8) Subset of NRL need not be NRL

9) Superset of RL need not be R
" " NR " " " NRL

10) Finite Union of Regular Language is a RL

11) Finite Intersect of RL is a RL

Regular Expressions (RE)

(Used in some programming lang. to find substring)

→ One way to represent a RL.

→ Operators: $*$, $\cdot$, $+$

$*$ ↓ Kleen closure

$\cdot$ ↓ Concatenation

$+$ → OR/ Union

$+$ in superscript

$r = a^{+} = \{a, aa, aaa, \ldots\}$

$r = a^{+} = a^{*} - \{\epsilon\}$

Ex:

1) $r = a^{*} = \{\epsilon, a, aa, aaa, aaaa, \ldots\}$

2) $r = a+b = \{a, b\}$

3) $r = ab = \{ab\}$

4) $a^{*} + ba = \{\epsilon, a, aa, \ldots\} \cup \{ba\}$

5) $(ab)^{*} + b^{*}a^{*} = \{\epsilon, ab, abab, ababab, \ldots\}$
$\cup \{\{\epsilon, b, bb, bbb \ldots\} \cdot \{\epsilon, a, aa, aaa, \ldots\}\}$

Operators can be applied/used on symbols, sets, RE, alphabets

* Order of Precedence.

1st : *

2nd : .

3rd : +

Ex: $(a+ba)^* = \{\epsilon, a, ba, aa, aba, baa, baba, \ldots\}$

$\downarrow$

$\epsilon \cup (a+ba)^1 \cup (a+ba)^2 \cup (a+ba)^3 \cup \ldots$

* Types of RE

- Restricted/standard RE $(*, \cdot, +)$
- Semi-Restricted RE $(*, \cdot, +, \cap)$
- Unrestricted RE $(*, \cdot, +, \cap, \sim)$ — $\sim$: compliment

* Observation

1) Every finite language is regular & we can express it using a regular expression

2) $r^* = \{\epsilon, r, rr, rrr, \ldots\}$

$r^+ = \{r, rr, rrr, \ldots\}$

~~Also~~ $L(r^*) = L(r)^*$

Ex: $r = ab$

$L((ab)^*) = \{\epsilon, ab, abab, \ldots\}$

$L(ab)^* = \{ab\}^* = \{\epsilon, ab, abab, \ldots\}$

Similarly, $L(r^+) = L(r)^+$

3) $r = \epsilon$; $r^* = \{\epsilon, \epsilon\epsilon, \epsilon\epsilon\epsilon, \ldots\} = \{\epsilon\}$

$r^+ = \{\epsilon, \epsilon\epsilon, \epsilon\epsilon\epsilon, \ldots\} = \{\epsilon\}$

$r = \phi = \{\}$; $r^* = \{\epsilon\}$

$r^+ = \{\} = \phi$

4) $r^+ \subset r^*$

$r^* = r^+ \cup \{\epsilon\}$

5) $(r^*)^* = r^*$ $r^3 = r.r.r$

$r^*.r = \{r, r^2, r^3, \ldots\}$

Exercise:

1) Which is correct

a) $r^* = r(*)$

b) $(r^*)^+ = r^+$

c) $(r^+)^* = r^*$

d) $(r^+)^* = r^+$

2) Which is/are correct

a) $r^* = r^+$

b) $r^* = \bigcup_{i \geq 0} r^i$

c) $r^+ = \bigcup_{i \geq 1} r^i$

d) $r^* = r^+$ iff $r = \epsilon$

3) Which is/are incorrect

True a) $r^* + r^+ = r^* \Rightarrow \{\epsilon, r, r^2, r^3, \ldots\} \cup \{r, r^2, r^3, \ldots\} = \{\epsilon, r, r^2, r^3, \ldots\}$

True b) $r^* . r^+ = r^+ \Rightarrow \{\epsilon, r, r^2, r^3 \ldots\} . \{r, r^2, r^3, \ldots\} = \{\epsilon r, \epsilon r^2, \epsilon r^3, rr, \ldots\}$

c) $r^* . r^* = r^+$ $r^* . r^* \neq r^+$

True d) $r^+ (r^*)^+ = r^+$

4) Choose the correct statement.

RL can have multiple RE
but
RE can have a unique RL

a) r^*, r^+ always represent ∞ lang. ⇒ not true for $r = \epsilon$, $r^* = r^+ = \{\epsilon\}$ = finite

b) r^*, r^+ " " finite lang.

c) r^*, r^+ are finite iff $r = \phi$ or $r = \epsilon$

d) none

5) Which of the following are identical?

a) $a^* = \{\epsilon, a, aa, aaa, \ldots\}$

b) $(aa)^* = \{\epsilon, aa, aaaa, \ldots\}$

c) $(aa^*)a = \{aa, aaa, aaaa, \ldots\}$

d) $(a+\epsilon)a^* = \{\epsilon, a, aa, aaa, \ldots\}$

6) Which is correct?

a) $(r_1^* + r_2^*) \neq (r_1 + r_2)^*$ ⇒ In RHS we can get $r_1 r_2$ but its not possible in LHS

b) $r_1^* r_2^* \neq (r_1 + r_2)^*$ ⇒ $r_2 r_1 \in$ RHS but $\notin$ LHS

c) $(r_1^* + r_2)^* = (r_1 + r_2)^*$

d) $(r_1 r_2)^* \neq (r_1 + r_2)^*$ ⇒ $r_2 r_1 \in$ RHS but $\notin$ LHS

7) Let $r_1 = 0^*1$ & $r_2 = (0^* 1^*)^*$. which is correct?

