

AFLL

UNIT - 2

CLASS NOTES

feedback/corrections: vibha@pesu.pes.edu

Vibha Masti

REGULAR EXPRESSIONS

Representing Regular Languages

- DFA
- NFA
- λ -NFA
- Regular expressions

Regular Expression

- pattern that contains a sequence of characters that is matched against some text

1) Atoms

- what you are looking for
- where in the text

2) Operators

- combine atom & operators - complex expression

Atoms

$$(A) \Rightarrow \Sigma \rightarrow L(A) = \{A\}$$

$\emptyset \rightarrow$ RE that denotes empty language (set)
 $L(\emptyset) = \{\}$

$\lambda / \epsilon \rightarrow$ RE that denotes empty string
 $L(\lambda) = \{\lambda\}$

Operators

R_1 and R_2 — L_1 and L_2

1) Union (+)

$R_1 + R_2 = L_1 \cup L_2$ (order does not matter)

2) Concatenation (.)

$R_1 \cdot R_2 = L_1 \cdot L_2$ (order matters)
 L_1 followed by L_2

3) Closure (*)

$R^* = L^*$ (0 or more times)

4) Incomplete closure (+)

$R^+ = L^+$ (1 or more times)

Precedence

- 1) Brackets ()
 - 2) Closure *, +
 - 3) Concat .
 - 4) Union +
- 

Algebraic Laws

Identity for operator

operator applied to identity & another value, result is the other value

Question 1

Construct RE over $\Sigma = \{a,b\}$ that accepts a string that starts with ab

$$L = ab(a+b)^*$$

Question 2

Construct RE over $\Sigma = \{a,b\}$ that accepts a string that contains a substring ab

$$L = (a+b)^* ab (a+b)^*$$

Question 3

Construct RE over $\Sigma = \{a,b\}$ that accepts a string that ends with ab

$$L = (a+b)^* \cdot ab$$

concatenation;
dot not req.

Question 4

RE that ends with ab or ba

$$L = (a+b)^*(ab + ba)$$

Question 5

RE, starts & ends with same symbol $\Sigma = \{a, b\}$

Case 1: $a \rightarrow a$
or

Case 2: $b \rightarrow b$

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

Question 6

RE, $\Sigma = \{a, b\}$, starts & ends with diff symbols

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

Question 7

third symbol from the start is a (left)

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

Question 8

2nd symbol from right (end) is a

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

Question 9

3rd symbol from right (end) is a and 4th symbol from right (end) is b

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

Question 10

At least one a and at least one b

$$L = \{ab, ba \dots\}$$

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

Question 11

At least one a in the first 3 symbols and at least one b in the last two symbols

min length: 3, but we take 5

$$\begin{aligned} & [(a(a+b)(a+b)) + ((a+b)a(a+b)) + ((a+b)(a+b)a)] \cdot (a+b)^* \cdot \\ & \quad \cdot [(a+b)b + b(a+b)] \end{aligned}$$

Question 12

length is exactly 2 , $\Sigma = \{a, b\}$

$$L = (a+b)(a+b) \quad (\textcircled{OR}) \quad (a+b)^2$$

Question 13

$|w| \geq 2$, $w \in \{a, b\}^*$

$$L = (a+b)^2 (a+b)^*$$

Question 14

$|w| \leq 2$, $w \in \Sigma^*$, $\Sigma = \{a, b\}$

$$L = \{\lambda, (a+b), (a+b)^2\}$$

$$L = \lambda + (a+b) + (a+b)^2$$

(OR)

$$L = (\lambda + a + b)(\lambda + a + b)$$

$$L = (\lambda + a + b)^2$$

generic - lengths

$$|w| = n, \quad L = (a+b)^n$$

$$|w| \geq n, \quad L = (a+b)^n (a+b)^*$$

$$|w| \leq n, \quad L = (\lambda + a+b)^n$$

Question 15

$$|w| \bmod 2 = 0$$

$$L = [(a+b)^2]^*$$

Question 16

$$|w| \bmod 2 = 1$$

$$L = (a+b)[(a+b)^2]^*$$

Question 17

$$|w| \bmod 3 = 0$$

$$L = ((a+b)^3)^*$$

Question 18

$$|w| \bmod 3 = 2$$

$$L = (a+b)^2 ((a+b)^3)^*$$

Question 19

$$\mathcal{L} = \{ n_a(w) = 2, w \in \{a,b\}^* \}$$

$$\mathcal{L} = b^* a b^* a b^*$$

Question 20

$$\mathcal{L} = \{ n_a(w) \geq 2, w \in \{a,b\}^* \}$$

$$\mathcal{L} = (a+b)^* a (a+b)^* a (a+b)^*$$

Question 21

$$\mathcal{L} = \{ n_a(w) \leq 2, w \in \{a,b\}^* \}$$

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

Question 22

$$n_a(w) \bmod 2 = 0$$

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

Question 23

$$n_a(w) \bmod 2 = 1$$

*

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

Question 24

* $n_a(w) \bmod 3 = 0$

$$b^*(b^*a b^* a b^* a b^*)^* b^* \text{ or } (b^*a b^* a b^* a b^*)^* + b^*$$

Question 25

$$n_a(w) \bmod 3 = 2$$

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

Question 26 note: $a^n b^n$: impossible as DFA / RE

$$L = \{a^n b^m \mid n+m \text{ is even}\}$$

even a's	even b's
odd a's	odd b's

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

Question 27

$$L = \{a^{2n} b^{2m} \mid n, m \geq 1\}$$

$$(aa)^+ (bb)^+$$

Question 28

$$\mathcal{L} = \{ a^n b^m \mid n, m \geq 1, \text{ product}(n, m) \geq 3 \}$$

No. of a's	No. of b's	≥ 3
Case 1	1	≥ 3
Case 2	≥ 3	1
Case 3	≥ 2	≥ 2

$$a\bar{a}^* b b b b^* + a a a^* b b b^* + a a a a^* b b^*$$

Question 29

$$\mathcal{L} = \{ a^n b^m \mid n \geq 4, m \leq 3 \}$$

$$aaaaa^* (\lambda + b + bb + bbb)$$

or

$$aaaaa^* (\lambda + b)^3$$

Question 30

$$\mathcal{L} = \{ a^n b^m c^k \mid n+m \text{ is odd and } k \text{ is even} \}$$

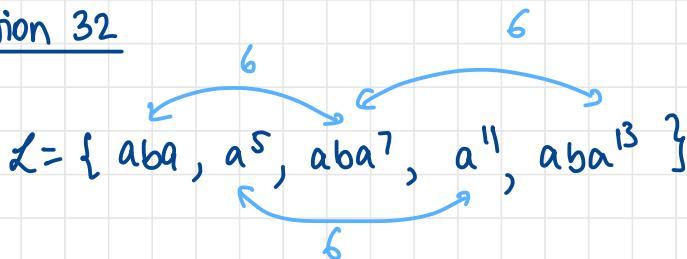
$$(aa)^* a (bb)^* (cc)^* + (aa)^* (bb)^* b (cc)^*$$

Question 31

$$L = \{a^n b^m c^k \mid n \leq 4, m \geq 2, k \leq 2\}$$

$$(a+a)^4 (bb)b^* (a+c)^2$$

Question 32



$$(aba + a^5)(a^6)^*$$

Question 33

$$L = \{uvw \mid u, v, w \in \{a, b\}^*, |u|=2, |w|=2\}$$

$$(a+b)^2 (a+b)^* (a+b)^2$$

Question 34

$$L = \{w \mid |w| \text{ divisible by 2, not by 3}\}$$

$$\left((a+b)^2 + (a+b)^4\right) \left((a+b)^6\right)^*$$

Question 35

Even no. of a's followed by even/odd b's

$$L = (aa)^* b^*$$

Question 36

Begin with at least 2 a's, end with even b's

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

Question 37

- * Accepts binary string such that if dec. eq. of the no. is even then the length of the string is even, if dec. eq. is odd then length of string is odd

$$(0+1^2)^* 0 + (0+1^2)^* 1$$

Question 38

$L = \{ \text{Binary string whose decimal eq. is twice an odd no} \}$

X

2 × odd	2 × 1
2 × 3	6
2 × 5	10
2 × 7	14

Dec.	2
6	
10	
14	

Bin	$(10)_2$
	(110)
	$(1010)_2$
	$(1110)_2$

ends with

10

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

Question 39

Bin string that does not have 2 consecutive 0s

$$(11^*0)^* + (011^*)^* + 1^* + 01^*$$

Question 40

Exactly one pair of consecutive 0s

$$(01+1)^* 00 (10+1)^*$$

Question 41

No consecutive a's and b's

$$(b+\lambda)(ab)^*(a+\lambda) + (a+\lambda)(ba)^*(b+\lambda)$$

Question 42

What is L for $(a+b)^* aa(a+b)^*$

strings with at least one pair '(aa)'

Equivalence of Regular Expressions

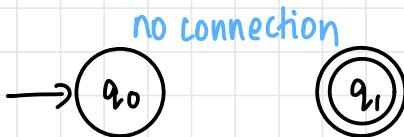


- Thompson Construction method : $RE \rightarrow FA$
- State elimination method : $FA \rightarrow RE$
- Kleene's Theorem

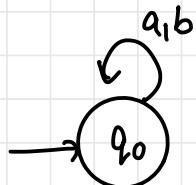
RE to FA

- RE to λ -NFA

i) RE accepts empty language \emptyset



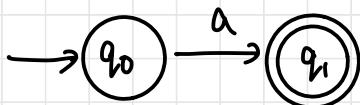
for $\Sigma = \{a, b\}$



2) RE accepts λ (empty string)



3) RE is single symbol a



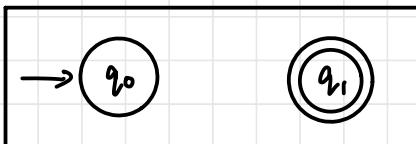
Without loss of generality,

R_1 and R_2

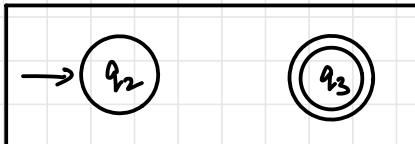
union	$R_1 + R_2$	$\rightarrow L_1 \cup L_2$
concat	$R_1 R_2$	$\rightarrow L_1 L_2$
closure	R_1^*	$\rightarrow L_1^*$

