Chapter 01

Automata : The methods and the madness

✓ Brain function (situational knowledge adaption)

Intractable problem/ NP-hard problem

NP hard Problem: Limited time এর মধ্যে solve করা

যায় না। (traveling salesman problem)

1.1 : Why study automata theory

1.1.1 : Introduction to finite automata

1) software for checking /design for digital circuit.

(Ex digital clock)

2) Lexical analysiszer (separate identifier,

keywords (compiler এর একটা part). & ~~syntetic~~

punctuation

3) Scanning large body of text (frequency of a

character)

security break

সবচেয়ে বেশি ব্যবহৃত letter e.

1.1.2 structural Representation

1. Grammer

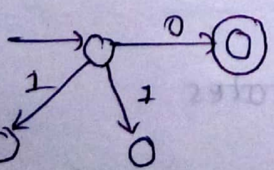
2. Regular Language/expression

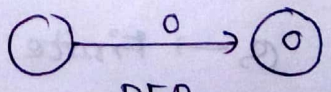
2-C

DFA: Deterministic Finite Automata

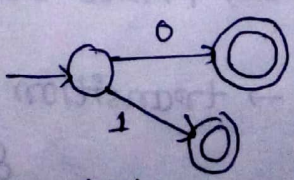
NFA: Nondeterministic Finite Automata

DFA: Refines to the fact that on each input there is only and one state to which the automation can transation from its current state.

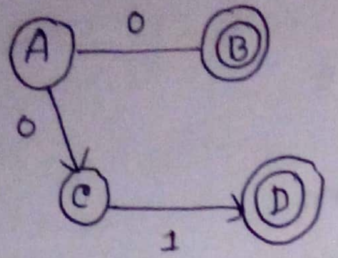




DFA



DFA



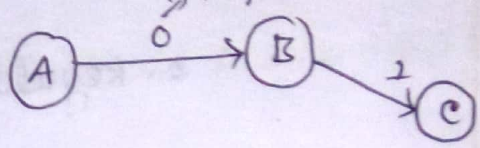
node $\rightarrow q$

q_0, q_1, q_2

input $\rightarrow a$

NFA

input

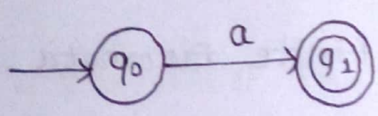


transition diagram

transition function:

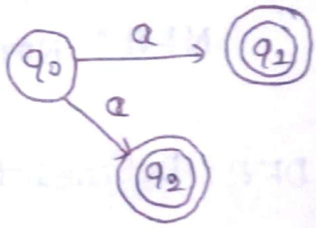
$\delta(q, a) =$

δ (current state, input) = output node



DFA

$\delta(q_0, a) = q_1$



$\delta(q_0, a) = q_1 q_2$

Definition of Deterministic Finite Automata.

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

$Q \rightarrow$ Finite sets of states

$\Sigma \rightarrow$ Finite set of symbols (input/edge)

$\delta \rightarrow$ transition fn

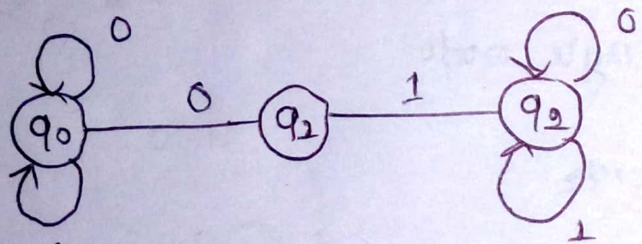
$$\delta(q_0, 1) = q_1$$

4. $q_0 \rightarrow$ starting state

5. Final state — F

2.2.2 How a DFA process strings.

L = { w | w is of the ~~form~~ x01y ~~for~~ some string x and y consisting of 0's and 1's }



Lab : ~~finite~~ automata DFA implement.

transition (input, current node

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

একের অধিক হতে পারে,

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

ending node

q_0 q_2 q_1

δ (current node, input) = next node

$\delta(q_0, 1)$ = single node

$= q_1, q_2$

2.2.3 Simpler notation for DFA's :

1. A transition diagram

2. A transition table.

64 page: $x \; 0 \; 1 \; y$ string

0,1 এর combination

~~যেকোনো~~ সংখ্যক

0,1 এর combination

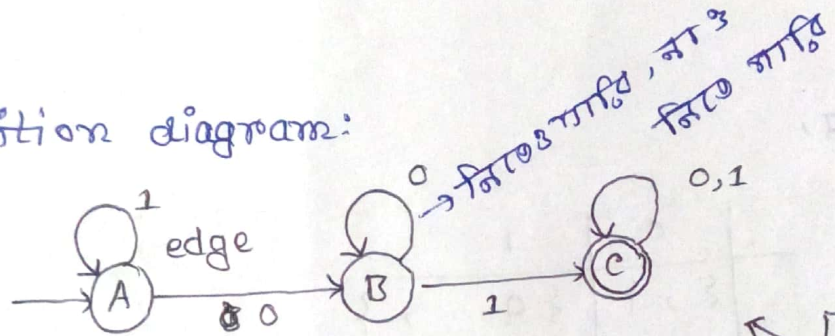
01

001 01 000

001 01 00100

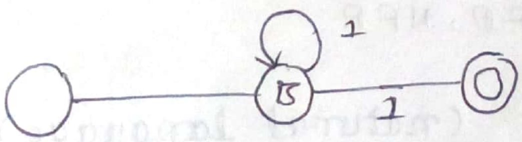
but 100 00 ~~000~~

transition diagram:



$\boxed{x\ 01\ y}$

DFA



1টা input নিয়ে একের অধিক node এ যেতে পারলে NFA

66 page:

2. A transition Table:

→ table বানাবো

input

Row (state/node) Q	0	1
→ q_0	q_2	q_0
q_2	q_2	q_1
* q_1	q_1	q_1

→ single

NFA হলে,

	0	1
	{ , }	{ q, , }

Lab: table বানাবো DFA, NFA

x 01 y (natural language)

Example: 2.4:

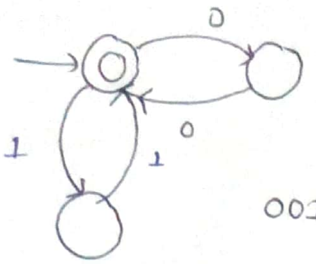
$L = \{ w / w$ even number of 0's and even number of 1's $\}$

✓ 0011

✓ 0101

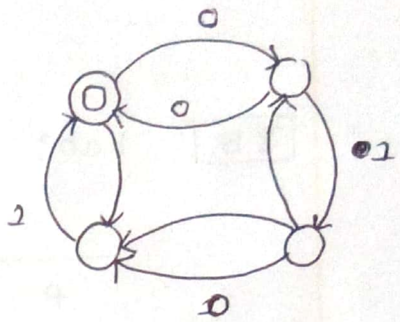
✓ 10111000

◎ starting, ending হতে পারে।

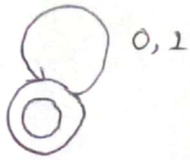


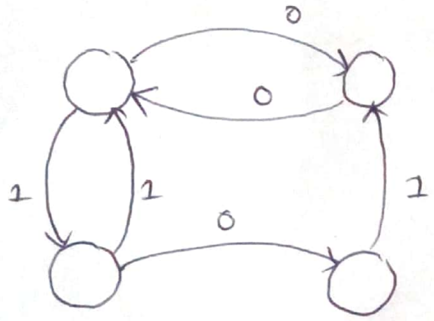
0011

DFA

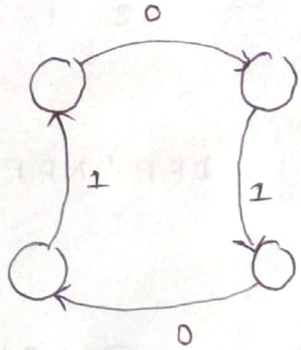


NFA দিয়ে



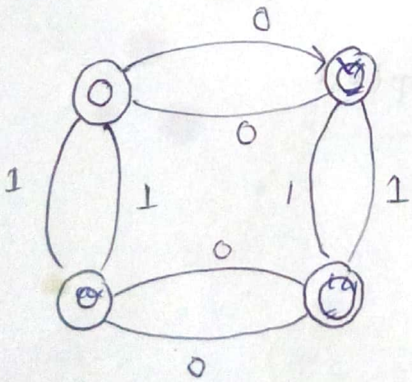


1010

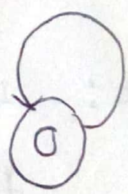


01 01

68 pg



DFA তে acceptable



NFA তে acceptable

71 page (exercise) 2.2.6

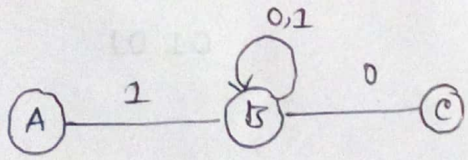
2.2.1 − 2.2.11

$\boxed{3B}$ Lab:

	0	1
A	B	A
B	C, A	A
C	C	C

DFA / NFA ?

2. w = 1 . 01 0



input : 1001 ⟶ not accepted

1000 ⟶ accepted.

p[1] [0] [0] [1] [1]

cout

~~10~~ 3B

if (0,0,...,)

if (0,0,0,0)

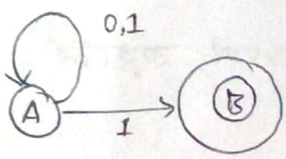
1011

Enter

for (i=0; i<4; i++)

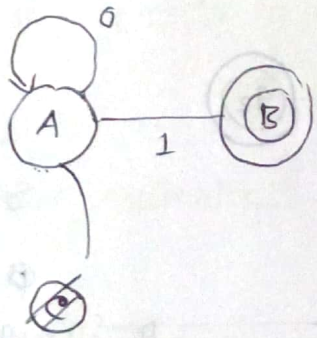
if (

NFA :

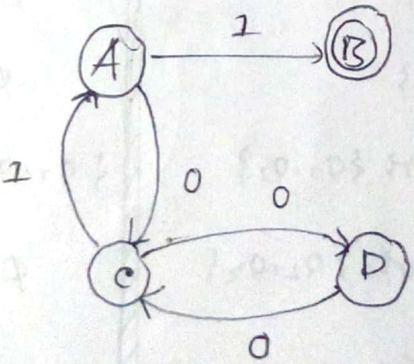


probability create হয়,
computation টাইম বেড়ে যা

DFA :

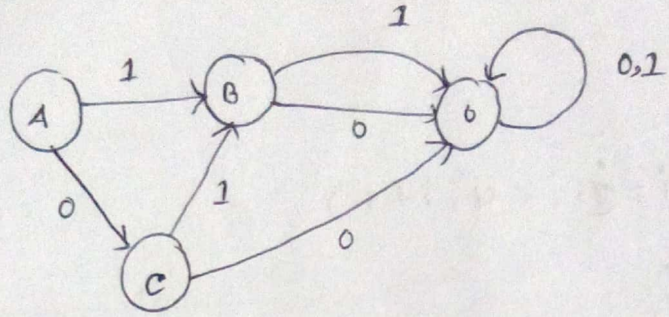


A



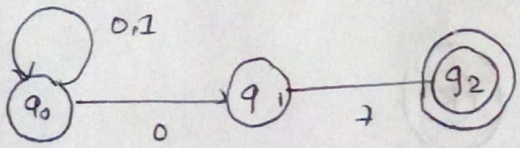
string

DFA:



NFA থেকে DFA তে convert করলে maximum 2^n হবে।

$2^2 = 2^2 = 4$

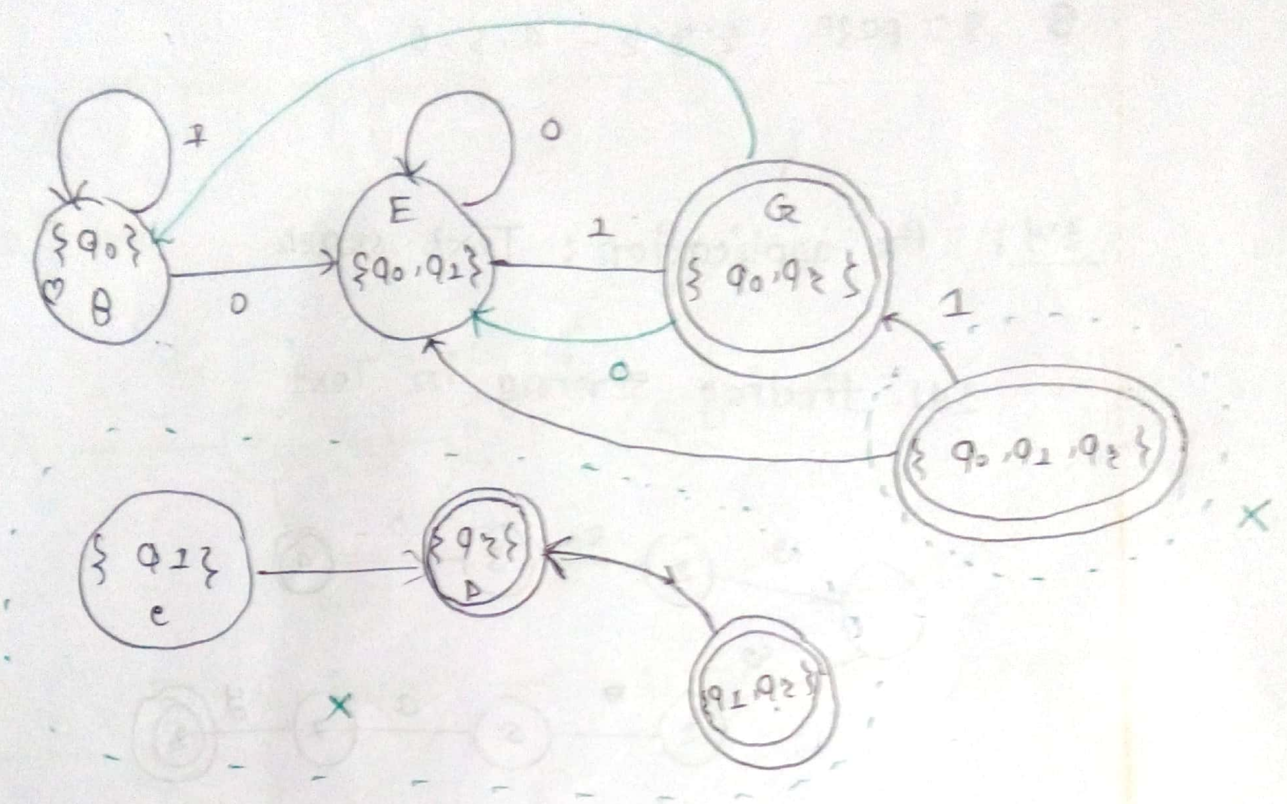


max $2^3 = 8$

min (5 - 8).

		0	1
A	ϕ	ϕ A	ϕ A
B	$\{q_0\}$	$\{q_0, q_1\}$ E	$\{q_0\}$ B
C	$\{q_1\}$	ϕ A	$\{q_2\}$ D
* D	$\{q_2\}$	ϕ A	ϕ A
E	$\{q_0, q_1\}$	$\{q_0, q_1\}$ E	q_0, q_2 G
* F	$\{q_1, q_2\}$	ϕ A	q_2 D
* G	$\{q_0, q_2\}$	$\{q_1, q_0\}$ E	$\{q_0\}$ B
* H	$\{q_0, q_1, q_2\}$	q_0, q_1, q_0 E	q_0, q_2 G

⟨Linklist / Graph⟩.



Lab: NFA থেকে DFA convert

Page no: 79, equivalent input 2

$\{q_0\} \rightarrow \{q_0\} \rightarrow 1$

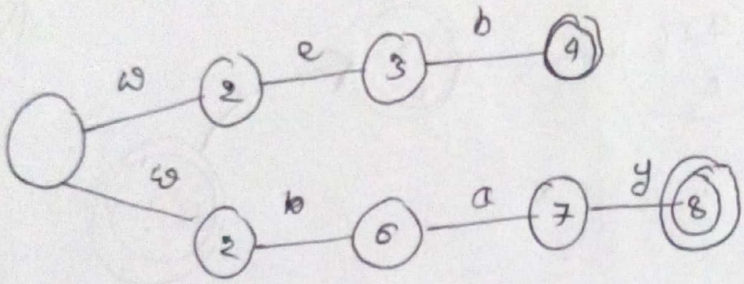
$\{q_0\} \rightarrow \{q_0, q_1\} \rightarrow 0$

Scanned by CamScanner

৬ ৪৩ page ২·৩·২ - 4, 5, 6

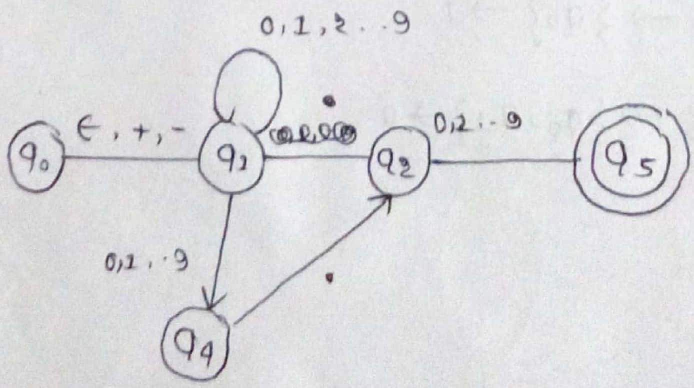
২·4: An application: Text seach..

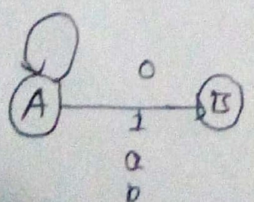
২৪১ finding string in Text



৩৫

২·5. Finite Automata with Epsilon Transitions (ε)





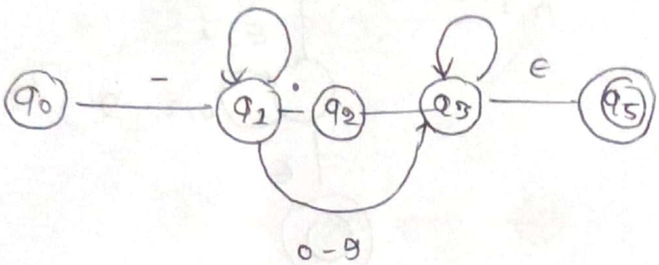
input ছাড়াই এক node থেকে অন্য node এ ~~[illegible]~~

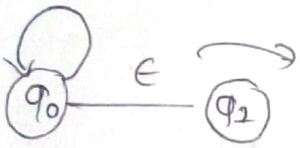
~~[illegible]~~ যাওয়া যায়।

* যেনো যেকোনো number generate করা যায়।

$\epsilon \rightarrow$ no input

-85





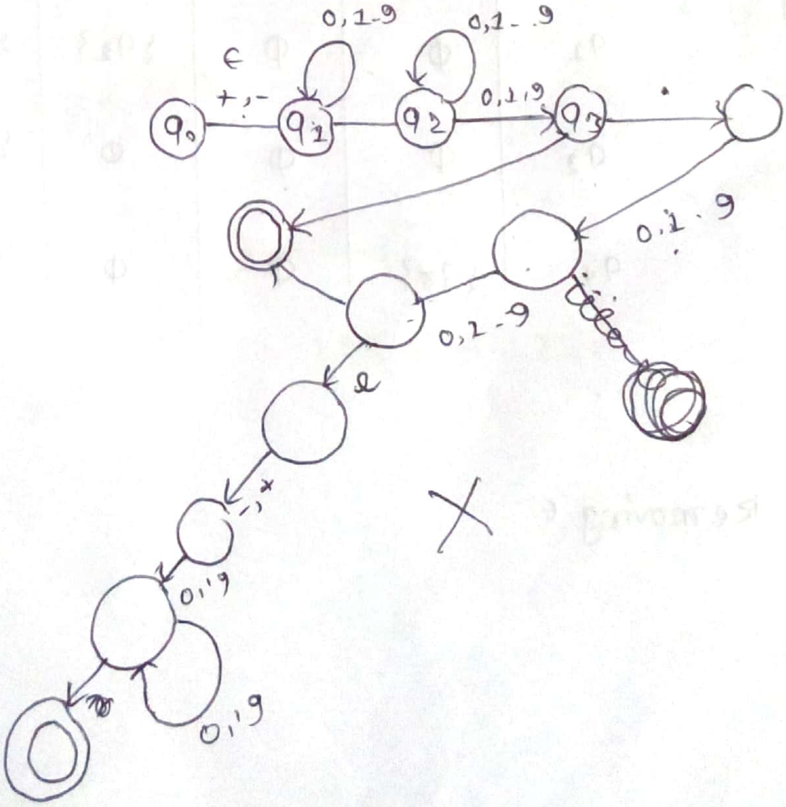
→ no input তাই q1 কে q0 দ্বারা replace করা যায়।

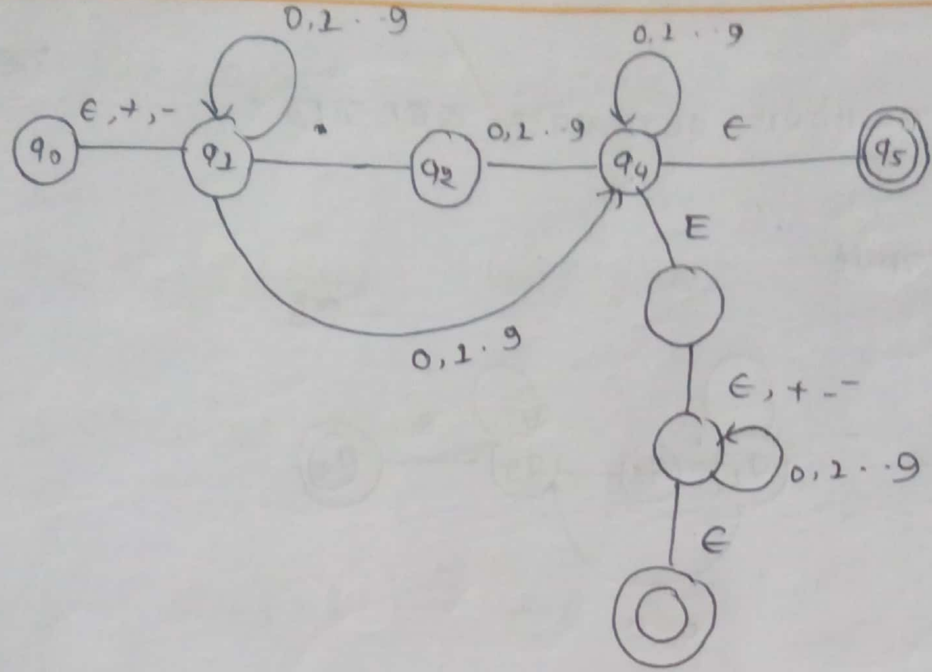
$+123$

$+123.10$

$+123.10\times10^{-23}$

$+123.10\times10^{-24}$





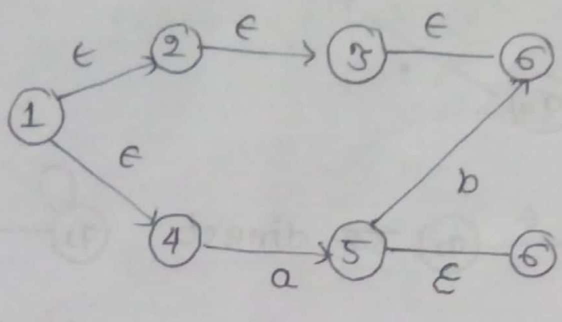
	ε	+, -	.	0,1 . . 9
q_0	$\{q_1\}$	$\{q_1\}$	ϕ	ϕ
q_1	ϕ	ϕ	$\{q_2\}$	$\{q_1\}$
q_2	ϕ	ϕ	ϕ	$\{q_4\}$
q_4	$\{q_5\}$	ϕ	ϕ	$\{q_4\}$