a) $L(r_1) = L(r_2)$

b) $L(r_1) \subset L(r_2)$

c) $L(r_1) \supset L(r_2)$

d) none

$L(r_1) = \{1, 01, 0001, 001, 00001, \ldots\}$

$L(r_2) = \{\epsilon, 0, 1, 01, \ldots\}$

8) $r = (a^*b)^*$ $\quad s = (a+b^*)^*$. Which is correct?

a) $L(r) = L(s)$

b) $L(r) \subset L(s)$ ✓

c) $L(r) \supset L(s)$

d) none

$L(r) = \{\epsilon, a, b, ab, aab, bab, \ldots$

$L(s) = \{\epsilon, a, b, ab, aab, bab, ba, \ldots\}$

9) Which doesn't contain the string '1010'.

a) $0^*(10)^*1$ ✓

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

c) $(110)^*0^*1$

d) $0^*(101)^*1$

10) Which doesn't contain/generate string containing substring '100'.

a) $0^*(10)^*$ ✓

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

c) $(01^*+0)^*(01)^*1^*0^*$

d) $(110^*+1)^*0$

11) Is $(r_1r_2)^*r_1 = r_1(r_2r_1)^*$? $\Leftarrow$ (shifting

yes

$*=0$: $\quad \epsilon r_1 = r_1\epsilon$

1: $\quad (r_1r_2)r_1 = r_1(r_2r_1)$

2: $\quad (r_1r_2r_1r_2)r_1 = r_1(r_2r_1r_2r_1)$

$\vdots$

Note : $(r_1 + r_2)^* = (r_1^* + r_2^*)^*$

$= (r_1^* + r_2)^*$

$= (r_1 + r_2^*)^*$

$= (r_1^* \cdot r_2^*)^*$

Powers of alphabet:

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

$\Sigma^1 = \Sigma = a+b$

$\Sigma^2 = \Sigma \cdot \Sigma = (a+b)(a+b) = (a+b)^2$

$\Sigma^3 = \Sigma \cdot \Sigma \cdot \Sigma = (a+b)^3$

⋮

$\Sigma^* = (a+b)^*$

* Constructing RE

1) a) RE that generates all strings including ϵ

$r = (a+b)^*$

b) RE that generates all strings excluding ϵ

$r = (a+b)^+$

1) Strings that start with '0'
$r = 0(0+1)^*$

2) starts with '10' $\Rightarrow r = 10(0+1)^*$

3) starts with (0110) $\Rightarrow r = 0110(0+1)^*$

4) a) strings that end in '0' $\Rightarrow (0+1)^* 0$
b) " " end in '10' $\Rightarrow (0+1)^* 10$
c) " " end in '1001' $\Rightarrow (0+1)^* 1001$

5) a) contains substring '10' $\Rightarrow (0+1)^* 10 (0+1)^*$
b) contains substring '010' $\Rightarrow (0+1)^* 010 (0+1)^*$

6) a) Strings that starts & ends in '0' $\Rightarrow 0(0+1)^* 0$
b) starts & ends in same symbol $\Rightarrow 0+0(0+1)^* 0 + 1(0+1$
c) starts & ends in different symbol $\Rightarrow 0(0+1)^* 1 + 1(0+$

7) a) 3^{rd} symbol from the left is '1' $\Rightarrow r = (0+1)^1 (0+1)^1 1 (0+$
b) 3^{rd} " " " right is '0' $\Rightarrow r = (0+1)^* 0 (0+1)^2$

8) a) exactly two 0's $\Rightarrow r = 1^* 0 1^* 0 1^*$
*b) ~~exactly~~ atmost two 0's $\Rightarrow r = 1^*(0+\epsilon)1^*(0+\epsilon)1^*$
c) atleast two 0's $\Rightarrow r = 1^* 0^+ 1^* 0^+ 1^*$ or $r = (0+1)^*$ 0 (
*d) # zeros $\equiv 0 \pmod 3$ $\Rightarrow$ # zeros = 0, 3, 6, 9, 12, ...
$$\Rightarrow r = \underbrace{1^*}_{0} + \underbrace{(1^* 0 1^* 0 1^* 0 1^*)^*}_{3,6,9,12....}$$

e) $\# 1's = 2 \pmod 3 \Rightarrow \# 1's = 2, 5, 8, 11, 13, \ldots$

$$\Rightarrow \underbrace{\underbrace{0^*10^*10^*}_{2} \cdot \underbrace{(0^*10^*10^*10^*)^*}_{0,3,6,9,\ldots}}_{2,5,8,11,\ldots}$$

9) a) $|w| = 2 \Rightarrow r = (0+1)^2$

b) $|w| \le 2 \Rightarrow r = (0+1+\epsilon)^2$

c) $|w| \ge 2 \Rightarrow r = (0+1)^2(0+1)^*$

d) $|w| \equiv 0 \pmod 2 \Rightarrow r = ((0+1)^2)^*$

$0, 2, 4, 6, 8, \ldots$

e) $|w| \equiv 1 \pmod 2 \Rightarrow r = (0+1)((0+1)^2)^*$

$1, 3, 5, 7, 9, \ldots$

f) $|w| \equiv 2 \pmod 3 \Rightarrow 2, 5, 8, 11, \ldots$

$$\Rightarrow r = \underbrace{(0+1)^2}_{2}\underbrace{((0+1)^3)^*}_{0,3,6,9,12,\ldots}$$