UNION

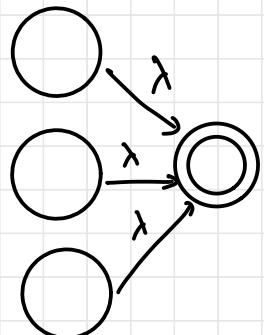
R_1

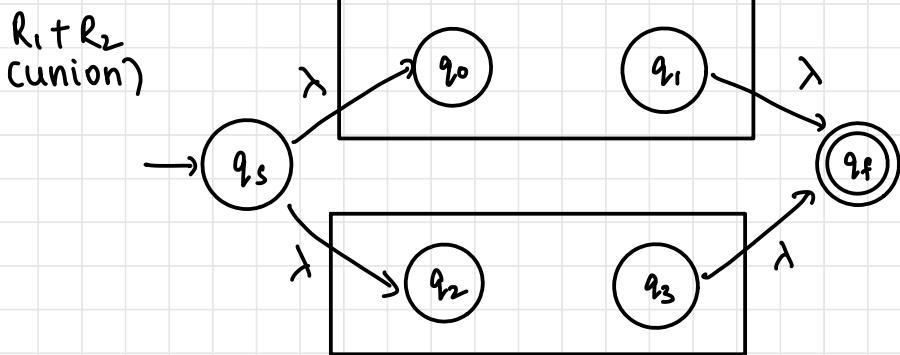


R_2

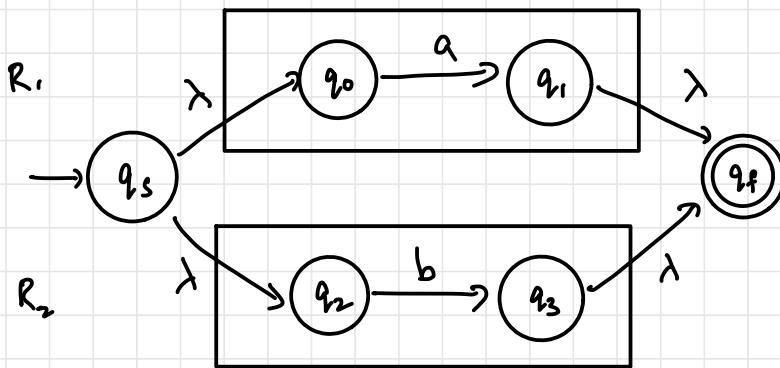


for > 1 FS

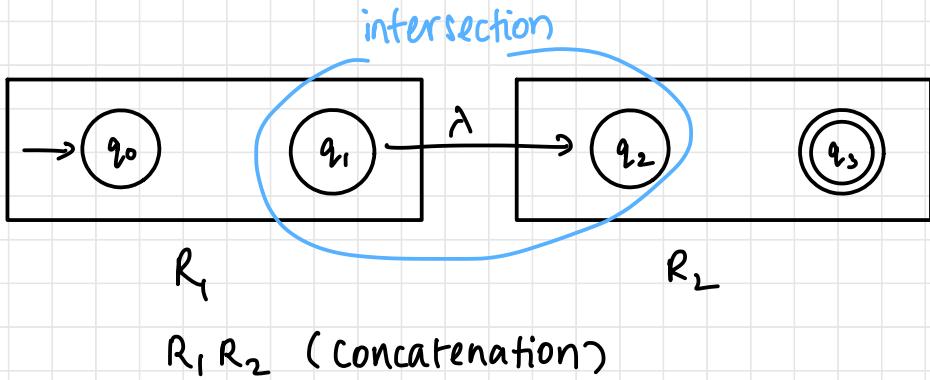




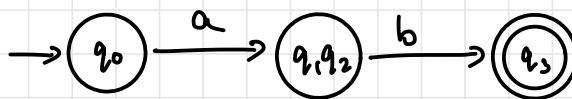
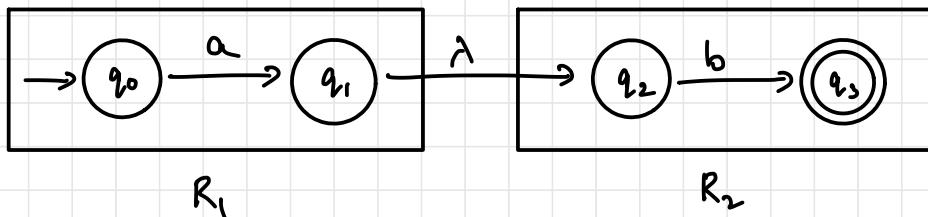
eg: $(a+b)$



CONCATENATION

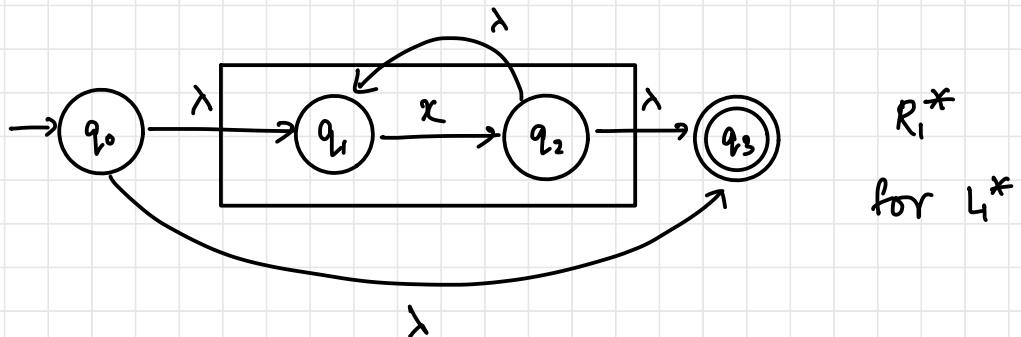


eg: (ab)



CLOSURE

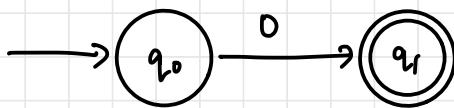
$$R_1^* \longrightarrow L_1^* = L_1^0 \cup L_1^1 \cup L_1^2 \cup \dots$$