Removing ε

4 A

২০৫'৩

Epsilon closure :

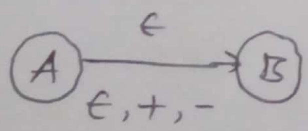


1,2,3,6,4 একই node বলা যায়।
1 থেকে কোনো কিছু না নিয়ে 2,3,6,4 এ easily যাওয়া যায়।
কিন্তু 1 থেকে 5 এ যাওয়া যায় না।

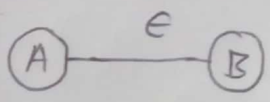
edge point 3টা
ϵ, a, b

Eclose(1) = {1, 2, 3, 4, 6}

5 is not Eclose (1)

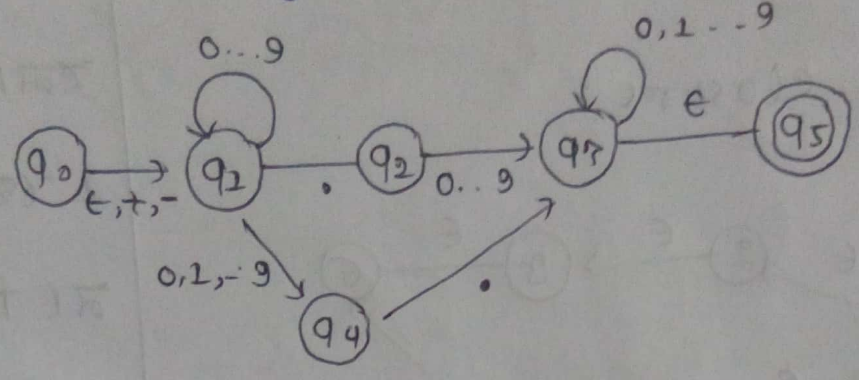


+128
-128
or 128



128 ≠ +128

2.5.5 Eliminating ϵ-transition:



only ϵ হলে $q_0 \xrightarrow{\epsilon} q_1$ কে direct $q_1 \longrightarrow q_2$ দ্বারা denote করা যায়

$\epsilon, +, -$ হলে করা যায় না।

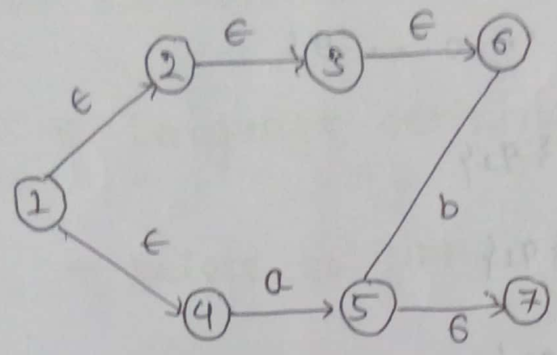
q_0 এর enclosure = $\{q_0, q_1\}$

starting node এর enclosure বের করতে হবে।

4A

2.5.4

Epsilon closure:

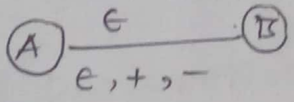


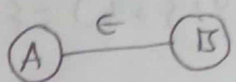
1,2,3,6,7 একই node বলা যায়। 1 থেকে 2,3,6,4 এ easily যাওয়া যায়। কিন্তু 1 থেকে 5 এ যাওয়া যায় না।

edge input ৩টা ε, a, b

Eclose(1) = {1,2,3,4,6}

5 is not Eclose(1)



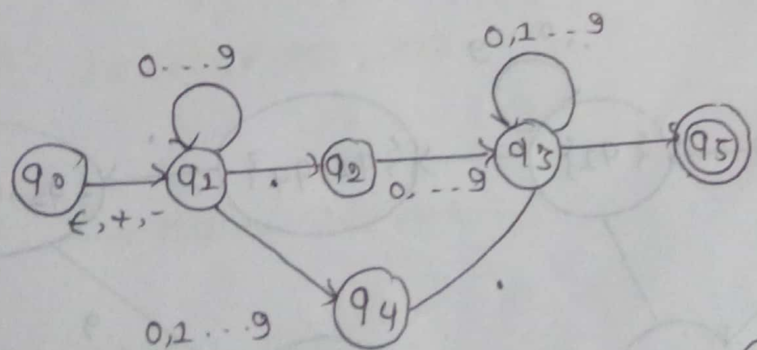


128 7 +128

+128

-128

2.5.5 Eliminating ε-transition:



only 'ε' হলে, $q_0 \xrightarrow{\epsilon} q_1$ কে direct $q_1 \rightarrow q_2$ দ্বারা denote করা যায়। ε, +, - হলে করা যায় না।

q_0 এর enclosure $= \{q_0, q_1\}$

1. starting node এর enclosure বের করতে হবে।

$\{q_0, q_1\} = \{+, -, \cdot, 0 \ldots 9\}$

$\delta(\{q_0, q_1\}, +) = \{q_1\}$

$\delta(\{q_0, q_1\}, -) = \{q_1\}$

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

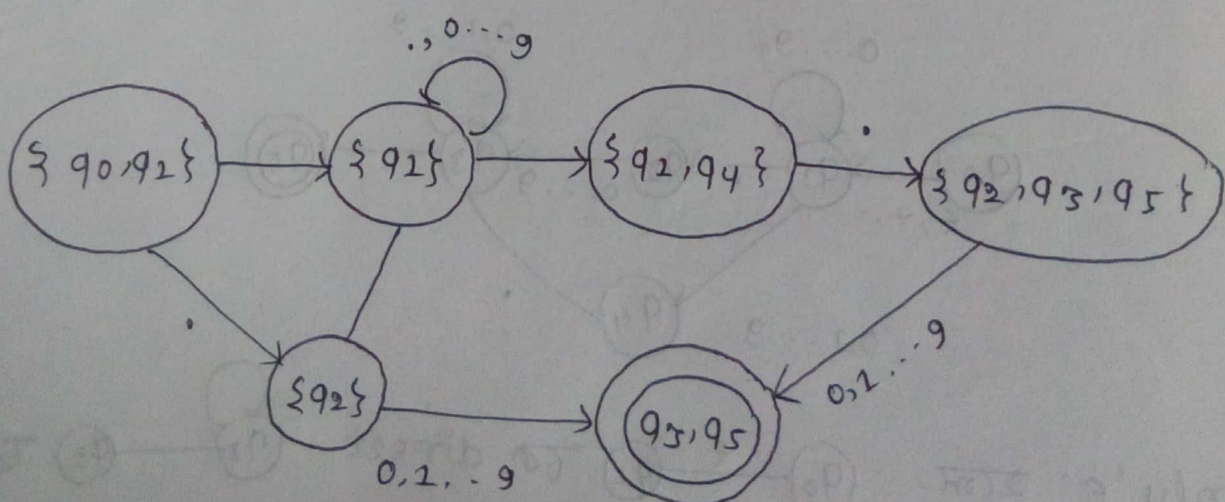
$\delta(\{q_0, q_1\}, 0) = \{q_1, q_4\}$

$\delta(\{q_0, q_1\}, 1) = \{q_1, q_4\}$

$\delta(\{q_0, q_1\}, +) = \{q_1\} = \{\cdot, 0, 1 \ldots 9\}$

$\delta(\{q_0, q_1\}, \cdot) = \{q_2\} = \{0, 1 \ldots 9\}$

$\delta(\{q_0, q_1\}, 0) = \{q_1, q_4\} = \{\cdot, 0, 1 \ldots 9\}$



Chapter 03.

Regular Expression and Languages

* Language defining notation

* Relate to NFA

* User friendly (NFA)

* Impose Algebric law

* Declarative way.

যেকোনো সংখ্যক 1 accepted.

$\epsilon, 1, 11, 111 \ldots$

1* দ্বারা denote করা যায়।

$L^* \longrightarrow$ star/ closure

$\epsilon, L, LL, LLL, LLLL \ldots$

Applications:

1. Search command
2. Lexical Analyser (source code থেকে taken (keyword) separate করে যাতে machine language এ easily convert করা যায়।

① Union:

$L = \{001, 10, 111\}$

$M = \{\epsilon, 001\}$

$L \cup M = L + M = \{\epsilon, 001, 10, 111\}$.

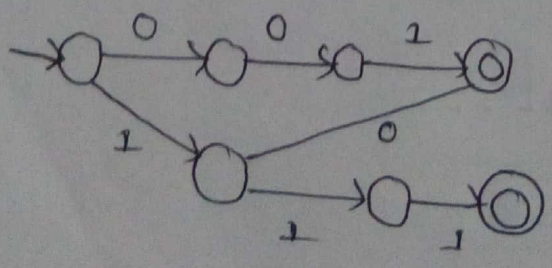
② concatanation; (dot or no operator)

$LM = \{001, 10, 111, 001001, 10001, 111001\}$

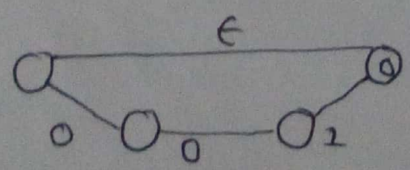
③ closure (star/knee/closure)

$L^* = \{\epsilon, L, LL, LLL \ldots\}$

L =



M =



$(10)^* = \epsilon, 10, 1010, 101010 \ldots$

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

$= \epsilon, 1+0, 11, 10, 01, 00 \ldots$

$10^* = 1(\epsilon, 0, 00, 000 \ldots$

$= 1, 10, 100, 1000$

$\boxed{((1+0)^* + 0)^*}$

3.1.4 → 109

express করতে হবে।

closure Properties :

$L^* = \epsilon, L, LL, LLL, \ldots$

$L = (10)$

$L^* = (10)^*$

$= \epsilon, 10, 1010, \ldots$

$U_{i \geq 0} L^i$

$L^0 = \{\epsilon\}$

$L^1 = L$

$L^2 = L \cdot L = LL$

$\vdots$

$L^i = L^i$

* $L = (1+0)$

$L^* = (1+0)^*$

$= \epsilon, 1+0, (1+0).(1+0), \ldots$

$= \epsilon, 1+0, \{11, 10, 01, 00\}$

$L = \{0, 1 \ldots\}$

$L \cdot L = \{0,1\} \cdot \{0,1\} = \cancel{\{0,0},$

$L2 = \{00, 01, 10, 11\} = 4$

ϵ এবং ϕ এর ~~সমার্থক~~ নয়।

input, but কোনো input এর value নেই।

$\{\epsilon\} \longrightarrow$ একটা element হিসেবে

ϵ^*

$= \{\epsilon, \epsilon\epsilon, \epsilon\epsilon\epsilon, \ldots\}$

$= \epsilon$

$\phi^* = \{\epsilon, \phi, \phi\phi \ldots\}$

$= \{\epsilon\}$

something $\times \epsilon$ = something

$1 \% 2 * 3$

$= \frac{1}{2} \times 3$

$= \frac{3}{2}$

$1 * 2 \div 3$

$= 2 \div 3$

$= \frac{2}{3}$

1. ( ) { } [ ]

2. * /

3. + , -

Operation of Regular Expression:

Union $\cup (+)$

concatenation $c (*)$

closure $* \longrightarrow 1$

$(1+0)$

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

$= (1+0) (\epsilon, 1, 11, 111, \ldots) + 0$

$= \{(1+0), (1+0)1, (1+0)111, \ldots\}$

$= \{1, 0\}, \{11, 01\}, \{111, 011\} + 0\}$

+0

starting 1, পরবর্তীতে শুরু 0 দিয়ে এবং যেকোনো সংখ্যক 1 থাকতে পারে।

$\{\{11, 10, 1\} + 0\}$

$\{\{11, 10, 1\} 0\}$

3.1.4

3.1.2 এর example (b)

b) The set of strings of 0's and 1's whose number of 0's is divisible by five.

$$1\,\underline{(1^*0^*1^*0^*)}^*\,0$$

সংখ্যক

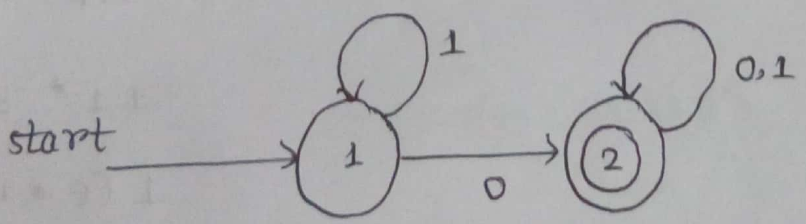
5, 10, 15, 20, → 0 থাকতে হবে।

——— 0 ———

3.2.1 5B

From DFA's to Regular Expression (RE);

$$R^{(K)}_{ij} = R^{(K-1)}_{ij} + R^{K-1}_{iK}\left(R^{K-1}_{KK}\right)^* R^{(K-1)}_{Kj}$$



For $k=0$;

$R_{11}^{(0)}$	$\epsilon+1$
$R_{12}^{(0)}$	0
$R_{21}^{(0)}$	ϕ
$R_{22}^{(0)}$	$\epsilon+0+1$

$K=1;$

$$R_{ij} = R_{ij}^{0} + R_{i1}^{(0)} (R_{11}^{(0)}) R_{1j}^{(0)}$$

$R_{11}^{(1)}$	$\epsilon+1 + (\epsilon+1)(\epsilon+1)^*(\epsilon+1)$	1^*
$R_{12}^{(1)}$	$0+(\epsilon+1)(\epsilon+1)^* 0$	$1^* 0$
$R_{21}^{(1)}$	$\phi + \phi(\epsilon+1)^*(\epsilon+1)$	ϕ
$R_{22}^{(1)}$	$\epsilon+0+1+\phi(\epsilon+1)^* 0$	$\epsilon+0+1$

$\epsilon+1+(\epsilon+1)(\epsilon+1)^*(\epsilon+1)$

$= \overset{\epsilon+1}{\cancel{(\epsilon+1)}} + (\epsilon+1)\{\epsilon, \epsilon+1, (\epsilon+1)(\epsilon+1) \ldots\}(\epsilon+1)$

$= \epsilon+1+\epsilon+\{(\epsilon+1)(\epsilon+1) \ldots\}$

$= \epsilon+1+\epsilon+\epsilon+1+ \cdot 1 + \ldots \cdot 1+1+ \ldots$

$1.\epsilon = 1$

$1.\phi = \phi$

$RR^* = R^*$

$11^* = 1^*$

$1(\epsilon, 1, 11, 111$

$\{\epsilon, 1, 11, 111\}$

$R^*R = R^*$

$R+R^* = R^*$

$0+(\epsilon+1)(\epsilon+1)^* 0$

$= 0 + (\epsilon+1)^* 0$

$= 0 + 1^* 0 \quad = \{0+(\epsilon, 1, 11, 111 \ldots) 0$

$\quad \{(0, 10, 110, 1110 \ldots)$

$= 0, 1^* 0$

$= 0, 0, 10, 110, 1110, \ldots$

$= 0, 10, 110, 110 \ldots$

For $k=2$,

R_{12} accepted only.

$R_{11}^{(2)}$	$1^* + 1^*0(\epsilon+0+1)^*\phi$	1^*
$R_{12}^{(2)}$	$1^*0 + 1^*0(\epsilon+0+1)^*(\epsilon+0+1)$	$1^*0(0+1)^*$
$R_{21}^{(2)}$	$\phi + (\epsilon+0+1)(\epsilon+0+1)^*\phi$	ϕ
$R_{22}^{(2)}$	$\epsilon+0+1+(\epsilon+0+1)(\epsilon+0+1)^*(\epsilon+0+1)$	$(0+1)^*$

*

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

$= (\epsilon, 1, 11, 111 \ldots)\, 0\, (0+1)^*$

$= (0, 10, 110, 1110 \ldots)(\epsilon, (0+1), (0+1)(0+1) \ldots)$

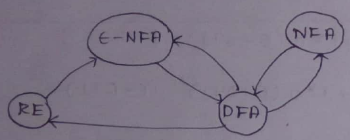
$= (0, 10, 110, 1110 \ldots)(\epsilon, 0, 1, 00, 01, 10, 11 \ldots)$

$= 0, 10, 110, 1110, \ldots, 00, 100, 11100 \ldots$

5C

3.2

Finite Automata and Regular Expression



Different conversion..

$\Phi R = R\Phi = \Phi$

$\Phi + R = R + \Phi = R$

$\boxed{(R^*)^* = R^*}$ prove

$(\epsilon + R)^* = R^*$

$R + S = S + R$

$(\epsilon + 0 + 1)^*$

$= (\epsilon+0+1), (\epsilon+0+1)$

$= (0+1)^*$

$= \epsilon, 0+1, (0+1)(0+1).$

$= \epsilon, 0, 1, 00, 01, 10, 11$

R (Regular Expression) =

$(R^*)^* = R^*$

Let $R = 1$,

$R^* = 1^* = \epsilon, 1, 11, 111 \dots$

$(R^*)^* = (1^*)^* = (\epsilon, 1, 11, 111)^*$

$= (\epsilon + 1 + 11)^*$

$= \epsilon, \epsilon+1+11, (\epsilon+1+11)(\epsilon+1+11) \dots$

$= \epsilon, 1, 11, 111, \dots$

$\epsilon + 0 + 1 + (\epsilon+0+1)(\epsilon+0+1)^*(\epsilon+0+1) = (0+1)^*$

Let, $R = \epsilon + 0 + 1$

So, $R + R(R)^* R$

$= R + (R)^* R$

$\boxed{\begin{array}{l} = R + R^* \\ = R^* \end{array}}$

$= (\epsilon+0+1)^*$

$= (0+1)^*$

$a + R^* R^*$

$= R^*$

$R = 1$

~~$(1+1)^*$~~

$1 + 1^*$

$= 1 + \epsilon, 1, 11 \dots$

$= \epsilon, 1, 11, 11 \dots$

$= 1^*$

3.2.3

Converting RE ~~of~~ to Automata : (ϵ-NFA)

Regular expression

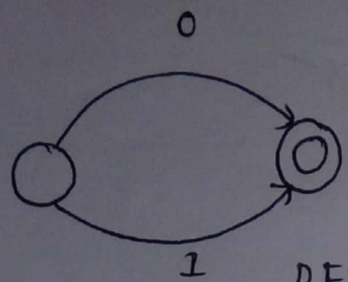
1^*

$(1+0)^*$

$\boxed{0+1}$

$\{0,1\}$

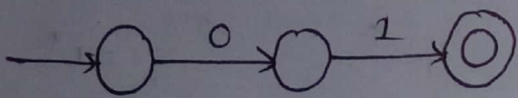
0 अथवा 1



DFA

$\boxed{01}$

$(10)^*$

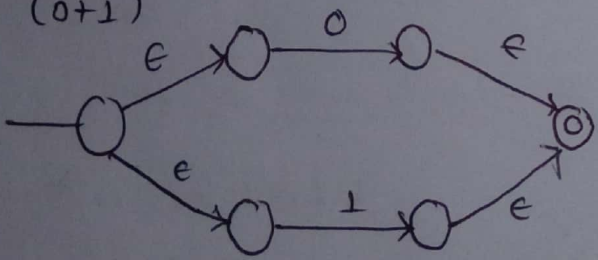


$1^* = \epsilon, 1, 11, 111 \ldots$ | $(0+1)^* = \epsilon, (0+1), (0+1)(0+1) \ldots$

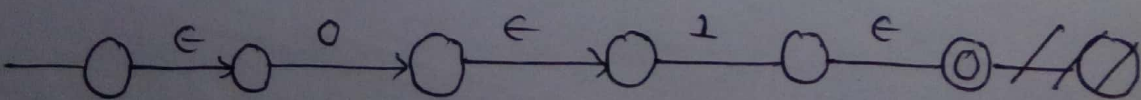
$= \epsilon, 0, 1, 00, 01, 10, 11, \ldots$

$000, 001, 010, \ldots$

NFA (0+1)

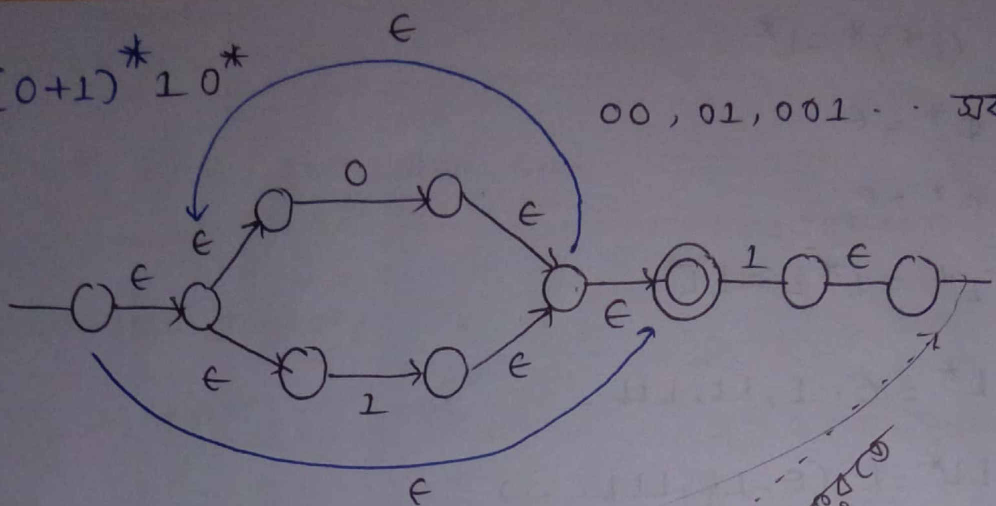


NFA (01)



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

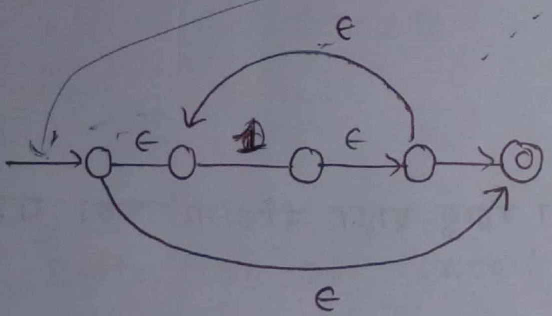
00, 01, 001 ... সবই accepted...



ε NFA

add করতে হবে,

1^*



3.3.4, 3.3.2, 3.3.5, 3.3.6

Identities: $0+x=x$, $1+x=x+1$

Annihilator: $0\times x = x\times 0 = 0$

$\phi + L = L + \phi = L$

$\epsilon L = L\epsilon = L$

$\phi L = L\phi = \phi$

Idempotent law

$L+L=L$

Distribute Law: $L(M+N) = LM + LN$

$(M+N)L = ML + NL$

Scanned by CamScanner

$(L^*)^* = L^*$

$\phi^* = \epsilon$

$\epsilon^* = \epsilon$

$L^+ = L^* L = L L^*$

$L^* = \epsilon, L, LL, LLL \ldots$

$LL^* = L\ (\epsilon, LL, LLL, \ldots)$

$= L\epsilon, LL, LLL$

$= L, LL, LL \ldots$

$= L^+$ (কোনো কিছু ছাড়া finish করা সম্ভব নয়)

$\boxed{৩.৪.৪}$

$(A \cap B)' = A' \cup B'$

$(A^* B^*)^* = A^* + B^*$

$$\text{proof}\begin{cases} (L^* m^*)^* = (L+m)^* \\ (L+m)^* = (L^* m^*)^* \end{cases}$$

chapter - 05

Context Free grammer and Language.