10) a) $L = \{0^n \mid n \ge 0\} \Rightarrow r = 0^*$

b) $L = \{0^n \mid n \ge 1\} \Rightarrow r = 0^+$

c) $L = \{1^n \mid n \ge 3\} \Rightarrow r = 1111^*$

d) $L = \{0^m 1^n \mid m, n \ge 0\} \Rightarrow r = 0^*1^*$

e) $L = \{0^m 1^n \mid m \ge 1; n \ge 1\} \Rightarrow r = 0^+1^+$

f) $L = \{0^m 1^n \mid m \ge 0; n \ge 1\} \Rightarrow r = 0^*1^+$

g) $L = \{0^m 1^n \mid m \ge 2; n \ge 3\} \Rightarrow r = 00^+111^+$

h) $L = \{0^m 1^n 2^p \mid m \ge 0; n \ge 1; p \ge 2\} \Rightarrow r = 0^*1^+22^+$

11) $L = \{0^m 1^n \mid m+n = \text{even}\}$

As even+even or odd+odd = even

$x \geq 0$

m : even $(2x)$ | m : odd $(2x+1)$

n : even $(2x)$ | n : odd $(2x+1)$

$0^{2x} 1^{2x}$ | $0^{2x+1} 1^{2x+1}$

$(00)^x (11)^x$ | $(00)^x 0 (11)^x 1$

$(00)^* (11)^*$ | $(00)^* 0 (11)^* 1$

$\therefore r = (00)^*(11)^* + (00)^* 0 (11)^* 1$

12) a) starts with '0' & $|w|$ = even $\Rightarrow$ 0 $\underbrace{\quad}_{\text{odd}}$

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

b) starts with '1' & $|w|$ = odd $\Rightarrow$ 1 $\underbrace{\quad}_{\text{even}}$

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

13) starts with '0' & donot contain two consecutive ones

$r = (0+01)^+$

14) strings do not contain two consecutive 0's or 1's

$r = (1+\epsilon)(01)^*(0+\epsilon) + (0+\epsilon)(10)^*(1+\epsilon)$

15) strings contains exactly two consecutive 1's

$r = 0^*110^*$

16) # string of length ≤ 3 generated by

$r = (1+01)^*0$

leng = 0 → { }

1 → {0}

2 → {10}

3 → {110, 010}

⇒ 4

17) # string of length ≤ 3 generated by

$r = (0+01)^* 1 (0+1)^*$

leng = 0 → { }

= 1 → {1}

2 → {01, 10, 11}

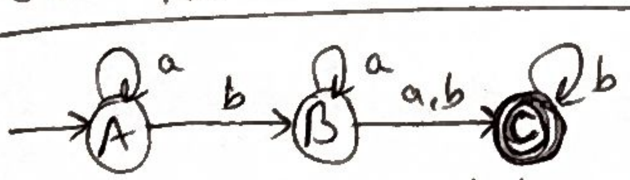
3 → {001, 011, 100, 111, 101, 110, 010}

⇒ 10 + 1 = 11

FA → RE

For NFAs & DFAs as FA

ex1:



M:

$A = \epsilon + Aa$

$B = Ab + Ba$

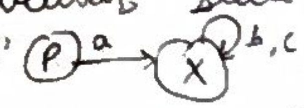
$C = B(a+b) + Cb$

what are this equ^n ?

How can we arrive at a particular state

↳ equ^n

ex:



$X = Pa + X(b+c)$

* **Arden's Lemma / Rule:** (Fails for ε-NFA)

If $R = Q + RP$ (can say linear equ^n in R) ($A = A^* B = ?$)

then

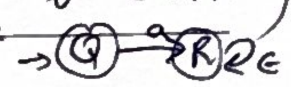
$R = QP^*$ & if P doesn't contain ε then QP^* is the unique sol^n

else

$R = Q + RP$ has ∞-many sol^n

[this is the reason why Arden's Lemma fails for ε-NFA]

Ex:



$R = Qa + R\epsilon$

[contains ε]

To verify substitute value of $R = QP^*$ in $R = Q + RP$
⇓
$QP^* = Q + QP^*P$
which are equal means LHS = RHS

Let us apply this rule for/on ex 1:

$$\overset{R}{A} = \overset{Q}{\epsilon} + \overset{RP}{Aa} \Rightarrow A = \epsilon a^* \Rightarrow A = a^*$$

$$B = Ab + Ba \Rightarrow \overset{R}{B} = \overset{Q}{a^*b} + \overset{RP}{Ba} \Rightarrow B = a^*ba^*$$