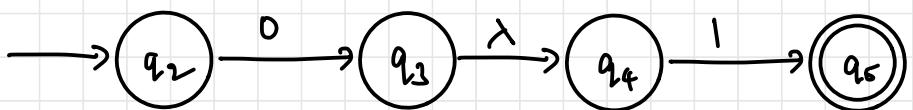
Question 43

convert $(0+01)^*$ using Thompson method

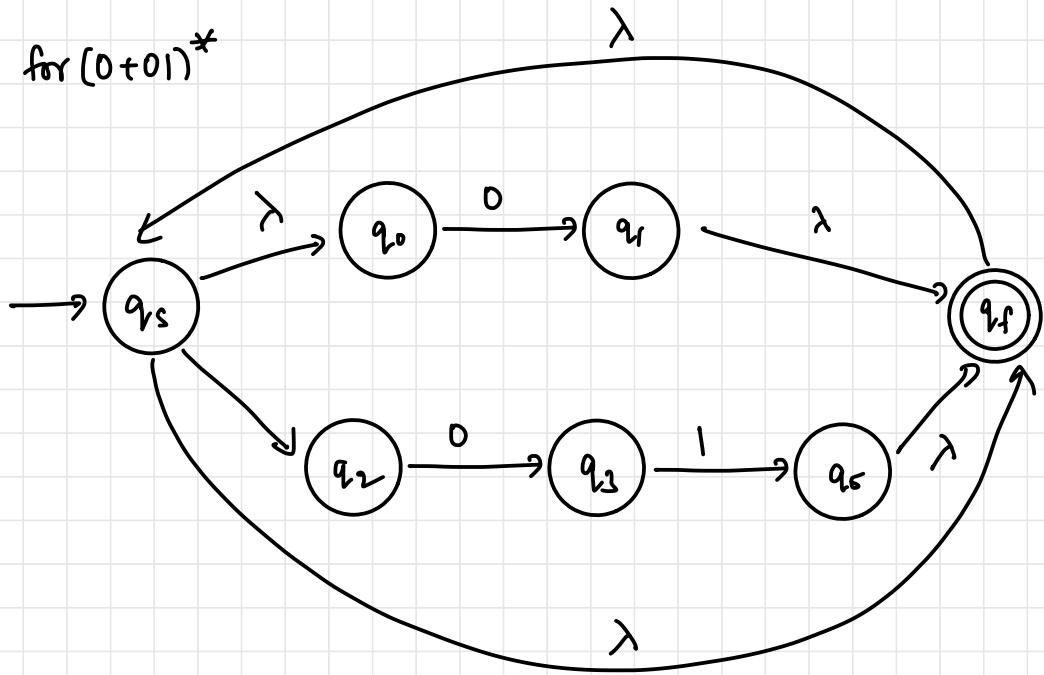
for 0



for 01

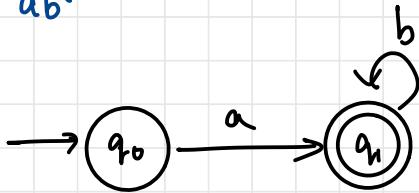


for $(0+01)^*$



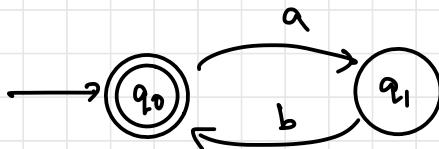
Question 44

ab^*



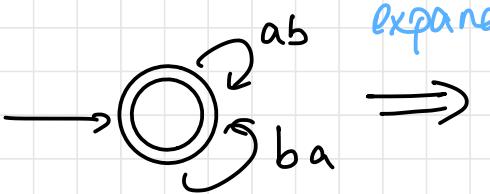
Question 45

$(ab)^*$

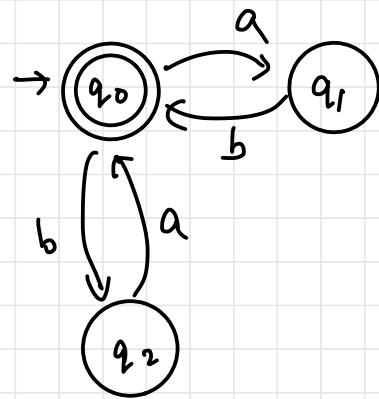


Question 46

$(ab + ba)^*$

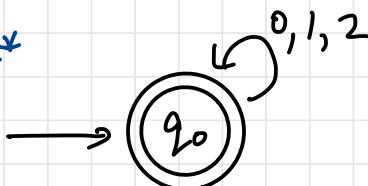


expand



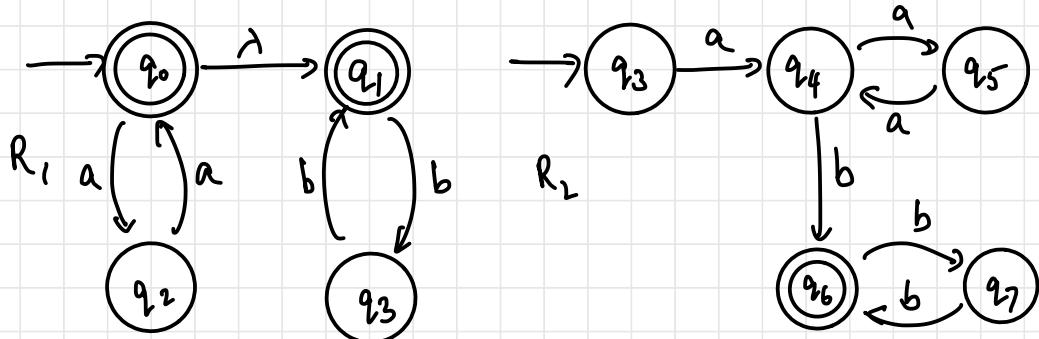
Question 47

$0^* + 1^* + 2^*$

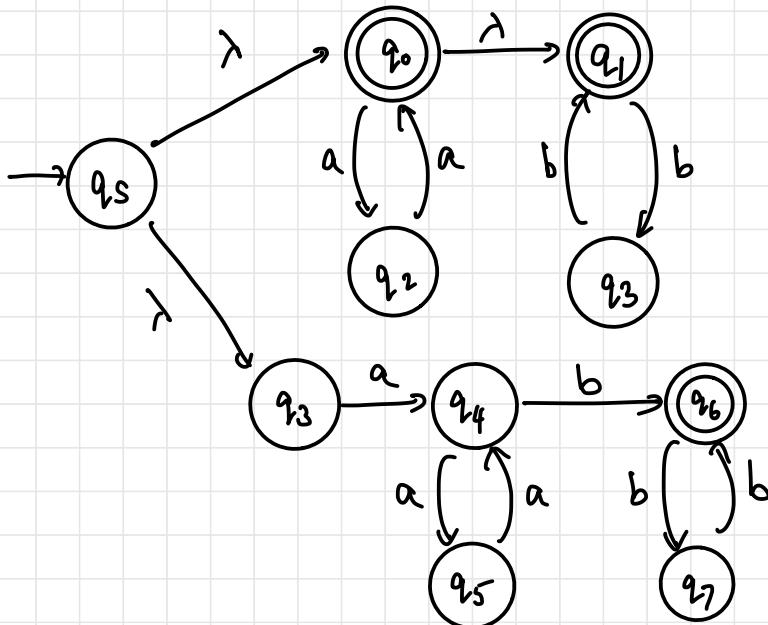


Question 48

$$R_1 (caa)^* (bbb)^* + a (caa)^* b (bbb)^* R_2$$

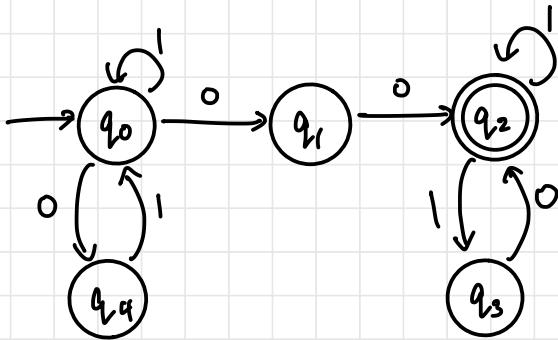


$$R_1 + R_2$$



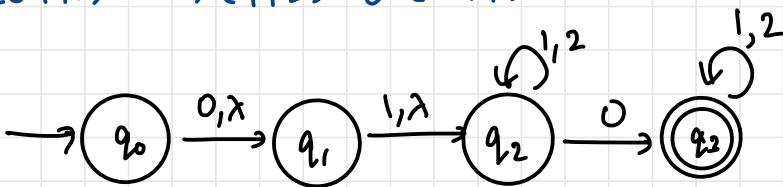
Question 49

$$(1+01)^* \text{ } 00 \text{ } (1+10)^*$$



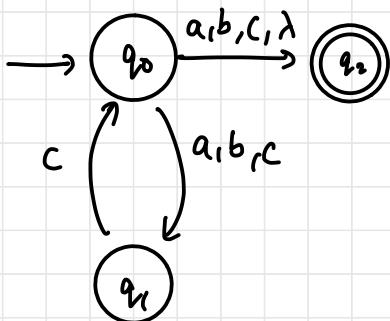
Question 50

$$(0+\lambda)(1+\lambda)(1+2)^* \text{ } 0 \text{ } (2+1)^*$$



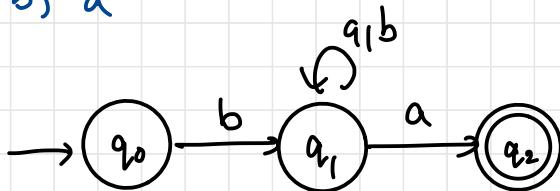
Question 51

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



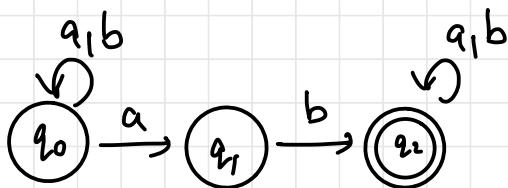
Question 52

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



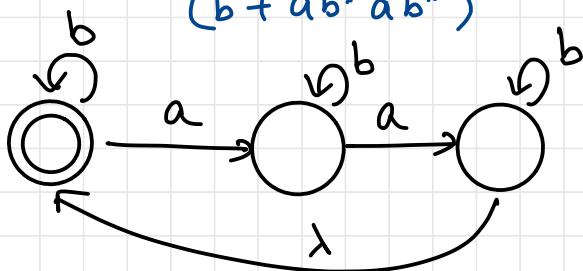
Question 53

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



Question 54

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

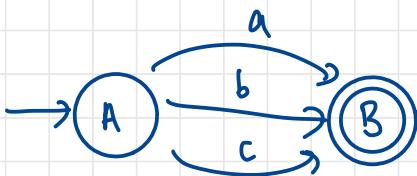


FA to RE

state reduction/ state elimination method

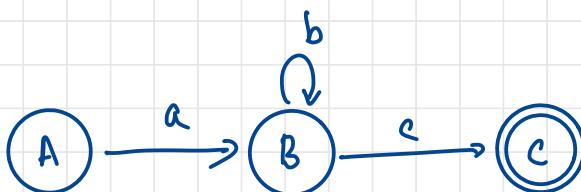
- Do not eliminate start & final state
- Start state having incoming edge, introduce new start state, make transition from NSS
- Final state having outgoing edge, introduce new final state
- Multiple final states, introduce new single FS.

Question 55



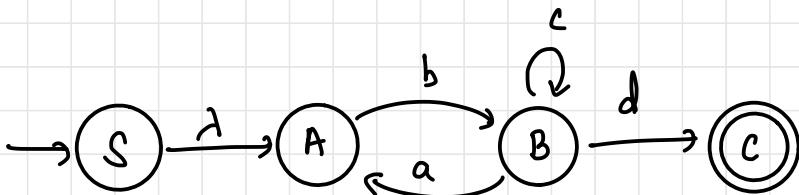
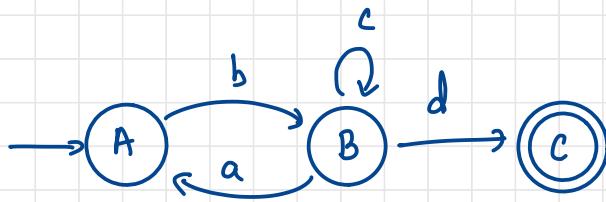
$(a+b+c)$

Question 56



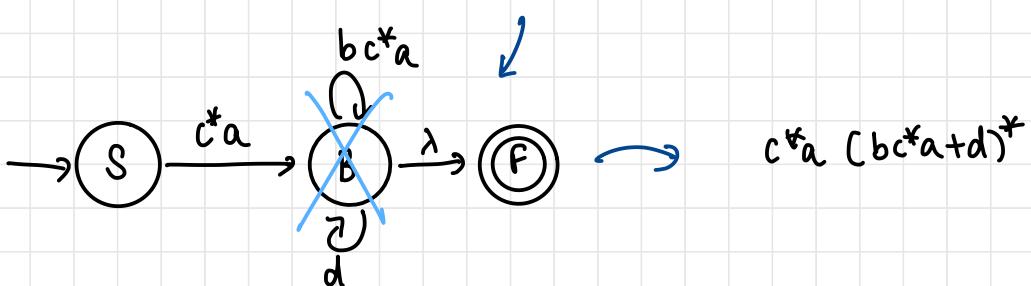
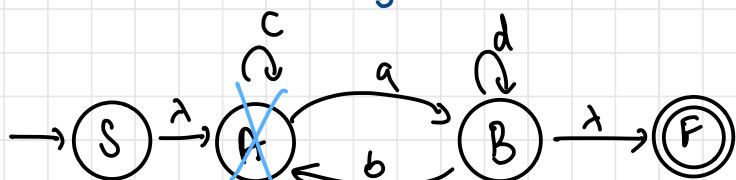
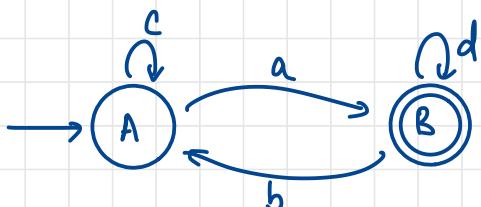
ab^*c

