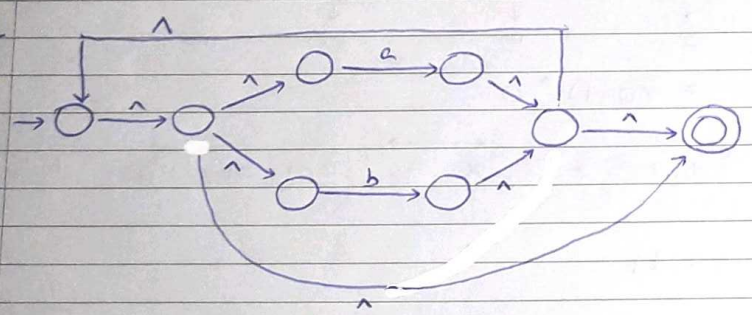
M2



M3



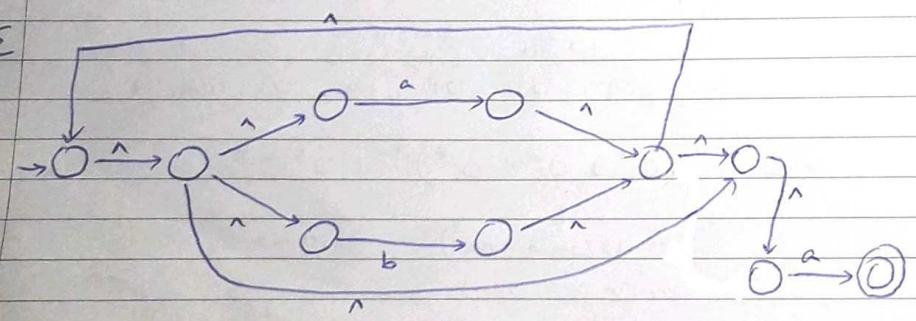
M4



aa

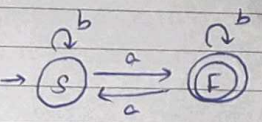
∧∧a∧∧∧a∧∧ = aa

M5



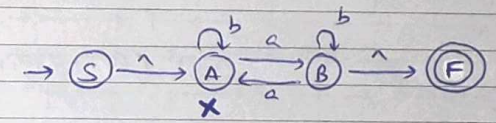
D : direct
I : indirect

Deterministic Finite Automata to Regular Expression
(DFA) (RE)



$q_0 = S$
$\Sigma = \{a, b\}$
$Q = \{S, F\}$
$f_n = \{F\}$

Step 1 Make Generalized Nondeterministic Finite Automata. (GNFA)

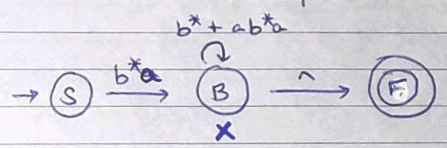


Concept: 1 by 1, remove states & RE will appear on transition arrows.

Pathway if we remove 'A'

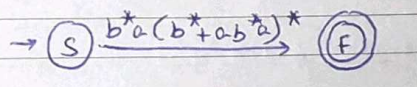
	D	I		D	I
$S \to B$?	x	b^*a	$B \to B$?	b^*	$b^*+(ab^*a)$
$S \to F$?	x	x	$B \to F$?	∧	x

no incoming arrow in start
x S-S
no outgoing arrow in end
x F-F



if we remove 'B' now,

	D	I
$S \to F$	x	$b^*a(b^*+ab^*a)^*\wedge$

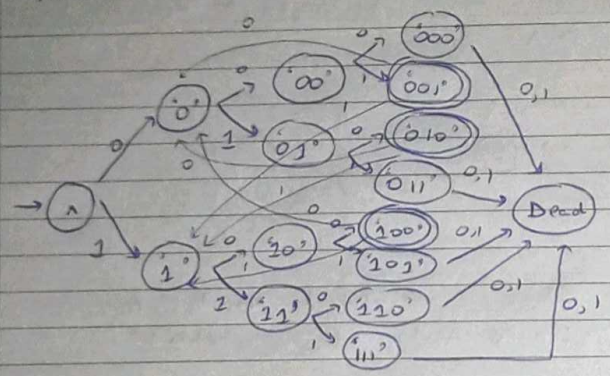


R.E
$= b^*a(b^*+ab^*a)^*$

20/9

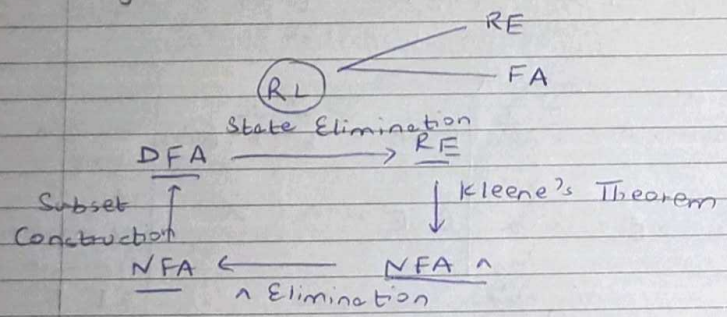
Given x is divisible of 3 & 2 0's.