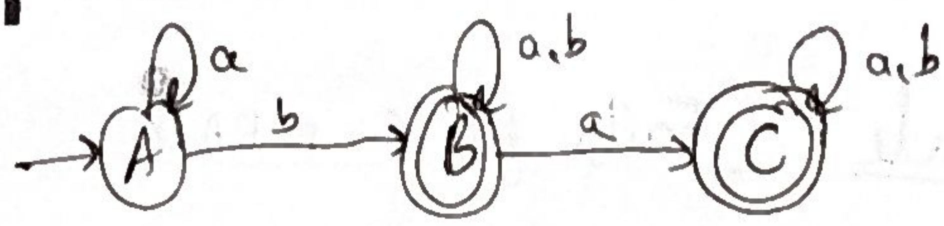
$$C = B(a+b) + Cb \Rightarrow \overset{R}{C} = \overset{Q}{a^*ba^*(a+b)} + \overset{RP}{Cb}$$

$$C = a^*ba^*(a+b)b^*$$

As C is the only final state in the given NFA, RE for NFA is

$$\boxed{RE = a^*ba^*(a+b)b^*}$$

2)



$A = \epsilon + Aa \Rightarrow A = \epsilon a^* = a^*$

Final States : B & C

$B = Ab + B(a+b) = a^*b + B(a+b) \Rightarrow B = a^*b(a+b)^*$

$C = Ba + C(a+b) = a^*b(a+b)^*a + C(a+b)$

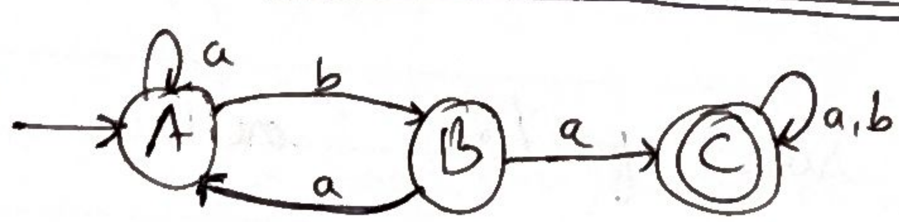
$\Downarrow$

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

As B & C both are final states RE corresponds to given NFA is B+C

$$\underline{RE = a^*b(a+b)^* + a^*b(a+b)^*a(a+b)^*}$$

3)



$A = \epsilon + Aa + Ba \Rightarrow A = \epsilon + Aa + Aba$

$B = Ab$

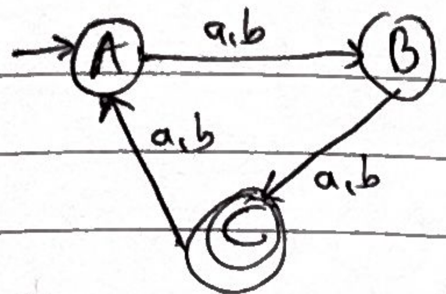
$A = \epsilon + A(a+ba)$

$B = (a+ba)^*b \Leftarrow \therefore A = \epsilon(a+ba)^*$

$C = Ba + C(a+b) \Rightarrow C = (a+ba)^*ba + C(a+b)$

$$\boxed{C = (a+ba)^*ba(a+b)^*}$$

4)



$A = \epsilon + C(a+b)$

$B = A(a+b)$

$C = B(a+b)$

Substituting C in A; $A = \epsilon + B(a+b)^2$

Now, substituting B in above A; $A = \epsilon + A(a+b)^3$

$\therefore A = \epsilon((a+b)^3)^*$

~~B = a+b~~ $B = A(a+b)$

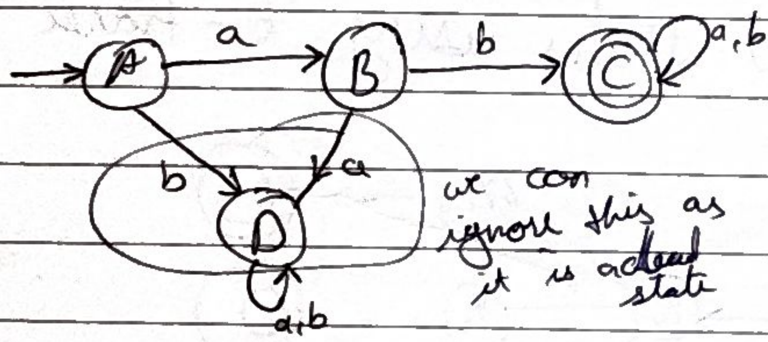
$= ((a+b)^3)^*(a+b)$

$C = B(a+b)$

$\boxed{C = ((a+b)^3)^*(a+b)(a+b)}$

This implies the Machine & RE is for/gives len of string $\equiv 2 \pmod 3$

5)



$A = \epsilon$

$B = Aa$

$C = B(a+b) + Bb$

$= Bb + C(a+b)$

$A = \epsilon$

$B = Aa = \epsilon a = a$

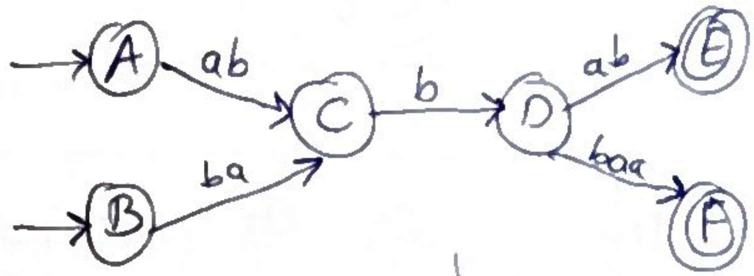
$C = Bb + C(a+b) = ab + C(a+b)$

$\Rightarrow$ Applying Arden's lemma $\Rightarrow$ $C = ab(a+b)^*$

☆ Transition Graph (TG)

↳ NFA + multiple init. states + transition on words

Ex:



Mathematically, $M = (Q, \Sigma, I, F, \delta)$

$I = \{A, B\}$

$F = \{E, F\}$

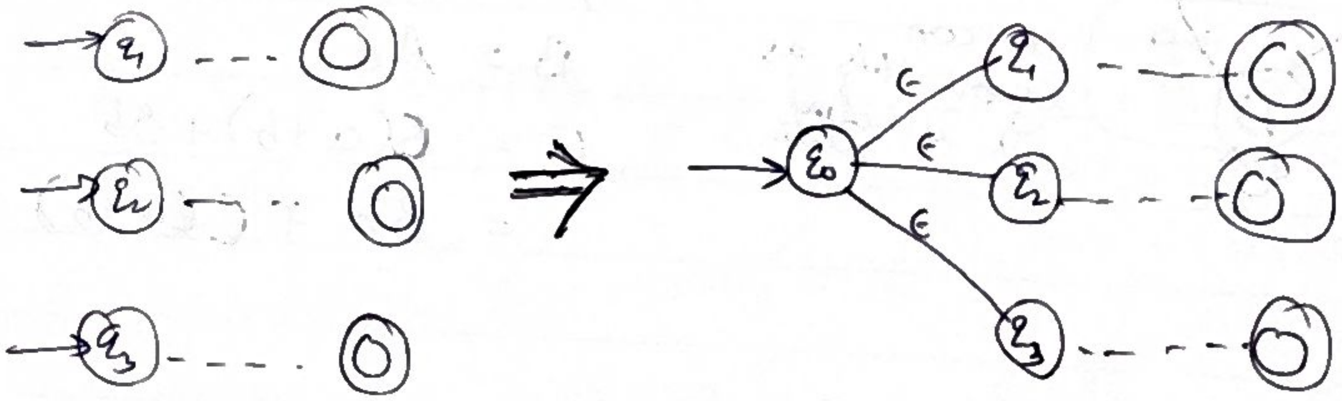
$Q = \{A, B, C, D, E, F\}$

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

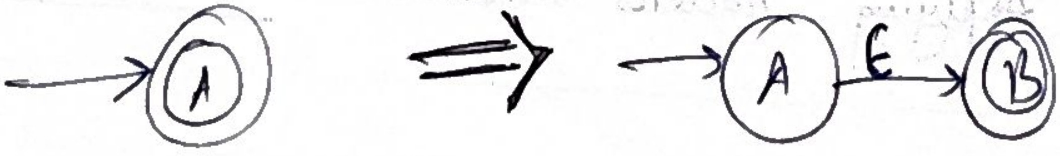
$$\delta : Q \times \Sigma^* \rightarrow 2^Q$$

* Elimination (State) Method (ϵ-NFA/NFA/DFA) TG → RE

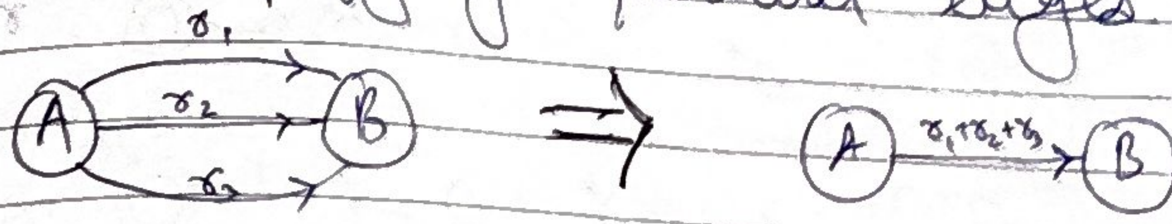
Step-1: Combine initial states using ϵ-transiⁿ