Question 57

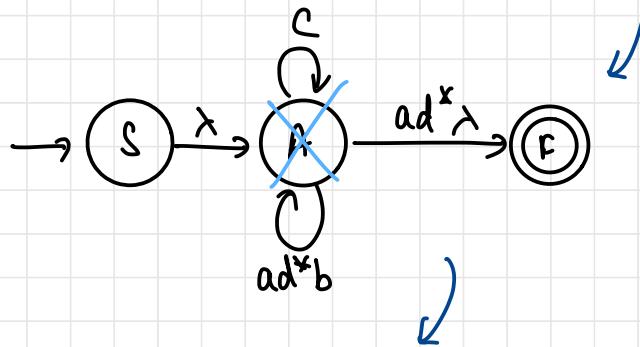
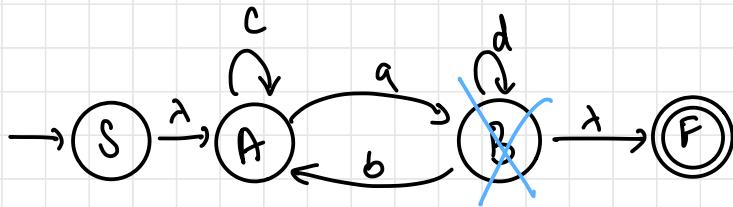


$$bc^* (ab + c)^* d \quad \text{or} \quad (b c^* a)^* b c^* d$$

Question 58

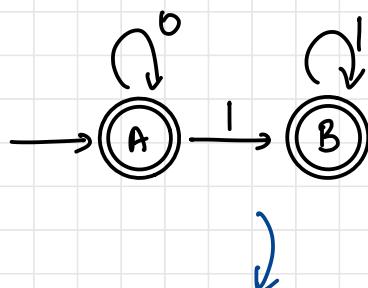
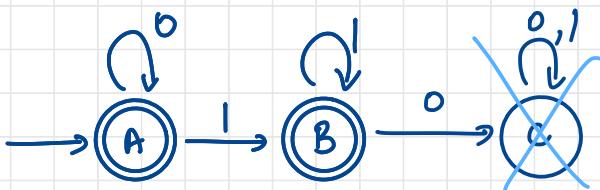


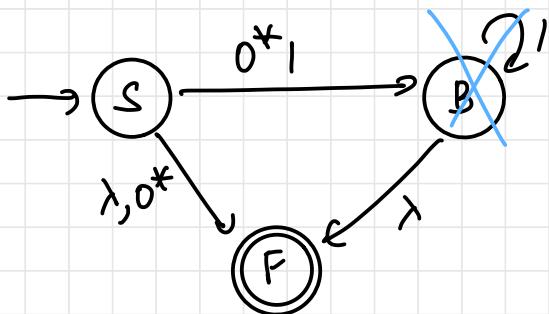
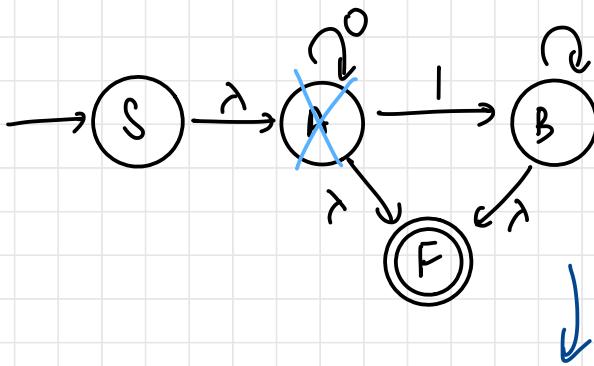
(OR)



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

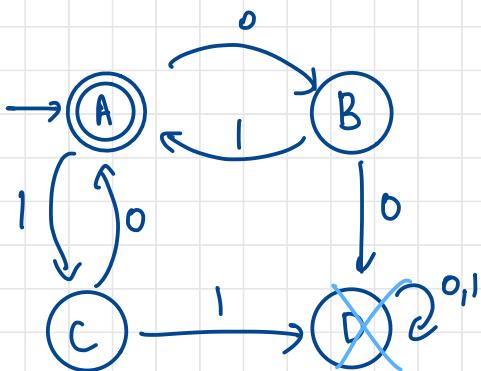
Question 59

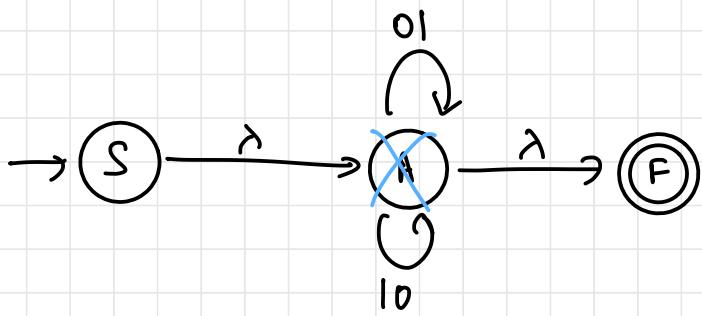
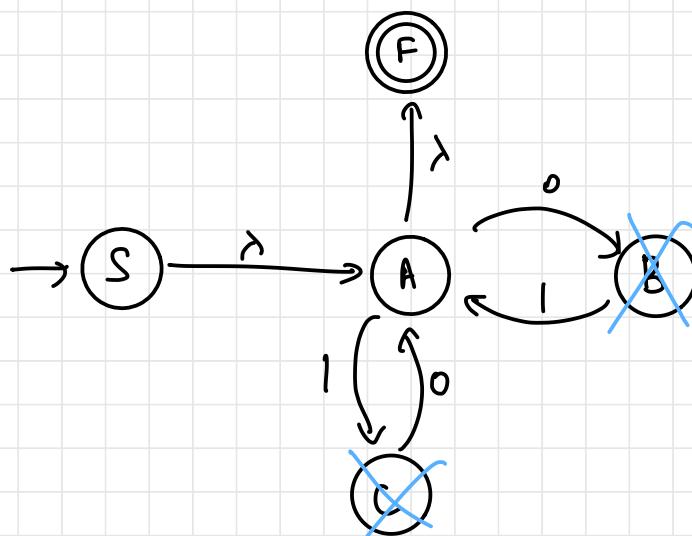




$$(0^* 1 1^* + 0^* + \lambda) = 0^* + 0^* 1 1^*$$

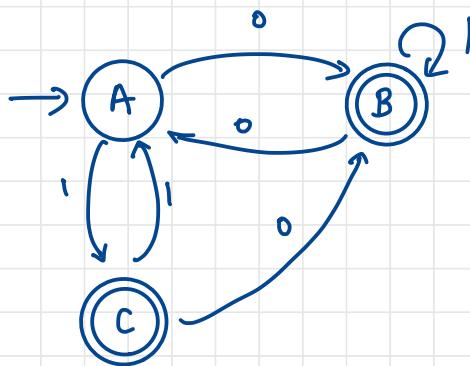
Question 60



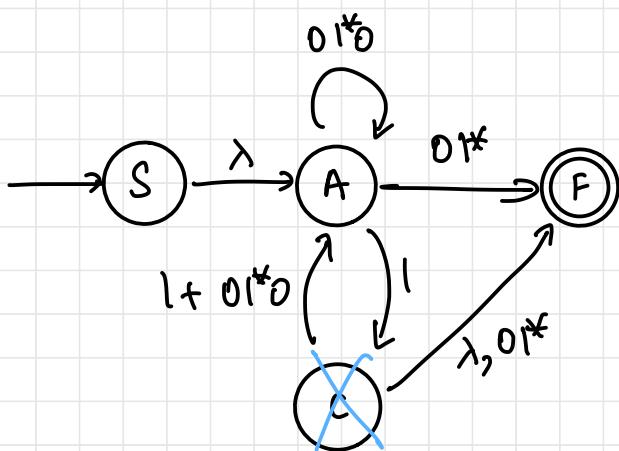
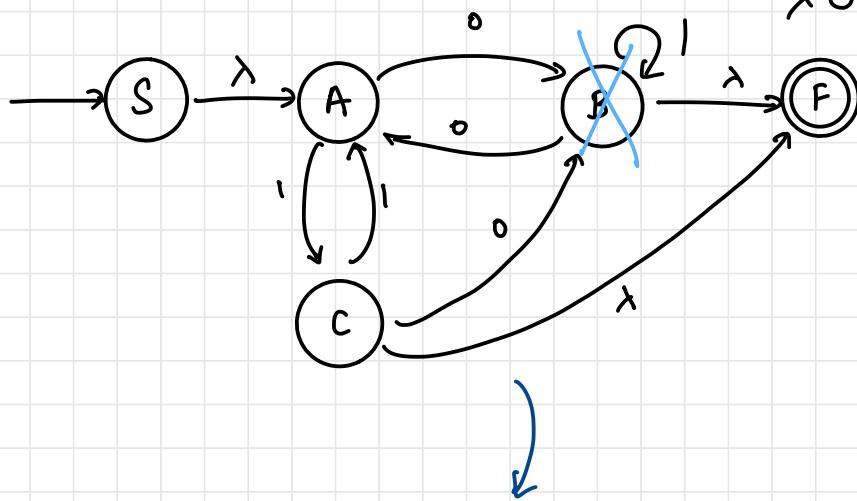


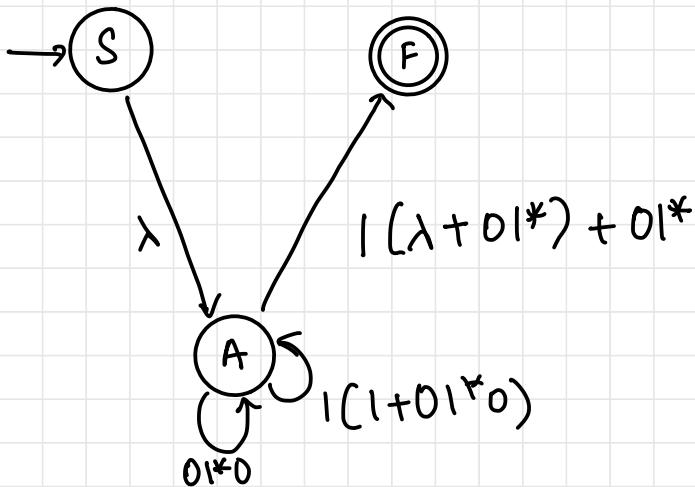
$$(01 + 10)^*$$

Question 61



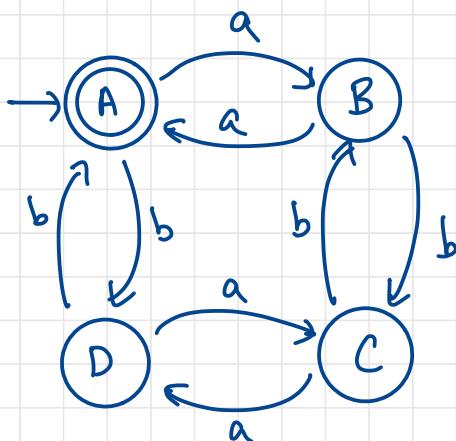
$\lambda 0^* \lambda$
 $\times 0^*(00)^* 1^* \lambda$