Palindrome number

		production rule
		$P \to \epsilon$
MADAM	101	$P \to 0$
	010	$P \to 1$
	00100	$P \to 0P0$
	01010	$P \to 1P1$

5.1.2 Definition of context free grammer (~~CF~~ CFG)

$G = (V, T, P, S)$

$V \to$ A set of ~~production~~ variable ~~rule~~

$T \to$ terminal

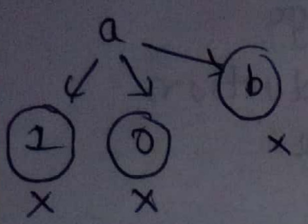
$P \to$ A set of finite ~~variable~~ produce

$S \to$ starting symbol

variable : i) terminal → small letter

ii) nonterminal

terminal কে আর ভাগ করা যাবে না।

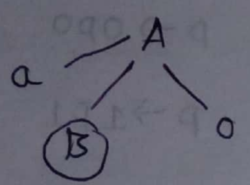


replace করা যায় না।

Nonterminal

nonterminal → production rule →

Terminal/nonter



non ⟶ term/non

nonterminal/terminal

দিয়ে replace

101

nonterminal - এই replace করা যাবে না

P ⟶ 1P1

↓ replace

⟶ 101

01 01 010

P ⟶ 0P0

⟶ 01P10

⟶ 010P010

⟶ 0101010

$(101)^*$

$(a+b)\ (a+b+0+1)^*$

$(a+b)\ \{\ \epsilon,\ (a+b+0+1),\ (a+b+0+1)\ (a+b+0+1)\}$

$(a+b),\ aa+ab+a0+a1+ba+bb+b0+b1$

$E \rightarrow \epsilon$

$E \rightarrow I$

$E \rightarrow E+E$

$E \rightarrow E*E$

$E \rightarrow (E)$

$I \rightarrow a$

$I \rightarrow b$

$I \rightarrow Ia$

$I \rightarrow Ib$

$I \rightarrow I0$

$I \rightarrow I1$

1^*

$\epsilon, 1, 11, 111, \ldots$

$E \rightarrow \epsilon$

$E \rightarrow 1$

$\underbrace{E}_{} \rightarrow E1$ → nonterminal

$E \rightarrow E11$

$\rightarrow E111$

$\rightarrow 1111$

a+b এর জন্য $E \rightarrow \epsilon$ নেওয়া যাবে না

$a+b$

$E \rightarrow E+E$

$I+I$

$aa + bba$

bba

$\rightarrow Ia$

$\rightarrow Iba$

$\rightarrow bba$

a*a*b*b

$E \to E*E$

$\to I*E$

$\to Ia*E$

$\to aa\,I$

$\to aaIb$

$\to aabb$

$E \to E*E$

$\to E*I$

$\to E*Ib$

$\to E*Ibb$

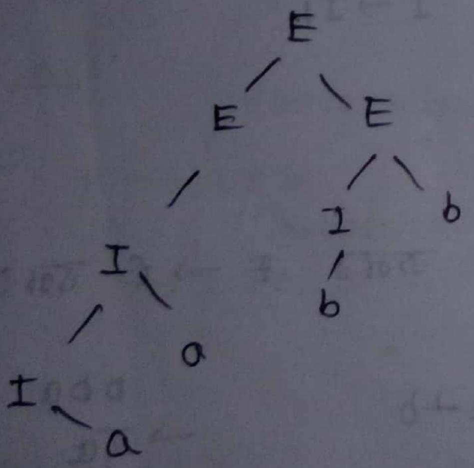
$\to E*Iabb$

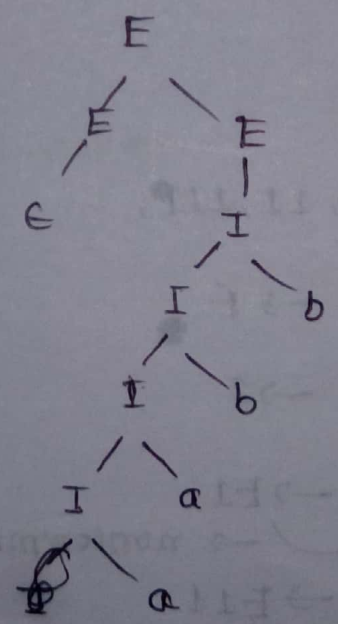
$\to E*aabb$

$\to \in aabb$

$\to aabb$

এটা part tree





Left most derivative / Right most derivative:

Grammer: Rule:

Rules:

$E \to E+E$

$E \to E*E$

$E \to (E)$

$E \to I$

$I \to a$

$I \to b$

accepted:

$a+b$

$a*(a+b)$

$a*(b)*b$

$a*b+a*b$

$a+b$

$E \to E+E$ (left এরটা replace করলে leftmost

$\to I+E$

$\to a+E$ Left most derivative

$\to a+I$

$\to a+b$

$E \to E+E$

$\to E+I$

$\to E+b$ Right most derivative

$\to I+b$

$\to a+b$

Parse Tree

$(a+b)*a$

$E \rightarrow E*E$

$\rightarrow (E)*E$

$\rightarrow (E+E)*E$

$\rightarrow (I+E)*E$

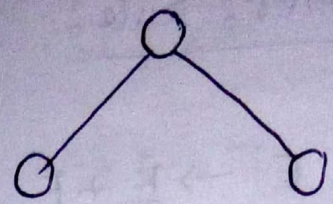
$\rightarrow (a+E)*E$

$\rightarrow (a+I)*E$

$\rightarrow (a+b)*E$

$\rightarrow (a+b)*I$

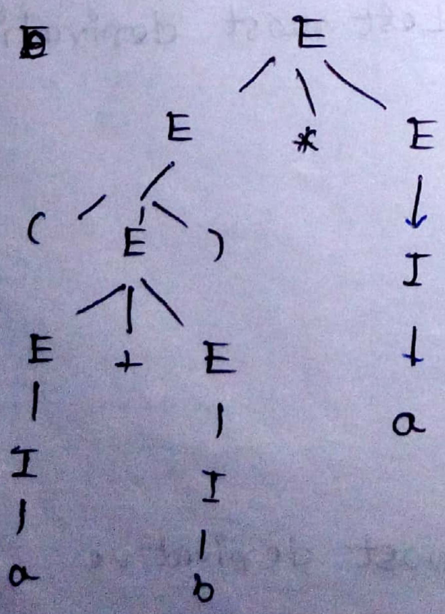
$\rightarrow (a+b)*a$



child

Left এর জন্য যে parse tree পাওয়া যাবে, তা যদি right parse tree এর সাথে না মিলে তাহলে তাকে ambigu parse।

Left most, right most এর জন্য parse tree এর structure change হবে না।



Right most

$* E \to E * E$

$* \quad \to E * I$

$\to E * a$

$\to (E) * a$

$\to (E+E) * a$

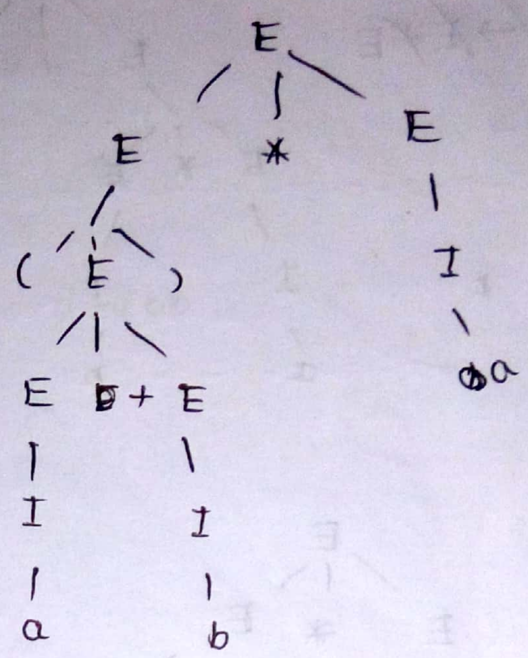
$\to (E+I) * a$

$\to (E+b) * a$

$\to (I+b) * a$

$\to (a+b) * a$

parse tree



$\boxed{a*b+b}$ Right most

$* E \to E + E$

$\to E + I$

$\to E + b$

$\to E * E + b$

$\to E * I + b$

$\to E * b + b$

$\to I * b + b \to a * b + b$

$E + E$

$\to E * E + E$

$\to E * E + I$

$\to E * E + b$

$\to E * I + b$

$\to E * b + b$

$E * E$

$E * E$

$E * E + E$

$E * E + I$

$E * E + b$

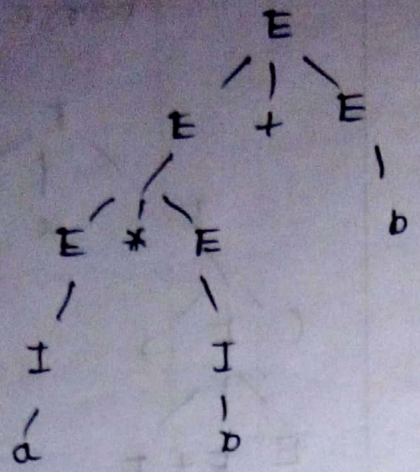
$E * b + b$

I

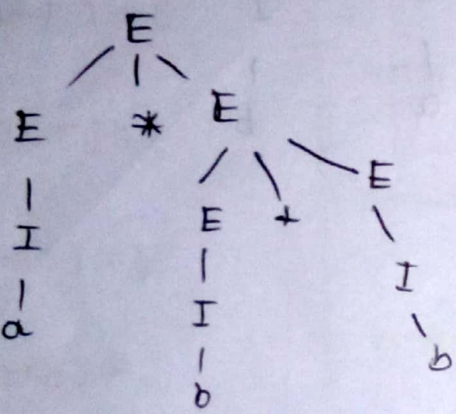
~~Left most~~

~~E → E*E~~

→



Ambiguous



Left most:

E → E+E

→ E*E+E

→ I*E+E

→ a*E+E

→ a*I+E

→ a*b+I

→ a*b+b

☐ bracket থাকলে ambiguous হবে না।

☐ ambiguity দূর করতে presidency create করতে হবে।

(a+b) * (b+c) * (b+c)
ambiguous

((a+b) * (b+c)) * (b+c)

"227 page"

Removal ambiguity.

~~⊞~~ operator replace

$S \to S a S$

$S \to Sa$ $\;\to\; S \to SS/Sa$

$S \to SS/Sa/bS/\epsilon$

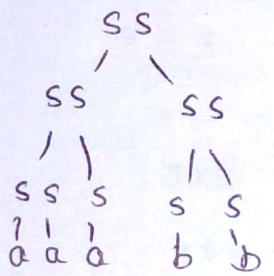
$S \to ASB \mid AS \mid BS$ aaabb

$S \to a \mid b \mid c$

$S \to SS \mid \epsilon$

$S \to ASB$

$S \to a \mid b \mid c$



$S \to /\epsilon$

$S \to \cancel{SS}/\cancel{\epsilon}$

$S \to aSb$

aSb
aaSbb] জোড়সংখ্যক হলে accepted...

$3 = 2 * 3 \div 2$ | Left এ যে থাকবে তার কাজ আগে হবে।

$2 \div 3 * 2$

$= 4/3$

Sentential Form :

Derivations from the start symbol produce

strings that have a special role.