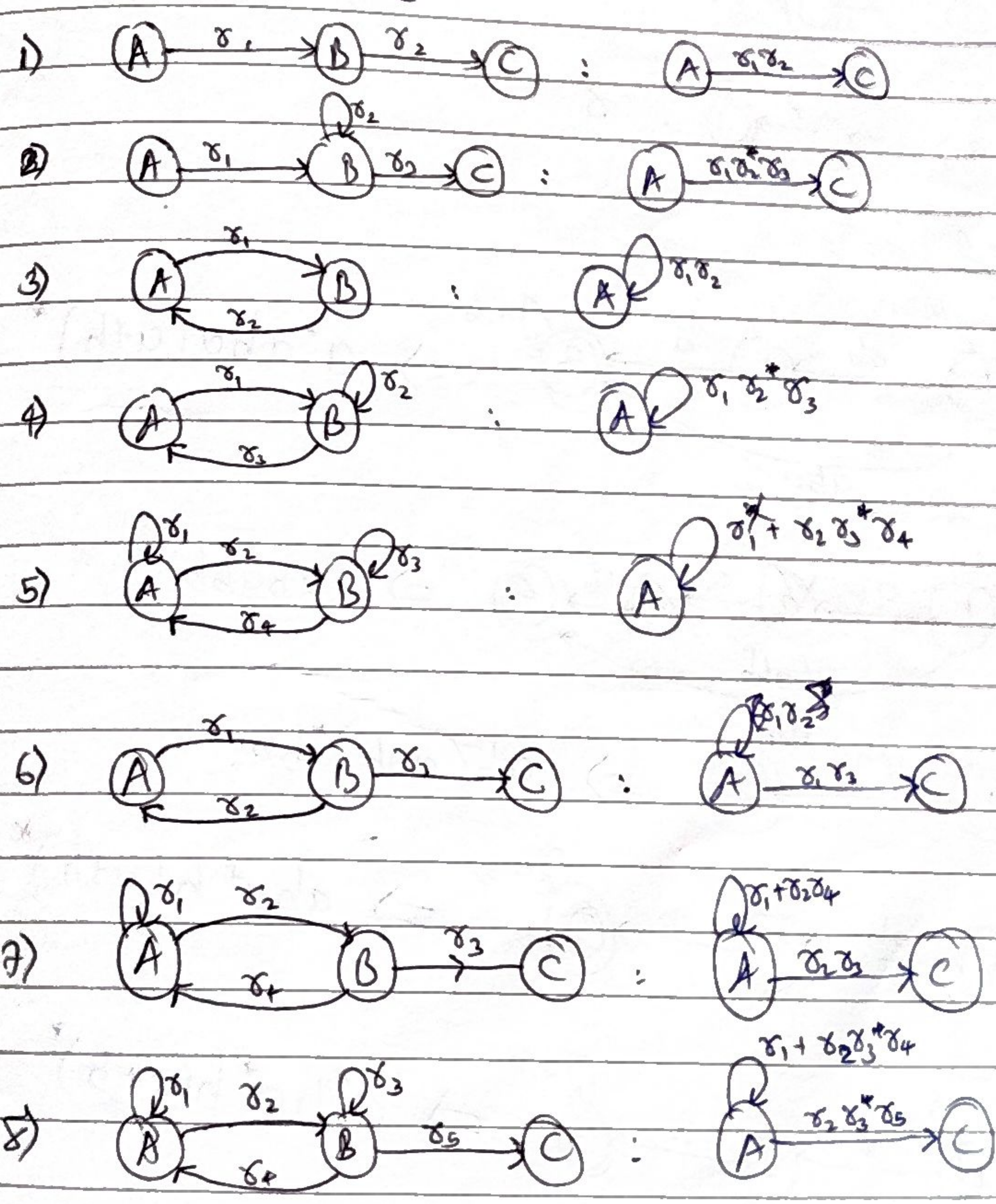
Step-2: Separate out initial & final states using ϵ-transiⁿ



Step 3: Simplifying parallel edges



Step-4: Eliminating states

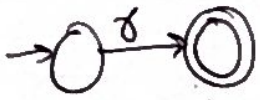


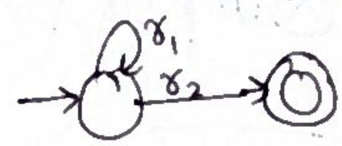
Step 5 : If TG / ε-NFA / NFA / DFA is any of the following, write RE

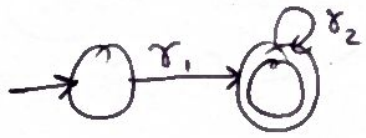
1.  : r^*

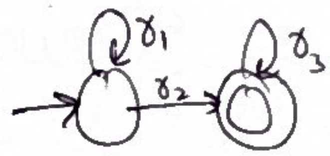
2)  : $(r_1+r_2)^*$

3)  : $(r_1 r_2)^*$

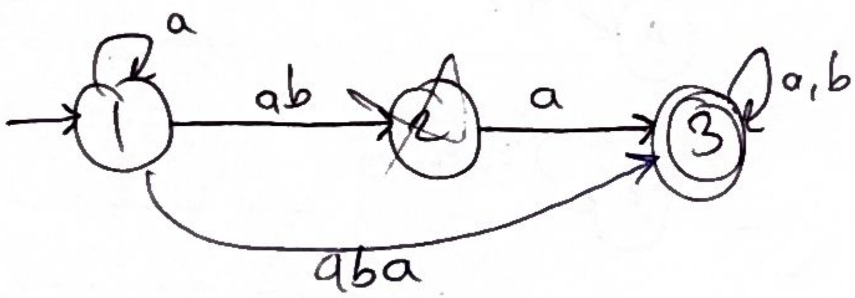
4)  : r

5)  : $r_1^* r_2$

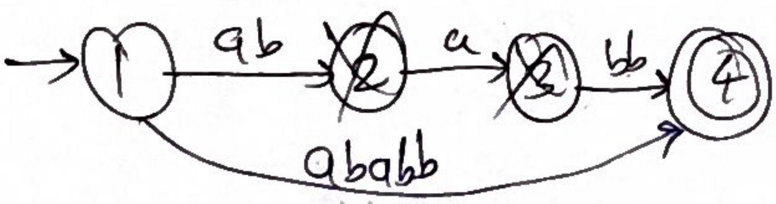
6)  : $r_1 r_2^*$

7)  : $r_1^* r_2 r_3^*$

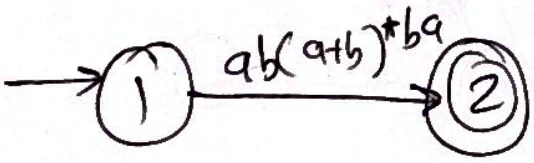
Exercise

1) 