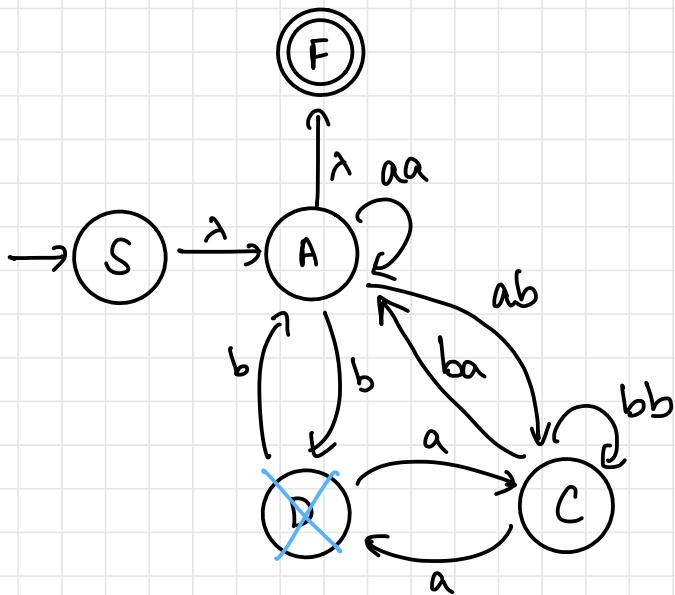
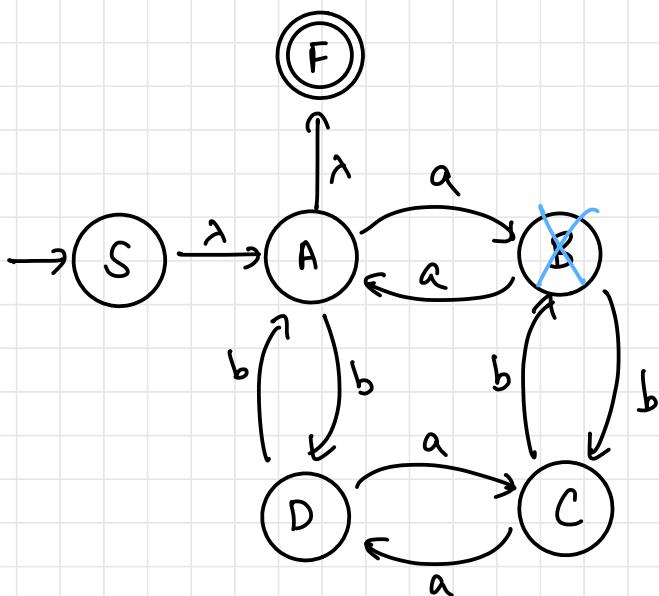


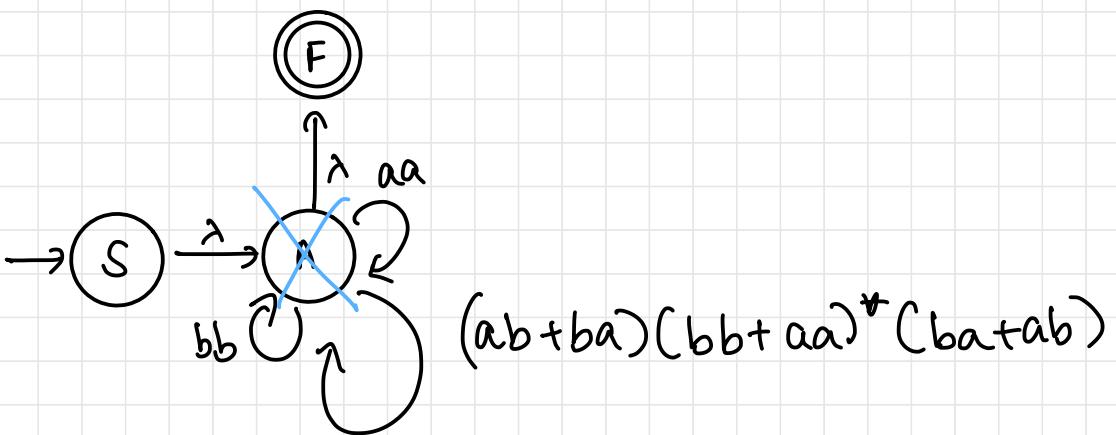
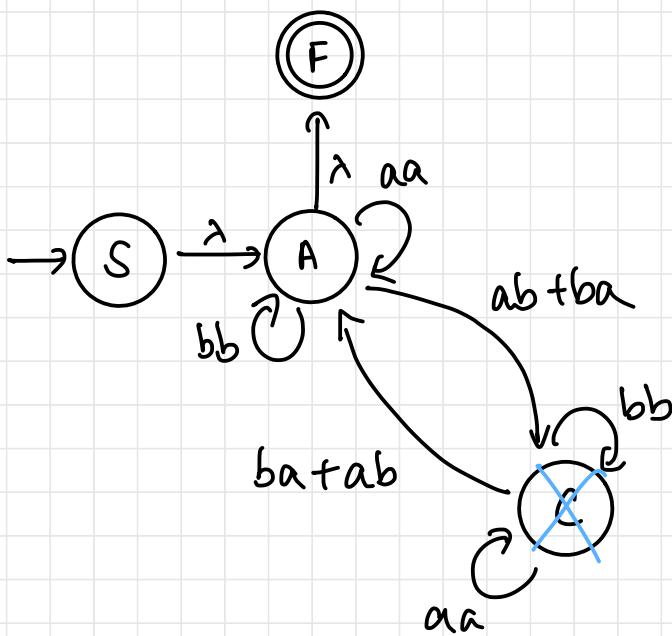


$$(01^*0 + I(1 + 01^*0))^* \quad (I(\lambda + 01^*) + 01^*)$$

Question 62







$$(aa+bb+ [(ab+ba)(bb+aa)^*(ab+ba)])^*$$

Equivalence of RE

Examples

Question 63

$$0^* + 0^* 1 1^* \equiv 0^* (\lambda + 1 1^*)$$

$$0^* (\lambda + 1 1^*) \equiv 0^* (\lambda + 1^*)$$

$$0^* (\lambda + 1^+) \equiv 0^* (\lambda + 1^+)$$

$$0^* 1^+ \equiv 0^* 1^*$$

Question 64

$(1 + 00^* 1) + (1 + 00^* 1)(0 + 10^* 1)^* (0 + 10^* 1)$ is
equal to $0^* 1 (0 + 10^* 1)^*$

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

$$(1 + 00^* 1) (\lambda + (0 + 10^* 1)^* (0 + 10^* 1)) \equiv 0^* 1 (0 + 10^* 1)^*$$

$$(1 + 00^* 1) (\lambda + (0 + 10^* 1)^+) \equiv 0^* 1 (0 + 10^* 1)^*$$

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

$$(\lambda + 00^*) 1 (0 + 10^* 1)^* \equiv 0^* 1 (0 + 10^* 1)^*$$

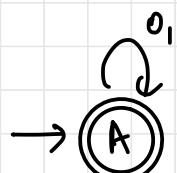
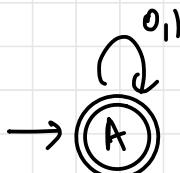
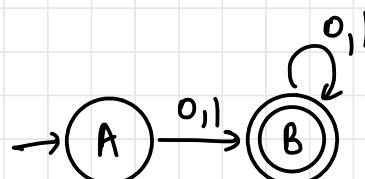
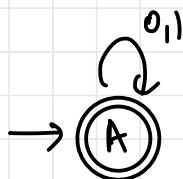
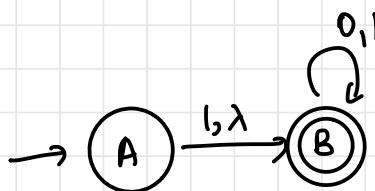
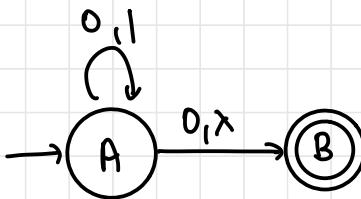
$$0^* 1 (0 + 10^* 1)^* \equiv 0^* 1 (0 + 10^* 1)^*$$

Formal Method

- Convert REs to NFAs, then to DFAs, then compare equality using table filling algorithm

Question 65

$$(0+1)^* (0+\lambda) \quad \text{and} \quad (1+\lambda) (1+0)^*$$



Regular Expressions in Practice

- Practical / real life strings
- pan card no, aadhaar card, date -2020, email
- packages in Python (re), Java etc.
- practical symbols / special characters meta characters

Characters

i) dot (.) - period operator

- matches a single character, no matter what that char is, except newline
- eg: $a.$ ← any char follows a

2) star (*)

- preceding character 0 or more times
- eg: a^* { λ , a, aa, aaa...}

3) Plus (+)

- preceding char one or more times
- eg: a^+ {a, aa, aaa...}

4) Question mark (?)

- preceding character upto one time
- eg: $x y^? z$ {xyz, xz}

5) Square brackets [] - character class

- range specified in []
- matches a single character in []
- eg: [x y z] x or y or z
[0-9] 0 or 1 or ... or 9
[a-zA-Z 0-9]

6) Cap/Carrot Symbol (^)

- (i) anchor for the start (starts with)

- eg: ^abc , string starts with abc

- (ii) negation

- eg: [^abc] except a or b or c, any symbol (could be xyef...)