বাদ না দিলে first টা starting role

$\to E \to E*E$

$E \to E*E$

$\to E*(E)$

$\to E*(E+E)$

$\to E*(I+E)$

$E \to E*E$
$E \to E+E$
$E \to (E)$
$E \to I$
$E \to a$
$E \to b$
$E \to e$

$E \to E+E$

$\to E*E+E$

$\to a*E+E$

not sentential,

middle থেকে শুরু হয়েছে

starting থেকে হয় নি।

$E \to E*E$	$E \to E*E$
$\to I*E$	$\to E*I$
$\to a*E$	$\to E*a$
Left	Right

5.1.7

Exercise 5.1

a) $L = \{0^n 1^n \mid n \geq 1\}$

$L = 0^1 1^1$

$= 01$

$L = 0^2 1^2$

$= 0011$

$L = 0^3 1^3$

$= 000111$

⋮

$L = 0^n 1^n$

CFG:

$A \rightarrow C01D$

$C \rightarrow 0C$

$D \rightarrow 1D$

$C \rightarrow \epsilon$

$D \rightarrow \epsilon$

$L = 0^3 1^3$ এর জন্য

$A \rightarrow C01D$

$\rightarrow 0C011D$

$\rightarrow 00C0111D$

$\rightarrow 00\epsilon0111\epsilon$

$\rightarrow 000111$

$A^i D^j c^k$ $\quad i \neq j$, $j \neq k$

abc ×

abbc ×

aaabcc ✓

$$RE \rightleftarrows CFG$$

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

$= (\epsilon, 0, 00, 000 \ldots) \; 1 \; (\epsilon, (0+1), (0+1)(0+1) \ldots$

$= (1, 01, 001 \ldots) \; (\epsilon, 0, 1, 00, 01, \ldots$

$= 1, 10 \ldots$ $\quad$ [No ϵ], minimum 1 থাকবে]

=

S → A1B

A → ε|0A|A0 ↷ RE

ε, 0A শুধু হবে 0*

B → ε|0B|1B

(5.1.7 – 5.1.8) Imp

$\boxed{5.2}$, 5.3, 5.4

⊞ একটা grammer এ কয়টি parse tree হবে?

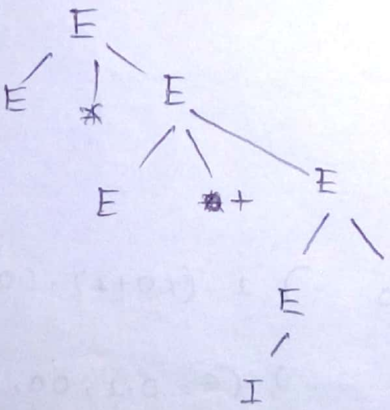
E → E*E

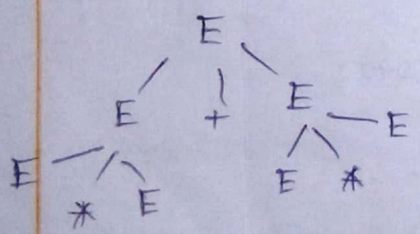
E → E*E

I → a

$\boxed{a*a+a}$ → দুটা parse

a*a+a*a





SC

Chapter 04:

$\boxed{\text{Properties}}$ of Regular Language.

✓ Closure properties

✓ Decision properties

→ minimize automata

→ switching circuit
cost reduction

4.1: Proving Language
Not to be Regular

** Pumping Lemma:-

Prove not every languag is a regular language.

Theorem 4.1:

$L = xy^k z$

1. $y \neq \epsilon$
2. $|xy| \leq n$
3. $k \geq 0$, string $xy^k z$ is L.

$L_1 = (01)^*$

$= \epsilon, 01, 0101,$

$L_2 = 1^* = \epsilon, 1, 11 \ldots$

$L_1 * L_2$ $\qquad L_1 + L_2 \longrightarrow$ Regular or not?

$=$

RE:

RL:

$(01)^* = \epsilon, 01, 0101 -$] not same

$(10)^* = \epsilon, 10, 1010 - \ldots$

same
$(0+1)^* = \epsilon, (0+1), (0+1)(0+1) = \epsilon, 0, 1, 00, 01, 10, 11$
$(1+0)^* = \epsilon, (1+0), (1+0)(1+0) = \epsilon, 1, 0, 11, 10, 01, 00$

minimise automata...

$(1+0)^* + (0+1)^*$ | $R^* + R^* = R^*$

$= (1+0)^*$

$= (0+1)^*$

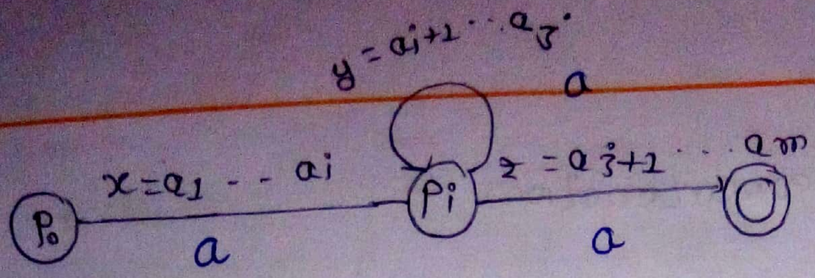
$L = xy^kz$ $\qquad k = n$

1. $y \neq \epsilon$

2. $|xy| \leq n$

3. $k \geq 0$; string xy^kz is L.

$y = a_{i+1} \cdots a_j$



aa → no looping edge = 2

$|aaa| = 3$ $p_i = j = n$

input string $= n-1$

1^* → loop / closure ...

4.1.2

Application of the pumping Lemma.

Ex 4.1.3

4.2 closure properties of Regular Language.

1. Union of two regular language is regular.

2. The Intersection of two regular language

3. " complement

4. " Differences

5. reversal

6. closure

7. " concatenation

8. " homomorphism.

reg Regular reg
L(R) = L(A) ∪ L(B)

↓ ↓

1^* 1^*

Reverse $(10)^* = \epsilon, 10, 10, 10, = \epsilon, 01, 0101 = (01)^*$

4.2.2

$w = a_1 a_2 \ldots a_{n-1} a_n$

$w^R = a_n a_{n-1} \ldots a_1$

L = = {001, 10, 111}

L^R = {100, 01, 111}

4.2.1 (L যদি regular হয় L^R regular)

L = ϵ, ϕ, a

$L^R = \epsilon^R = \epsilon$

$L^R = \phi^R = \phi$

$L^a = a$

$E = E_1 E_2$

$E_1 = (00, 10)$ $\quad$ $E_2 = \{10, 11\}$

$E = 0010, 0011, 1010, 1011$

$E^R = \{0100, 1100, 0101, 1101\}$

আগে element reve

$E_1^R = (00, 01)$

$E_2^R = (01, 11)$

$E = E_2^R E_1^R$

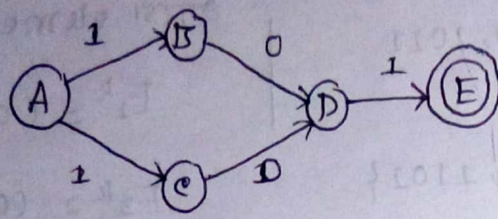
$= (01, 11) * (00, 01)$

$= \{0100, 0101, 1100, 1101$

8D

✓ closure property

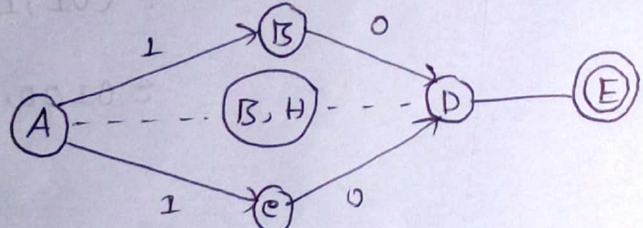
✓ Decision property ⟶ test Disgwishable state or not



101] same
101

B, C, equivalent

starting, ending node বাদে বাকী node গুলো check করতে হবে, same কিনা!



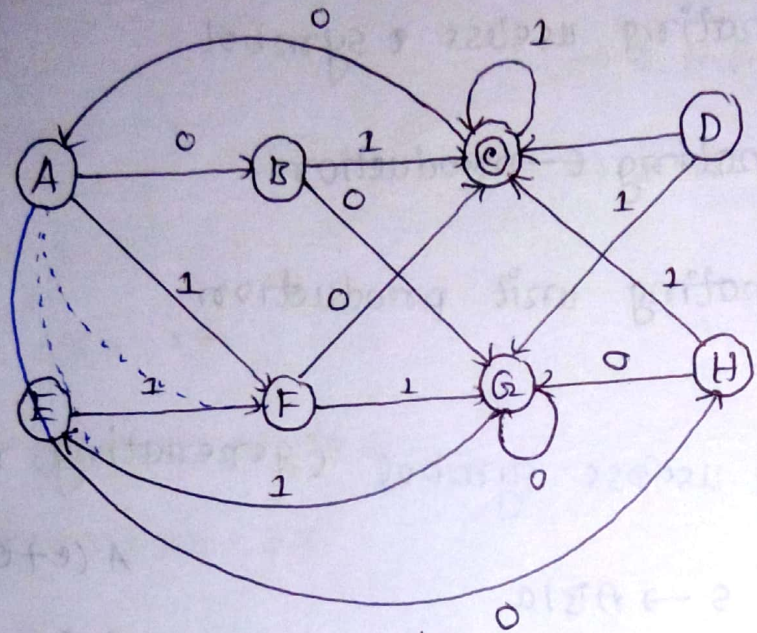
A, B
A, C
A D
A E
B C
B D
B E

□ E accepting state. E এর সাথে কোনোটাই equivalent হবে না।

□ বাকীগুলো replace হতে পারে।

4.4 Equivalence and Minimization of Automata

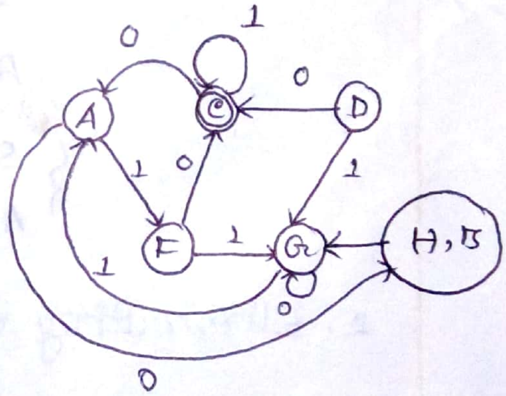
4.4.1 Testing equivalence of states

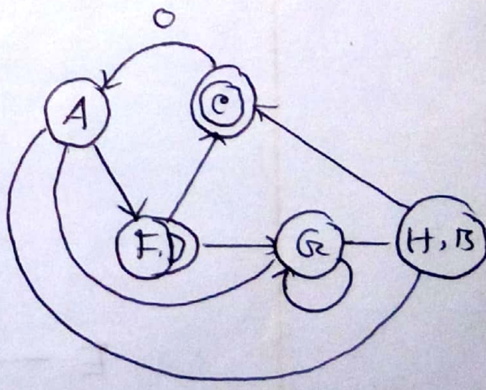


$$\frac{n(n-1)}{2}$$

$$=28$$

	A	B	C	D	E	F	G
B	×						
C	×						
D	×		×				
E	O		×				
F	×		×	O			
G	×		×				
H	X	O	×				





$\boxed{4.4.2}$, 4.4.3

181 , 183 (4.4.1 , 4.4.2 - 3 ,

Properties of context Free grammer

7.1: Normal form of CFG

1) Eliminating useless symbol

2) Eliminating ϵ-production

3) Eliminating unit production

1. Eliminating useless symbol: (generating, reachable)

$S \to AB \mid a$

$A \to b$

$S \to a$

$A \to b$

$A(C+\overline{C}) + \overline{A}C$

$\Rightarrow AC + A\overline{C} + \overline{A}C$

2. Eliminating ϵ-production:

$S \to AB$

$A \to aAA \mid \epsilon$

$B \to bBB \mid \epsilon$

$S \to AB \mid A \mid B$

$A \to aAA \mid aA \mid aA \mid a$

$B \to bBB \mid bB \mid bB \mid b$

$E \to E * E$

$E \to E + E$

$E \to I$

$E \to a/b$

$A \to Ab$ recursive

1. $S \rightarrow AB$

2. $A \rightarrow aAA \mid \epsilon$

3. $B \rightarrow bBB \mid \epsilon$

$S \rightarrow AB \mid AB \mid AB$

$S \rightarrow AB \mid B \mid A$

$A \rightarrow aAA \mid \epsilon$

$\rightarrow aAA \mid a\epsilon A \mid a\epsilon A\epsilon \mid a\epsilon\epsilon$