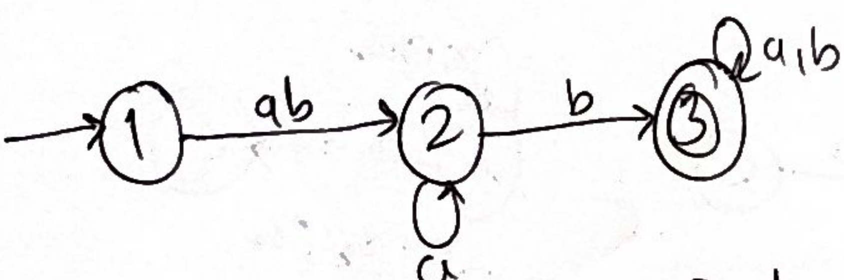
$\Rightarrow a^*aba(a+b)^*$

2) 

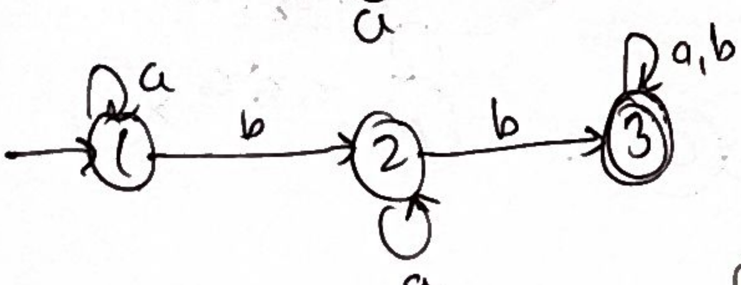
$\Rightarrow ababb$

3) 

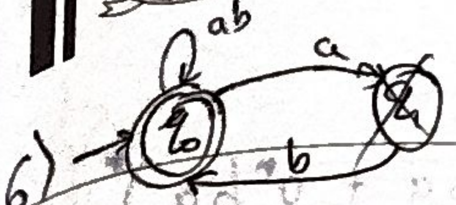
$\Rightarrow ab(a+b)^*ba$

4) 

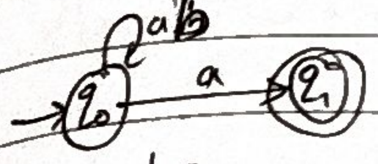
$\Rightarrow aba^*b(a+b)^*$

5) 

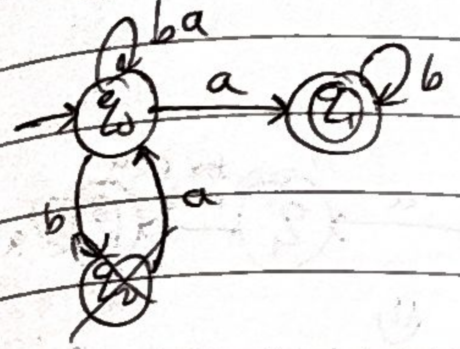
$\Rightarrow a^*ba^*b(a+b)^*$

6) 

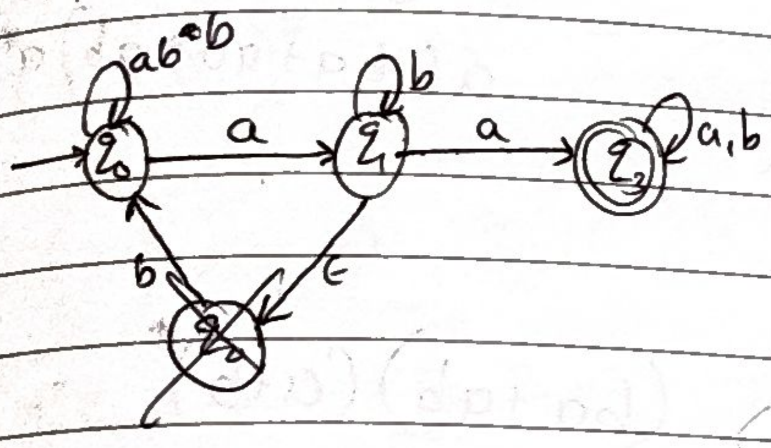
$\Rightarrow (ab)^*$

7) 

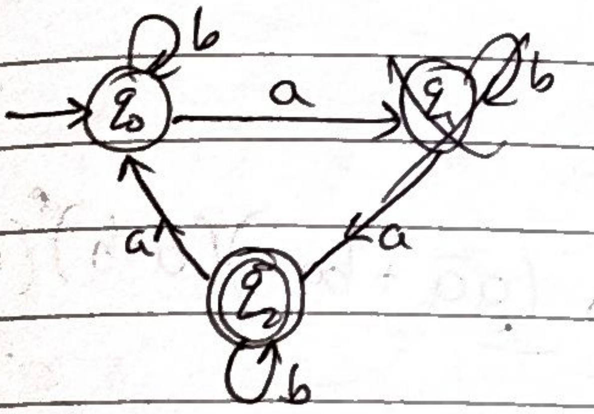
$\Rightarrow (ab)^*a$

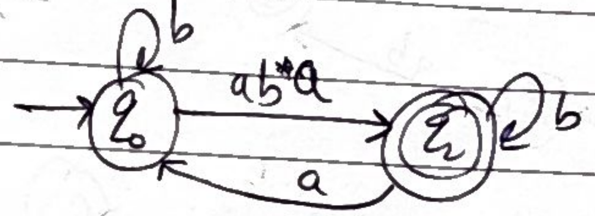
8) 