7) Dollar (\$)

- anchor for the end (ends with)

- eg: efg\$, ends with efg

Note: ^\$: empty string

8) Curly braces {m,n}

- repetition from m to n times

- eg: a{2,4} a 2, 3 or 4 times

• eg: $a\{0,3\} \rightarrow a^*$

$a\{1,3\} \rightarrow a^+$

$a\{2,3\} \rightarrow aa$ or aaa

$a\{2\} \rightarrow aa$

Pan Card Number

- 10 characters
- first 5 chars, uppercase from A to Z
- next 4 chars, any digit from 0 to 9
- last character, any uppercase letter

Regular Expression

can be $\backslash d$



1) $^{\wedge}[A-Z]\{5\}[0-9]\{4\}[A-Z]\$$

2) $^{\wedge}[A-Z]\{5\}\backslash d\{4\}[A-Z]\$$

Indian Mobile Number

- 10 digit number
- starts with 9, 8, 7, 6
- next 9: from 0 to 9
- optional 0 or +91

or

$[0|+91]?[6-9][0-9]\{9\}$

Aadhaar Number

- 12 digit number
- Space after every four digits
- first digit is not 0 or 1

[2-9][0-9]{3}\s[0-9]{4}\s[0-9]{4}

Date in the year 2020

31 days — Jan, Mar, May, July, Aug, Oct, Dec
30 days — Apr, Jun, Sep, Nov
29 days — Feb²

dd-mm-yyyy or dd/mm/yyyy or dd.mm.yyyy

[((0[1-9]| [10-31]) [-.] (0[13578]| 1[02])) |
((0[1-9]| [10-30]) [-.] (0[469]|))) |
((0[1-9]| [10-29]) [-.] (02))] - 2020 \$

Email Address

personal-info@domain-name (only .com & .org)

[a-zA-Z1-9-.]+@[a-zA-Z1-9-].[com|org]

Grammar

$$G = \{V, T, P, S\}$$

V = set of variables (non-terminals) \rightarrow capital (states)

T = set of terminals (symbols in the alphabet, end string)

P = production rules (rules of grammar)

P ($\alpha \rightarrow \beta$)
 ↕
 non-terminal String over $(VUT)^*$

S = start symbol $\rightarrow S$

i) Linear grammar

$$\text{LHS } \alpha \xrightarrow{\quad} \beta \text{ RHS}$$

at most one non-terminal at the RHS of the production

$$S \rightarrow aSb / bSa / Sa / aS / \lambda / aa$$

2) Non-linear grammar

no restrictions

$$S \rightarrow aSS / aSSb / SSa / Sa / AB$$

Regular Grammar \subset Linear Grammar

(i) Left linear Grammar

if nonterminal is leftmost symbol

$$S \rightarrow S\alpha | \lambda$$

(ii) Right Linear Grammar

if nonterminal is rightmost symbol

$$S \rightarrow \alpha S | \lambda$$

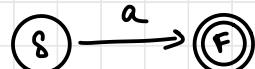
Automata to Grammar - Right Linear

Question 66

$L = \{a^f\}$, write RG for $RE = a^f$

RG:

$$S \rightarrow a$$



Question 67

$L = \{aaaaa\}$, write RG $RE = a^5$

RG: $S \rightarrow aaaa$

Question 68

$$L = \{\lambda, a, aa, aaa, \dots\}$$

$$RE = a^*$$

RG:

$$S \rightarrow aS | \lambda$$

eg: aaaa

Derivation

$$\begin{aligned} S &\Rightarrow aS \\ &\Rightarrow a aS \\ &\Rightarrow a a aS \\ &\Rightarrow a a a aS \\ &\Rightarrow a a a a \lambda \end{aligned}$$

Question 69

$$L = \{a, aa, \dots\} \quad RE = a^t$$

$$RG: \quad S \rightarrow aS | a = a^t$$

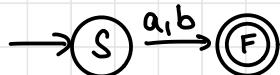
eg: aaaa

$$\begin{aligned} S &\Rightarrow aS \\ &\Rightarrow a aS \\ &\Rightarrow a a aS \\ &\Rightarrow a a a a \end{aligned}$$

Question 70

$(a+b)$

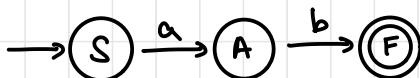
$s \rightarrow a|b$



Question 71

ab

$s \rightarrow ab$



$s \rightarrow aA$
 $A \rightarrow b$

Question 72

ab^*

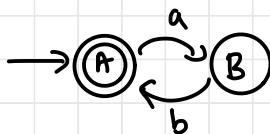
$s \rightarrow aA | \lambda$
 $A \rightarrow bA | b$



Question 73

$(ab)^*$

$s \rightarrow aA | \lambda$
 $A \rightarrow bS$



Question 74

$(a+b)^*$



$S \rightarrow aS \mid bS \mid \lambda$

Question 75

$(a+b)^+$

$S \rightarrow aS \mid bS \mid a \mid b$

Question 76

$ab + ba$

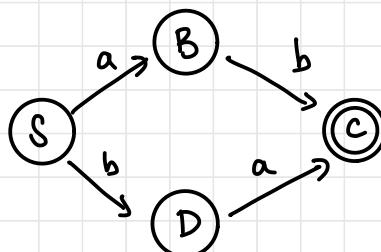
$S \rightarrow ab \mid ba$

OR

$S \rightarrow aB \mid bD$

$B \rightarrow b$

$D \rightarrow a$



Question 77

$(ab+ba)^*$

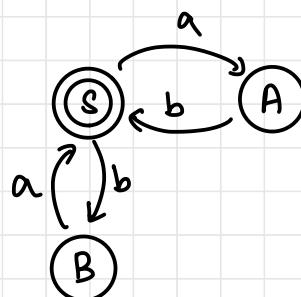
$S \rightarrow abS \mid baS \mid \lambda$

OR

$S \rightarrow aA \mid bB \mid \lambda$

$A \rightarrow bS$

$B \rightarrow aS$

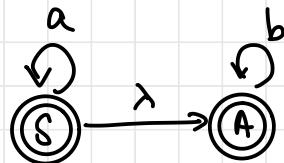


Question 78

$a^* b^*$

$$S \rightarrow aS \mid A$$

$$A \rightarrow bA \mid \lambda$$



e.g: aabb

$$\begin{aligned} S &\Rightarrow aS \\ &\Rightarrow aaS \\ &\Rightarrow aaaS \\ &\Rightarrow aabbA \\ &\Rightarrow aabbba \\ &\Rightarrow aabb\lambda \end{aligned}$$

Question 79

$(ab+ba)^+$

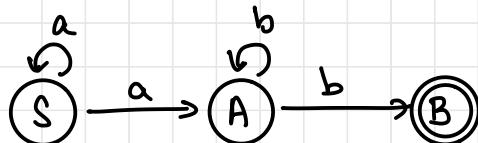
$$S \rightarrow abS \mid baS \mid ab \mid ba$$

Question 80

$$\mathcal{L} = \{a^n b^m \mid n, m \geq 1\} ; \text{RE} = a^* b^*$$

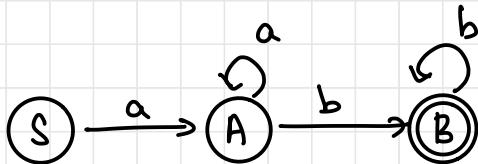
$$S \rightarrow aS \mid aA$$

$$A \rightarrow bA \mid b$$



OR

$$\begin{aligned} S &\rightarrow aA \\ A &\rightarrow aA \mid bB \\ B &\rightarrow bB \mid \lambda \end{aligned}$$



Question 81

$L = \{\text{at least one } a\}$

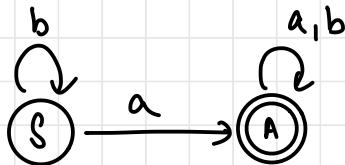
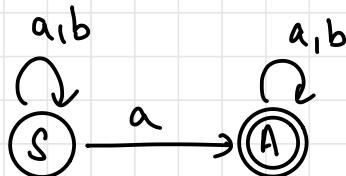
$$(a+b)^* a (a+b)^*$$

$$\begin{aligned} S &\rightarrow aS \mid bS \mid aA \\ A &\rightarrow aA \mid bA \mid \lambda \end{aligned}$$

OR

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

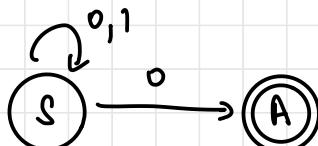
$$\begin{aligned} S &\rightarrow bS \mid aA \\ A &\rightarrow aA \mid bA \mid \lambda \end{aligned}$$