$\rightarrow aAA \mid aA \mid aA$

$B \longrightarrow bBb \mid \epsilon$

$\longrightarrow bBB \mid b\epsilon B \mid bB\epsilon$

$\rightarrow bBB \mid bB \mid bB$

$\rightarrow bBB \mid bB$

$S \rightarrow AB \mid \underset{\epsilon}{A}B \mid A\underset{\epsilon}{B}$

$\rightarrow AB \mid B \mid A$

$A \rightarrow aAA \mid aA \mid aA \mid a$

7.1.4

Eliminating Unit Production:

I → a | b | Ia | Ib | I0 | I1

F → I | (E)

T → F | T*F

E → T | E*F

nonter ← → nonterm / terminal

| non ← non
unit production

A → B unit production

* unit production এ variable থাকবে না

A → B
B → a | accepted string a

pair	production
(E, E)	E → E*F
(E, T)	E → T*F
(E, F)	E → (E)
(E, I)	E → a | b | Ia | Ib | I0 | I1
(T, T)	T → T*F
(T, F)	T → (E)
(T, I)	T → a | b | Ia | Ib | I0 | I1
(F, F)	F → (E)
(F, I)	F → a | b | Ia | Ib | I0 | I1
(I, I)	I → a | b | Ia | Ib | I0 | I1

$E \rightarrow E * F \mid T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$T \rightarrow T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$F \rightarrow (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$I \rightarrow a \mid a \mid Ia \mid Ib \mid I0 \mid I1$

293

7.1.1 , Exercise all...

$A \rightarrow B$

$B \rightarrow C$

$C \rightarrow A$

7.1.5

CFG without ϵ has a grammar G in which all productions are in one of two simple forms either:

1. $A \rightarrow BC$ where, A, B, and are each variable

2. $A \rightarrow a$, where A is a variable and a is a terminal

Further, G has no useless symbol

a grammar is said to be in (Chomsky Normal Form) (CNF)

Rogue   200, 100, ... 200, 0 → Rogue

$A \to BC$ (nonterminal production থাকলে)

Palindrome checker:

$P \to \epsilon$

$P \to 0$

$P \to 1$

$P \to 0P0$

$P \to 1P1$

1010   aabb

11011

$S \to aSbS$

$S \to bSaS$

$S \to \epsilon$

a যতবার ঘটবে b ততবার ঘটবে।

aaabbb

abababababab

~~ASbS~~   aSbS

aaSbS bS

a$\boxed{SbS}$

[aaabba

a SbS

aSbS bS

aSbS

aaaSbSbSbS

aaaSbS

98

7.1.5

Chomsky Normal Form (CNF):

1. $A \rightarrow BC$

2. $A \rightarrow a$

Tasks:

a) Arrange all bodies of length 2 or more consists of variables

b) Break bodies of length 3 or more into a cascade of productions ~~such~~ each variable body consists of two variable.

$A \rightarrow BCD$

$\boxed{C_1 \rightarrow CD}$

$A \rightarrow BC_1$ (rule 1)

$A \rightarrow \underline{BC}\ \underline{DE}$

$A \rightarrow C_1 C_2$

*

$I \rightarrow a | b | Ia | Ib | I0 | I1$

$F \rightarrow I \mid (E)$

$T \rightarrow F \mid T*F$

$E \rightarrow T \mid E+T$

After eliminating unit production...

$E \rightarrow E*T | T*F | (E) | a | b | Ia | Ib | I0 | I1$

$T \rightarrow T*F | (E) | a | b | Ia | Ib | I0 | I1$

$F \rightarrow (E) | a | b | Ia | Ib | I0 | I1$

$I \rightarrow a | b | Ia | I0 | I1$

295 pg

$E \rightarrow F + T$ $\qquad$ $E \rightarrow FL$

$G \rightarrow +$ $\qquad$ $L \rightarrow GT$

then $E \rightarrow F\underline{GT}$ $\qquad$ $G \rightarrow +$

$L = GT$

then $E \rightarrow FL$

$A \rightarrow a$

$B \rightarrow b$

$Z \rightarrow 0$

$O \rightarrow 1$

$P \rightarrow +$

$M \rightarrow *$

$L \rightarrow ($

$R \rightarrow)$

$E \rightarrow EPT \mid TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$

$T \rightarrow TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$

$F \rightarrow LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$

{ $C_1 \rightarrow PT$ $\qquad$ $C_2 \rightarrow MF$ $\qquad$ $C_3 \rightarrow ER$

$E \rightarrow EC_1$ $\qquad$ $E \rightarrow TC_2$ $\qquad$ $E \rightarrow LC_3$

$T \rightarrow TC_2$ $\qquad$ $T \rightarrow LC_3$

$F \rightarrow LC_3$

$E \rightarrow EC_1 \mid TC_2 \mid LC_3 \mid a \mid b \mid 1$

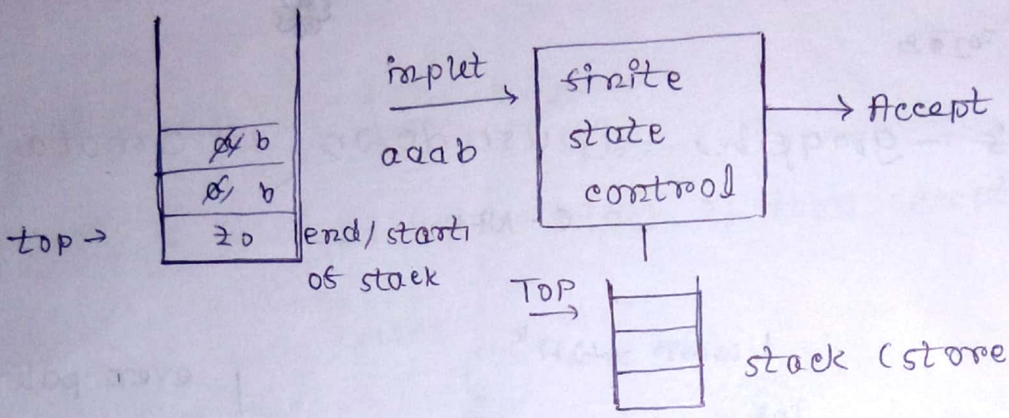
Scanned by CamScanner

chapter 06

Pushdown Automata:

CFG : (ϵ-NFA with stack)

aabb



stack (store করবে কিনা সেটা depend করছে stack এ previously কিছু আছে কিনা

$L = \{a^n b^n \mid n \geq 1\}$

ab

aabb ✓

abab ×

aabbb → underflow

push a | a | a |

pop a and push b | a | b |

push a | a | a |

6.1.2

The ~~formal~~ Definition of pushdown Automata

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

$\delta(q_0, 0, z_0) = (q, 0z_0)$

$P = \{Q, \Sigma, \Gamma, \delta, q_0, z_0, F\}$

$Q \rightarrow q_0\, q_1\, q_2$

$\Sigma \rightarrow a, b$

Γ

z_0, a, b

0
z_0

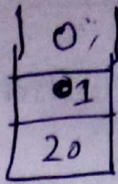
20

<OCR>

$w = 10$

6.2

$w^R = 01$



$L = ww$

4 $= 1001$

6

8

$= 01\,0\,1$

6.1.3 – graph .. (push down automata)

ϵ – NFA

9c $L_{wwr} = ww^R$ | even palindrom

Let, $w = 10$, $w^R = 01$

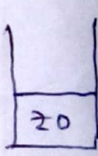
Example 6.2:

$L_{wwr} = 10(10)^R$ | current – next (loop)

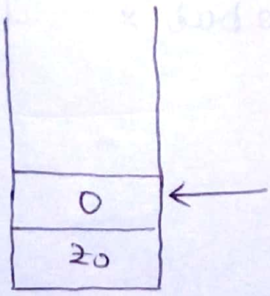
1. $\delta(q_0, 0, z_0) = \{(q_0, 0z_0)\}$

current → next

current state $= q_0$ top & stack $= z_0$ (see)



push it onto stack leaving top..



2. $\delta(q_0, 0, 0) = \{(q_0, 00)\}$, $\delta(q_0, 0, 1)$

$= \{(q_0, 01)\}$

$\delta(q_0, 1, 0) = \{(q_0, 10)\}$, $\delta(q_0, 1, 1) = \{(q_0, 11)\}$

current state $= q_0$, push (stack), leaving top & stack

3. $\delta(q_0, \epsilon, z_0) = \{(q_1, z_0)\}$, $\delta(q_0, \epsilon, 0) = \{(q_1, 0)\}$

stack q_0 = state q_1, leaving stack (intact)

Scanned by CamScanner

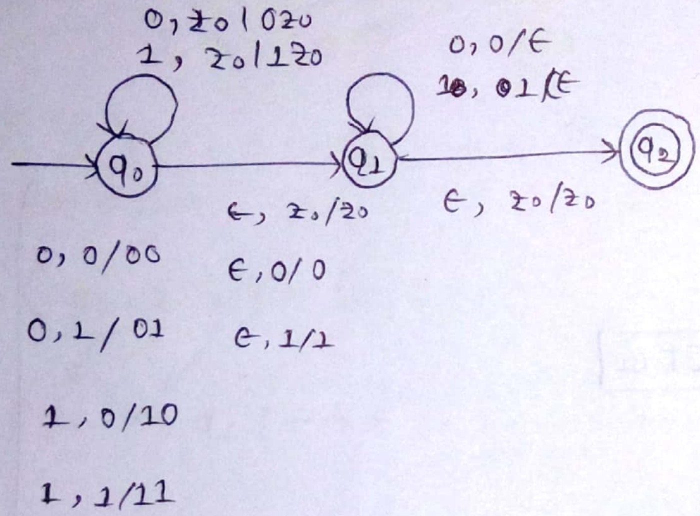
4. $\delta(q_1, 0, 0) = \{(q_1, \epsilon)\}$

$\delta(q_1, 1, 1) = \{(q_1, \epsilon)\}$

q_1-state match, pop of stack

5. $\delta(q_1, \epsilon, z_0) = \{(q_2, z_0)\}$

Finally $q_1 = q_2$ and top of stack z_0 then accept

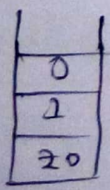


10/01/ε → middle খুঁজতে হবে। (matching খুঁজে, অর্ধেক করে

$1^n 0^n$

111000

যতক্ষণ 1 পাবো push করব।



** input signal এর সাথে top match করলে pop.

10c "Automata"

6.7:

Equivalence of PDA and CFG

1) CFG to PDA (push down automata)

2) PDA to CFG

Epsilon move

(1) CFG to PDA

11B

PDA to CFG

Given,

$M = \{q_0, q_1\}, \{0, 1\}, \{z_0, X\}, \delta, q_0, z_0, \phi\}$

$\delta(q_0, 0, z_0) = (q_0, Xz)$ --- (1)