001	010	100

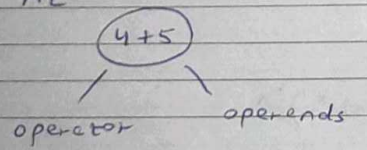


$\longleftrightarrow$ R.E $= (001+010+100)^*$

Regular Expression:-



AE



Q. $L = \{x \mid x \in \Sigma^*;\ x$ ends with 'a'$\}$

R.E $= \Sigma^* a$

Q. $L = \{x \mid x \in \Sigma^*;\ x$ is divisible by $5\}$

R.E $= (0+1+2+3+4+5+6+7+8+9)^* \cdot (0+5)^*$

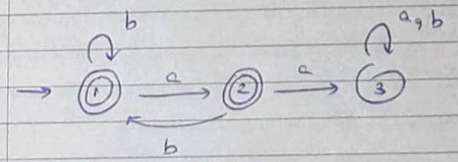
Q. $L = \{x \mid x \in \Sigma^*;\ x$ doesn't contain aa$\}$

R.E $= b^*(ab^+)^*$ ✗

$= \{\wedge, a, b, ab, ba, bb,$

$(\wedge b^* \wedge)(b^* + \wedge)$

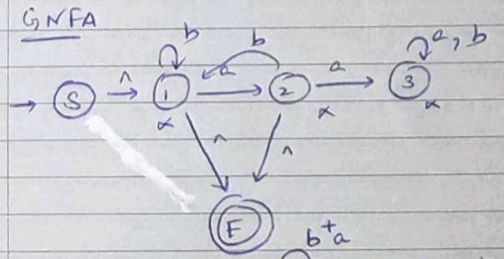
$\downarrow$ b^* $\quad$ $\downarrow$ b^* or $\wedge$

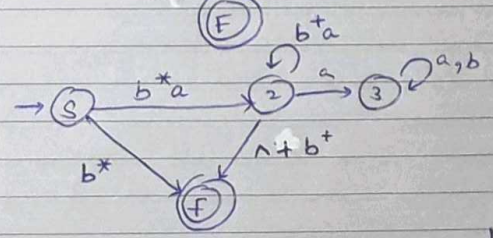


Ans: $b^*(ab^+)^*$

State Elimination Method:

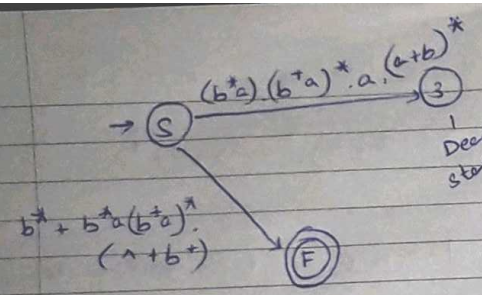
GNFA





Path	Direct	Indirect
$S \to 2$	✗	b^*a
$S \to F$	✗	b^*
$2 \to 2$	✗	b^+a
$2 \to F$	^	b^+

Source — Delete Node — Destination



	D	I
S-F	b^*	$b^* \ldots$
S-3	×	$(b^*a).(b^+a)^*.a.(a+b)^*$

$R.E. = b^* + b^*a.(b^+a)^*.(\wedge + b^+)$

for doesnt contain aa

$(aa+b)^*$

↓

NFA ^

Pg 102

John C Martin

22/9

Q1 :- $a(c+b)^*b$

Q5) :- $\delta^*(4, ab)$

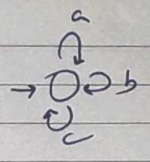
$\delta(\delta^*(4,a), b)$

$\delta(\delta(\underbrace{\delta^*(4,\wedge)}_{\downarrow}, a), b)$

$\delta^*(4,\wedge) = \wedge(\{4\})$
$= \{4, 1, 2, 5\}$

$\delta^*(4,a) = \wedge\{\delta(4,a) \cup \delta(1,a) \cup \delta(2,a) \cup \delta(5,a)\}$
$= \wedge\{4, 3\}$
$= \{4, 1, 2, 5, 3\}$

$\delta^*(4,ab) = \wedge\{\delta(4,b) \cup \delta(1,b) \cup \delta(2,b) \cup \delta(5,b) \cup \delta(3,b)\}$
$= \wedge\{3, 6, 7, 4\}$
$= \{3, 1, 2, 5, 6, 7, 4\}$
$\in f_n$ ✓



$\boxed{(a+b+c)^*} \neq a^* + b^* + c^*$

$\longleftrightarrow$