Question 82

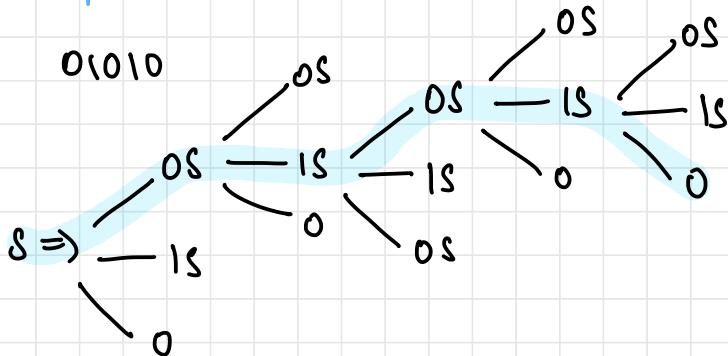
$$\text{Binary even no.} = (0+1)^* 0$$

$$S \rightarrow 0S \mid 1S \mid 0$$

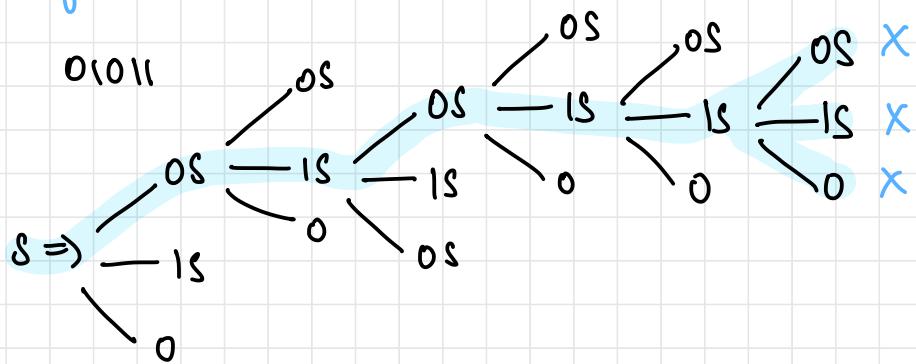


Derivation using tree

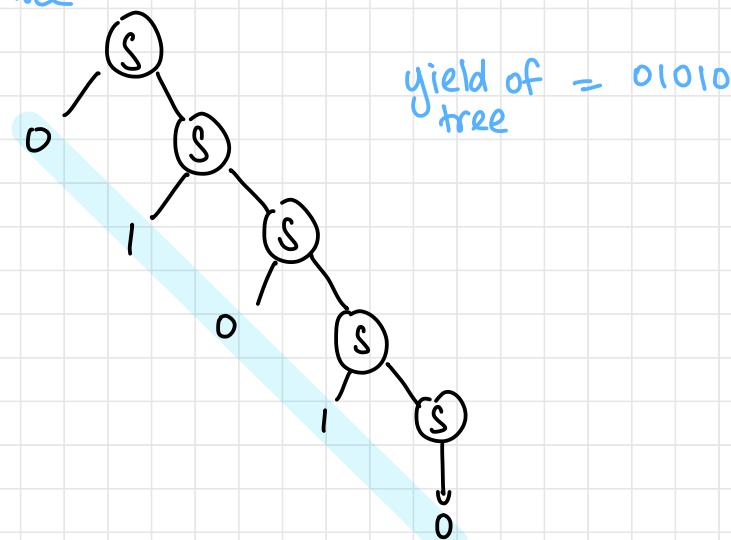
(i) Accepted



(ii) Rejected



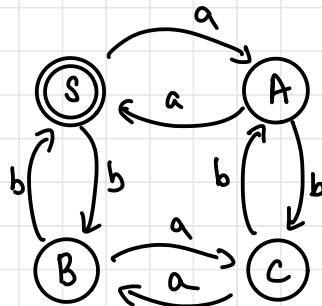
Parse tree



Question 83

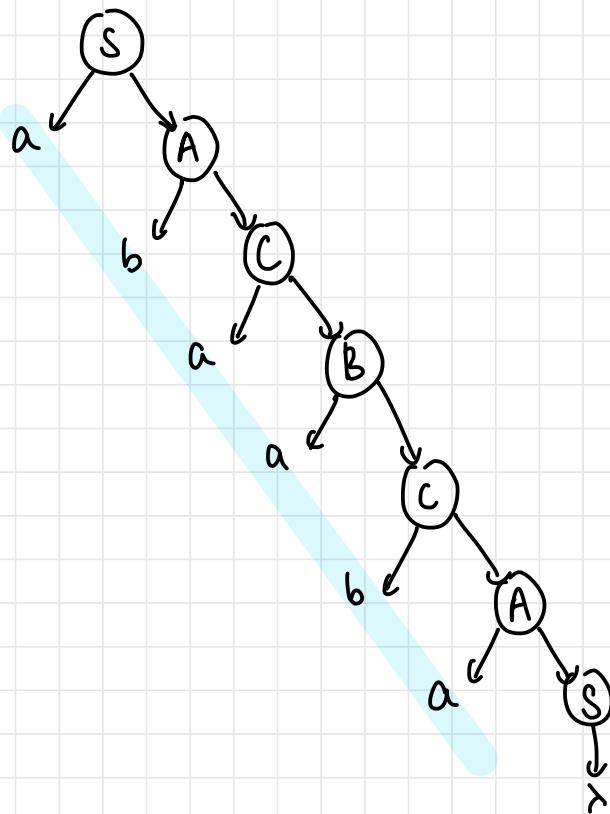
$$L = \{ n_a(w) \bmod 2 = 0 \text{ and } n_b(w) \bmod 2 = 0 \}$$

$S \rightarrow aA \mid bB \mid \lambda$
 $A \rightarrow aS \mid bc$
 $B \rightarrow bS \mid ac$
 $C \rightarrow aB \mid bA$



Parse tree

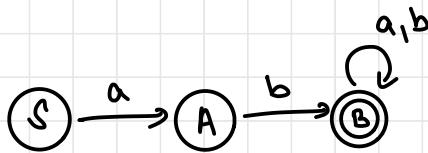
a b a a b a



Question 84

$$\mathcal{L} = \{abw, w \in \{a,b\}^*\}$$

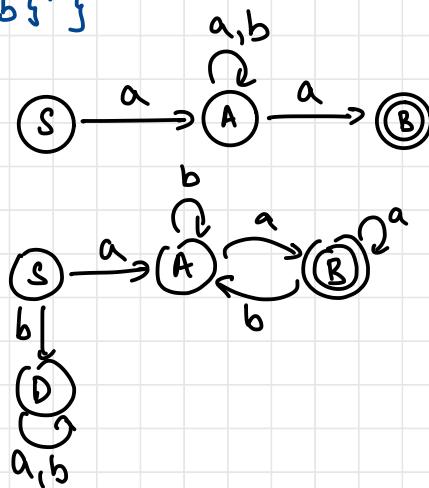
$$\begin{aligned} S &\rightarrow aA \\ A &\rightarrow bB \\ B &\rightarrow abB | bB | \lambda \end{aligned}$$



Question 85

$$\mathcal{L} = \{awa, w \in \{a,b\}^*\}$$

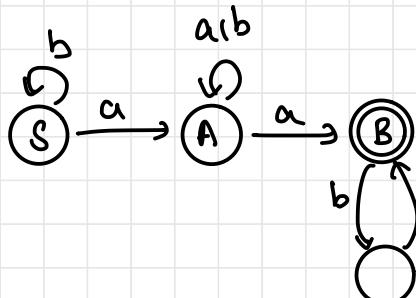
$$\begin{aligned} S &\rightarrow aA \\ A &\rightarrow aA | bA | a \end{aligned}$$



Grammar to Automata

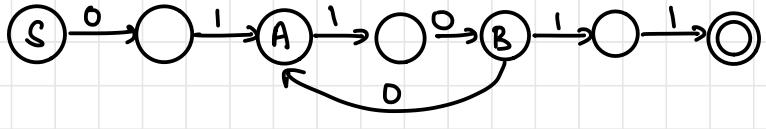
Question 86

$$\begin{aligned} S &\rightarrow bS | aA \\ A &\rightarrow aA | bA | aB \\ B &\rightarrow bbB \\ B &\rightarrow \lambda \end{aligned}$$



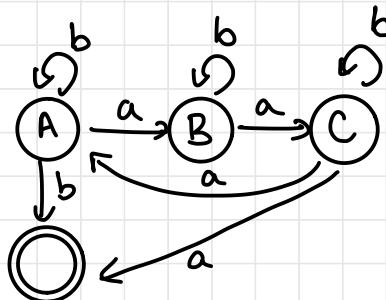
Question 87

$$\begin{aligned} S &\rightarrow 0 \mid A \\ A &\rightarrow 1 \mid 0B \\ B &\rightarrow 0A \mid 11 \end{aligned}$$



Question 88

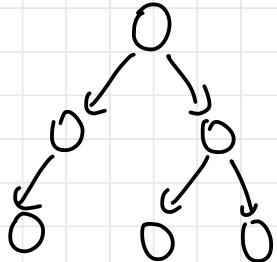
$$\begin{aligned} A &\rightarrow ab \mid bA \mid b \\ B &\rightarrow ac \mid bB \\ C &\rightarrow aA \mid bc \mid a \end{aligned}$$



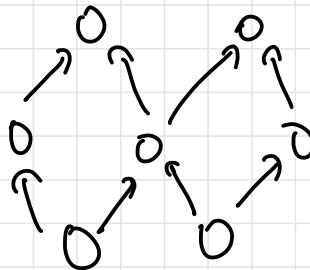
Parsing

Parsing is a process of determining whether a string $w \in L(G)$

1) Top-down parsing



2) Bottom-up parsing



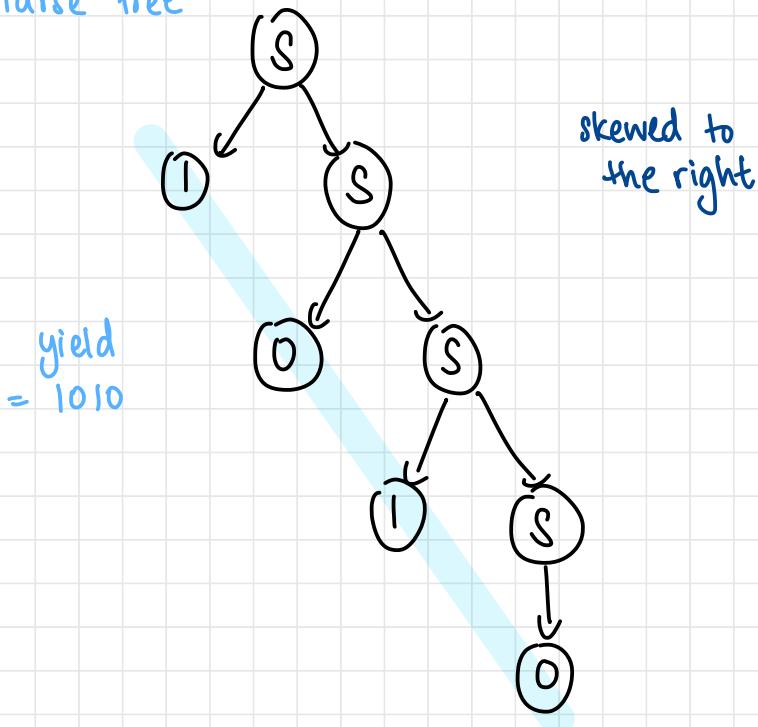
Parsing from Right Linear Grammar \rightarrow Rightmost derivation

$$S \rightarrow OS \mid 1S \mid 0$$

1010

$$\begin{aligned} S &\xrightarrow{RM} 1S \\ &\xrightarrow{} 1OS \quad (S \rightarrow OS) \\ &\xrightarrow{} 10IS \quad (S \rightarrow 1S) \\ &\xrightarrow{} 1010 \quad (S \rightarrow 0) \end{aligned}$$

Parse tree



Question 89

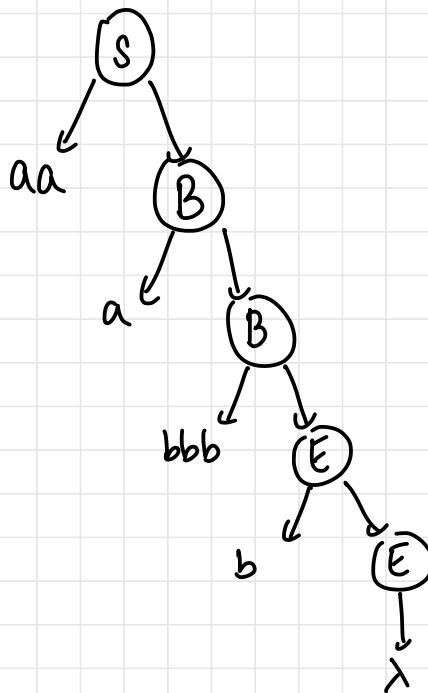
Write derivation & parse tree

$$\begin{aligned} S &\rightarrow aaB \\ B &\rightarrow aB \mid bbbE \\ E &\rightarrow bE \mid \lambda \end{aligned}$$

string = aaabbhhh

$$\begin{aligned} S &\xrightarrow{\text{RM}} aaB \\ &\Rightarrow aa aB \\ &\Rightarrow aaa bbbE \\ &\Rightarrow aaa bbbbE \\ &\Rightarrow aaa bbbb \end{aligned}$$