$\delta(q_0, 0, X) = (q_0\ XX)$ —— (ii)

push

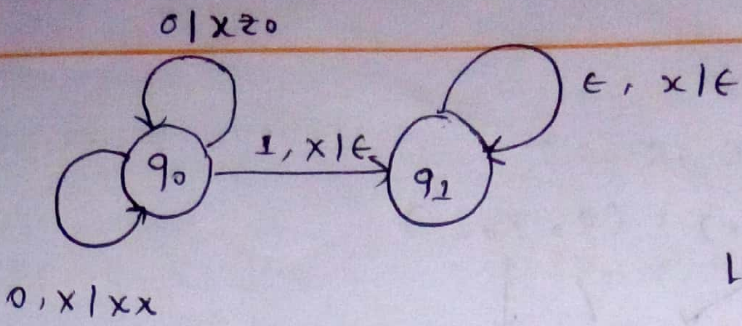
$\delta(q_0, 1, X) = (q_1, \epsilon)$ —— (iii)

$\delta(q_0, 1, \cancel{X}) = (q_1, \epsilon)$ —— (iv)

$\delta(q_0, \epsilon, X) = (q_1, \epsilon)$ —— (v)

$\delta(q_0, \epsilon, z_0) = (q_1, \epsilon)$ —— (v)

pop



Language

$L = \{0^m, 1^n \mid m \geq n\}$

Step: 01

push operation:

For δ S (head); ~~start~~ start initial top of stack stack change

$S \rightarrow [q_0, z_0, q_0] \longrightarrow R_1$ Rules R_1

$S \rightarrow [q_0, z_0, q_1] \longrightarrow R_2$

POP operation:

$\delta(q_0, 1, x) = (q, \epsilon)$

$[q_0, x, q_1] \longrightarrow \epsilon$ - - - - (R_3)

$\therefore \delta(q_1, 1, x) = (q_1, \epsilon)$

$\therefore [q_1, x, q_1] \rightarrow \epsilon$ - - - (R_4)

Similarity:

$[q_1, x, q_1] \longrightarrow \epsilon$ - - - (R_5)

$[q_1, z_0, q_1] \rightarrow \epsilon$ - - - (R_6)

$\boxed{\text{step 02}}$

push 1:

$\delta(q_0, 0, z_0) = (q_0\ x_0 z_0)$

$[q_0\ z_0\ , \underline{q_0}] = 0\ (q_0, x, \_\ ]\ [\_\ z_0\ \underline{q_0}$

For $\delta(q_0, 0, z_0) = (q_0, x z_0)$

→ recursive

$[q_0, z_0, q_0] = 0\ [q_0, x, q_0]\ [q_0, z_0, q_0]$

A A

same state ← $[q_0, z_0, q_0] = 0\ [q_0, x, q_1]\ [q_1, z_0, q_0]$ (no loop)

$[q_0, \cancel{z_0}, q_1] = 0\ [q_0, x, q_0]\ [\cancel{q_0, z_0}, q_1]$ → recursive

$[q_0, z_0, q_1] = 0\ [q_0, x, \underline{q_1}]\ [\underline{q_1}, z_0\ q_1]$

same for push 2. . . and other. . .

push (2)

$\delta(q_0, 0, x) = (q_0, xx)$

$[q_0, x, q_1] = 0\ [q_0, x, q_1]\ [q_1, x, q_1]$

$S \to [q_0, z_0, q_1]$ → start

$[q_0, z_0, q_1] = 0\ [q_0, x, q_1]\ [q_1, z_0, q_1]$

$[q_0, x, q_1] = 0\ [q_0, x, q_1]\ [q_1, x, q_1]$

$[q_0, X, q_1] \rightarrow 1$

$[q_1, X, q_1] \rightarrow \epsilon$

$[q_1, z_0, q] \rightarrow \epsilon$

$[q_0, z_0, q_1] = D$

$[q_0, X, q_1] = E$

$[q_1, z_0, q_1] = H$

$[q_1, X, q_1] = R$

Result:
$S \rightarrow D$

$D \rightarrow 0EH$

$E \rightarrow 0EF \mid 1$

$H \rightarrow \epsilon$

$F \rightarrow \epsilon$

6.3 : Equivalence of PDA and CFG

1) CFG to PDA (push down automata)

2) PDA to CFG

1) CFG to PDA :

A → aBb → push হবে

Rule 01 : For each variable A, $\delta(q, \epsilon, A) = \{(q_2, B)\} \mid A \to B$ is production of G

Rule 02 : For each terminal {a b c d}

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

// input আর top of stack match করলে pop করব ∈ দিয়ে

stack থেকে pop করেছি

S

Epsilon move :

$\delta(q, \epsilon, S) = (q, aSb)$

$\delta(q_0, \epsilon, z_0) = (q_1, b)$

pop move :

$\delta(q, 0, 0) = \{(q_1, \epsilon)\}$

$\delta(q, a, w) \to (q, \epsilon)$

// variable যদি right side এ থাকে কিন্তু left side এ নাই তাহলে সেটা useless.

$L = \{ a^n b^n \mid n \geq 0 \}$

$S \to aSb \mid ab \mid \varepsilon$

$n \geq 1$

$S \to aSb \mid ab$

$S \to aSb$

$S \to ab$

CFG to PDA:

Rule 1: $S \to aSb$

$S \to ab$

$\delta_1 (q, \varepsilon, S) = \{(q, aSb)\}$

$\delta_2 (q, \varepsilon, S) = \{q; ab)\}$

যদি ε থাকে, তবে $\delta_3 (q, q, \epsilon) = \{(q, \epsilon)\}$

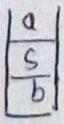
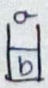
Rule 2:

$\delta_3 (q, a, a) = \{(q, \epsilon)\}$

$\delta_4 (q, b, b) = \{(q, \varepsilon)\}$

$\delta_5 (q, \varepsilon, \varepsilon) = \{(q, \epsilon)\}$

input aaa bbb → accept?

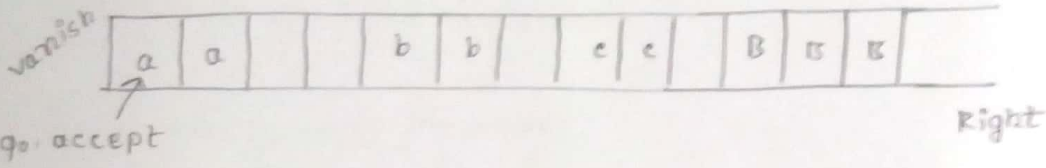
ε aaabbb   ab নিলে accepted হচ্ছিলো না।

6.12, 6.3.3 ⊛ nonterminal যত pop rule হবে তত।

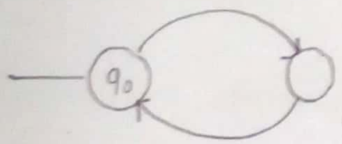
Turing Machine Design:

$L = \{a^n b^n c^n \mid n \geq 1\}$

$w = \{abc, aabbcc, aaabbbccc, \ldots\}$

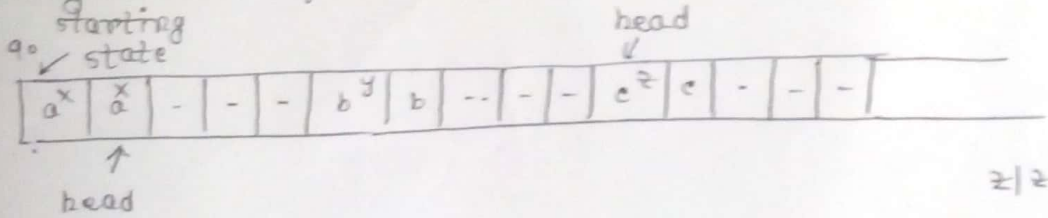


$a^n b^n$



NFA/DFA

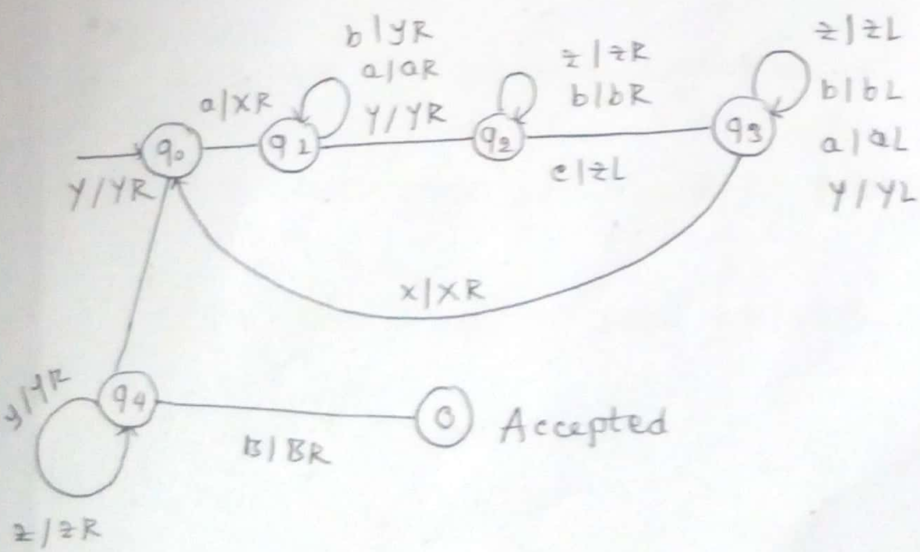
Turing Machine এর transition graph:



z|zL
b|bL
a|aL
y|yL

যা আছে তাই দিয়ে replace করে left এ যাবে।

exercise 8.2.5, ~~8.2.~~ 8.2.1, 8.2.2



* Regular Expression → convert

chapter 09:

* conversion

Undecidability:

(9.1) Recursively Enumerable

(9.2) Undecidable Problem

(9.3) Undecidabe Problem about Turing machine

(9.4) post's correspondence Problem.

(9.5) Other Undecidable Problems

chapter 10

10.1 classes P and NP problem

10.2 An NP - complete complement

10.3 Restricted satisfiability Problem

10.4 Additional NP-hard problem.

Scanned by CamScanner

4+5+1+4 $\boxed{a\,1\,4+16+16\ a}$

DFA 0,1 ছাড়া accepted

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

$L = \{a^n b^n \mid n \geq 0\}$ ⟶ অবস্থানেনা

$a^n b^n c^n$

$a^n b^m$

$S \to aSb \,/\, aS \mid a \mid b \mid Sb$

$a^n b^m$

$n \geq 2 \quad m \geq 2$

Language ⟶ DFA | PDA | CFG

$S \to aSb \mid aabb \mid$

$a^m b^n c^p \qquad m = n = r$

$S \longrightarrow$

Automata

11C

Chapter 08

tractable : সহজে solve

Introduction to Turing Machine:

Undecidable problem (নির্দিষ্ট টাইমে solve করতে পারে?)
docile

Intractable problem → (NP-hard) অনেক টাইম নিয়ে solve

Docile : সহজে চরানো

$x^n + y^n = z^n$

$n = 1$ $x = 3$ $y = 4$ $z = 5$

"solve করার জন্য

$\boxed{n = 2}$ $x = 3$ $y = 4$ $z = 5$

$n \times n \times n \rightarrow$ infinity

টাইম লাগবে।"

Notation for the turing machine:

string

$0^n 1^n$

01

0011

Finite control

..	B	B	X_1	X_2	- - -	X_i	- - -	X_n	B	B	- - - infinity

active region

tape symbol

tape head (initially blank)

Scanning

move : left to right

Right to left

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

Q = set of state

Σ = input symbol (সকল variable)

Γ : tape symbol (B ও tape symbol এর মধ্যে পড়ে)
symbol in Γ

input

$\delta = \delta(\underline{q, X}) = (P, Y, D) \rightarrow$ output

next state

Direction
(R, L)

q_0 = Final state

$q \rightarrow$

B = Blank symbol

F = Find state