Parse tree



Left Linear Grammar

Question 90

$$RE = a$$

$$\begin{array}{c} RLG \\ S \rightarrow a \end{array}$$

$$\begin{array}{c} LLG \\ S \rightarrow a \end{array}$$

Question 91

$$a+b$$

$$RRG$$

$$S \rightarrow a|b$$

$$LLG$$

$$S \rightarrow a|b$$

Question 92

$$ab$$

$$RLG$$

$$\begin{array}{l} S \rightarrow aA \\ A \rightarrow b \end{array}$$

$$LLG$$

$$\begin{array}{l} \epsilon \rightarrow Ab \\ A \rightarrow a \end{array}$$

Question 93

$$a^*$$

$$RLG$$

$$S \rightarrow aS|\lambda$$

$$LLG$$

$$S \rightarrow Sa|\lambda$$

Question 94

$(a+b)^*$

RLG

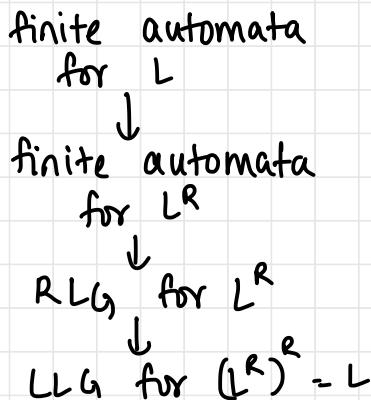
$$S \rightarrow aS \mid bS \mid \lambda$$

LLG

$$S \rightarrow Sa \mid Sb \mid \lambda$$

Convert RLG to LLG

- if converted to RLG by switching positions, reversal of language
- steps



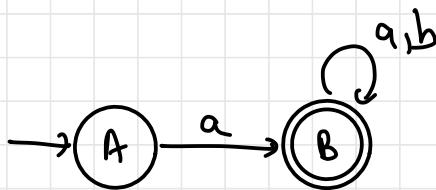
Question 95

$$A \rightarrow aB$$

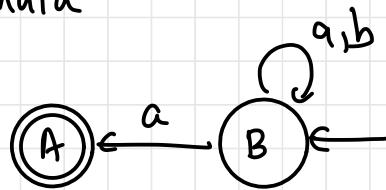
$$B \rightarrow aB \mid bB \mid \lambda$$

$L = \{ \text{starts with } a \}$

automata



reversed automata



write RLG

$$\begin{aligned} B &\rightarrow aB \mid bB \mid aA \\ A &\rightarrow \lambda \end{aligned}$$

convert to LRG

$$\begin{aligned} B &\rightarrow Ba \mid Bb \mid Aa \\ A &\rightarrow \lambda \end{aligned}$$

Convert LLG to finite automata

- 1) Swap (make RLG of L^k)
- 2) Make automata
- 3) Reverse

Question 96

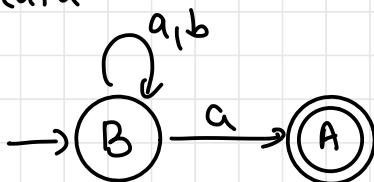
$$\begin{array}{l} B \rightarrow Ba \mid Bb \mid Aa \\ A \rightarrow \lambda \end{array}$$

convert to automata

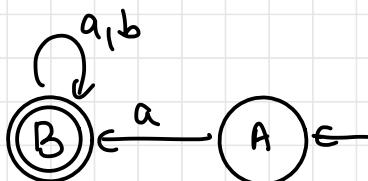
RLG of L^k

$$\begin{array}{l} B \rightarrow aB \mid bB \mid aA \\ A \rightarrow \lambda \end{array}$$

automata



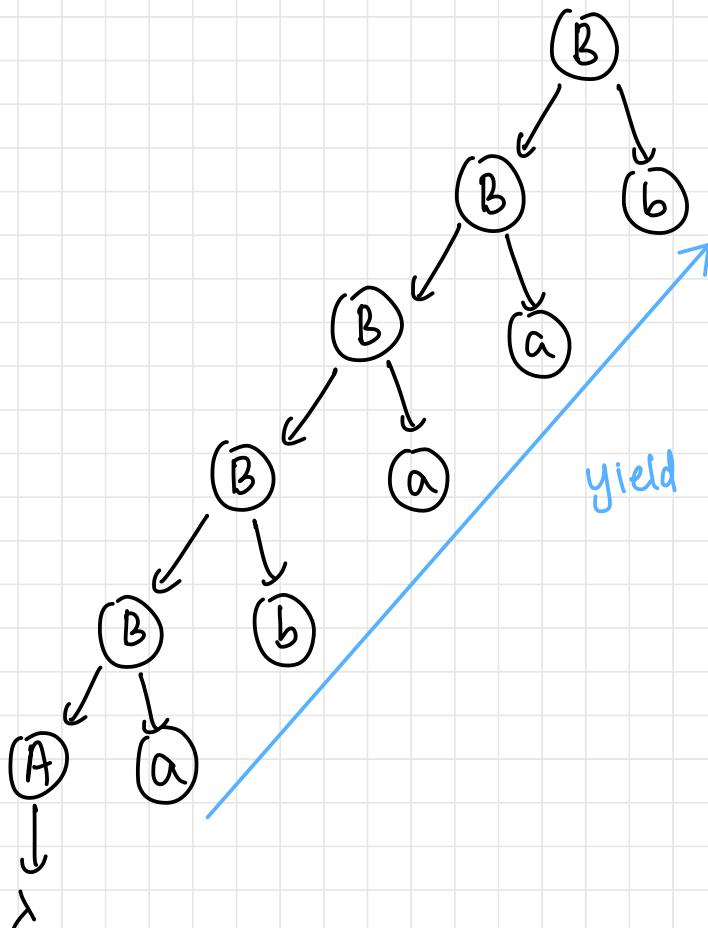
reversed



Parse tree for LLG

$$\begin{array}{l} B \rightarrow Ba \mid Bb \mid Aa \\ A \rightarrow \lambda \end{array}$$

abaab



Properties of Regular Languages

1. Concatenation

- if L_1, L_2 are RLs, $L_1 \cdot L_2$ is RL

2. Union

- if L_1, L_2 are RLs, $L_1 \cup L_2$ is RL

3. Closure

- if L is RL, L^* is RL

4. Reversal

- if L is RL, L^R is RL

5. Complement

- if L is RL, \bar{L} is RL
- change FS \rightarrow NFS & NFS \rightarrow FS
- only on DFA

6. Intersection

- $L_1 \cap L_2$ RLs, $L_1 \cap L_2$ RL
- Cartesian product / cross product
- Using De Morgan's Law
- only if accepted by both (both final states)

7. Set difference

- strings in L_1 and not in L_2
- cross products

Decidable Properties of Regular Languages

1) Testing emptiness

- is given RL empty
- no path from q_s to $F = \{ \dots \}$
- λ is a valid path; not empty L
- if grammar is non-producing

2) Membership

- given a string w, check if it belongs to the language or not

3) Finite / infinite

- loops: infinite

4) $L = \Sigma^*$

- accepts everything
- find complement of language and check if it accepts nothing

5) $L_1 = L_2$

- construct DFAs and minimise

Pumping Lemma

- Check if language is regular or not
- When impossible to convert language to DFA/NFA: not regular
- Languages are finite or infinite ($*$ → infinite)
- Finite languages are always regular
- $a^* b^*$ is infinite, regular

Question 97

Show that $L = \{a^n b^n\}$ is irregular

Pigeon hole principle

$n \rightarrow$ pigeons

$m \rightarrow$ pigeon holes

divide string into x, y & z

if no. of pigeons > no. of holes,
one hole will have > 1 pigeons

$$|zy| \leq n \quad |y| \geq 1 \quad z$$

$\begin{matrix} a^n & b^n \\ \downarrow & \downarrow \\ xy & z \end{matrix}$

aaa a bbb

if $y = 2$, lang not regular

Question 98

w | equal no. of 0's and 1s

bad choice: CODⁿ

good choice: 0ⁱ 1ⁱ → irregular

0011
xy z

	x	y	z
i = 0	0	0	11

011 X

i = 1
 0011 → bad eg

i = 2
 00011 X

Question 99

Palindrome

aⁿ b^m b^m aⁿ w w^R | w ∈ {a, b}^{*}

Assume n states in machine

Assume ww^R is regular

i) $a^n b b a^n$

$$|xyz| \leq n$$

$\underbrace{a^n}_{xy} \mid \underbrace{bb a^n}_z \rightarrow xyz$

$$n=3$$

$\underbrace{aaa}_{x} \mid \underbrace{bbaaa}_{y} z$
 $xyz \quad |y| \geq 1$

$i=1 \rightarrow$ bad choice

$$i=2$$

$aa aa bb a a a \notin L$

Revision - Regular Expressions

Question 1

Write RE for $L = \{ w \mid w \neq 000, \{0,1\}^*\}$

$|w| = 3, w \neq 000$

$|w| < 3$ or $|w| > 3$

$$\lambda + 0 + 1 + (0+1)^2 +$$

$$001 + 010 + 011 + 100 + 101 + 110 + 111 +$$

$$(0+1)^4 (0+1)^*$$

Question 2

All strings w such that in every prefix of w , the no. of 0's and 1's differ by at most 1.

$$(01+10)^* (\lambda + 0 + 1)$$

Question 3

At least 2 0's and at least one 1

$$\begin{aligned} & 0 (0+1)^* 0 (0+1)^* 1 (0+1)^* + \\ & 1 (0+1)^* 0 (0+1)^* 0 (0+1)^* + \\ & 0 (0+1)^* 1 (0+1)^* 0 (0+1)^* \end{aligned}$$