$\Rightarrow (ba)^*ab^*$

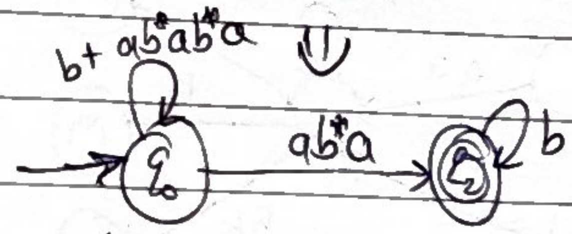
9) 

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

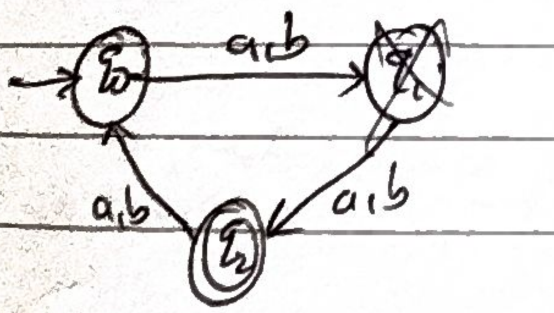
10) 

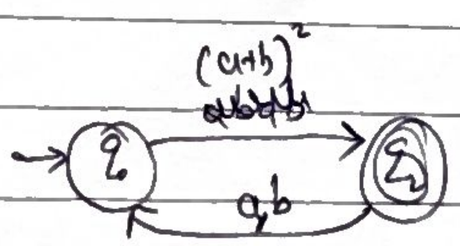
$\Rightarrow$ 

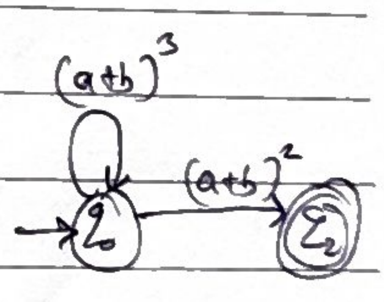
$\Downarrow$



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

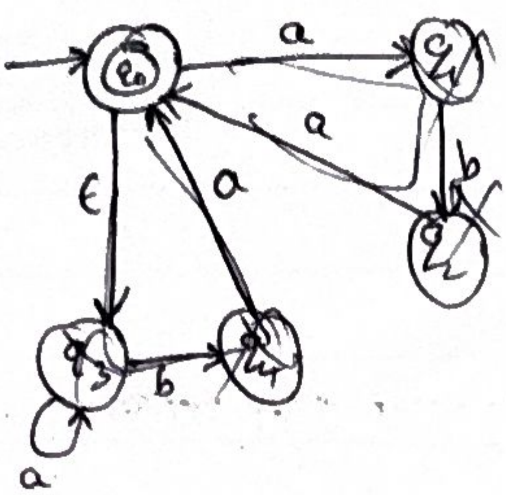
11) 

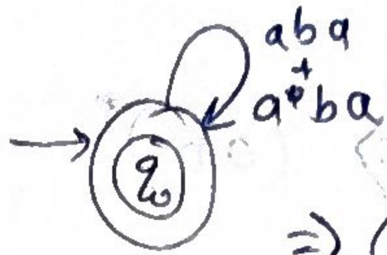
$\Rightarrow$ 

$\Rightarrow$ 

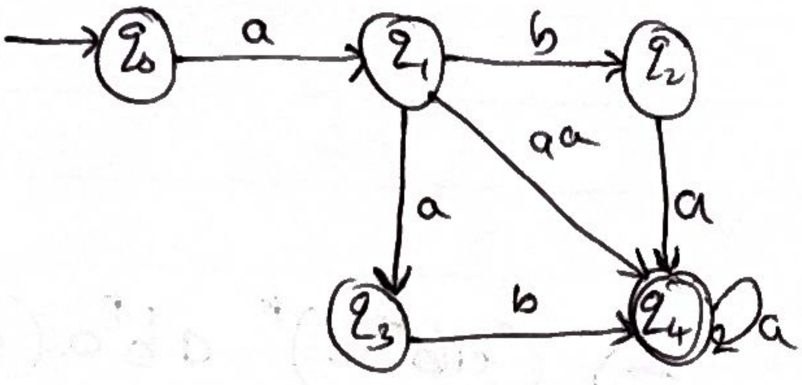
$\Downarrow$

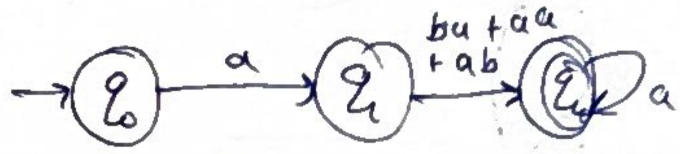
$((a+b)^3)^*(a+b)^2$

12) 

$\Rightarrow$ 

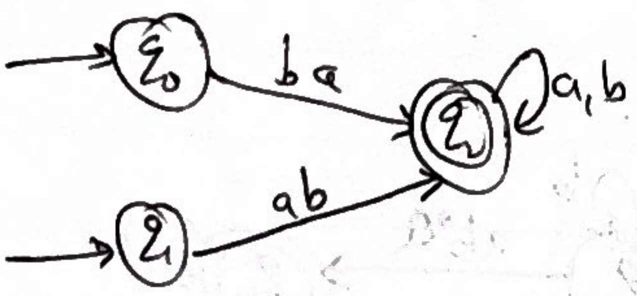
$\Rightarrow (aba + a^*ba)^*$

13) 

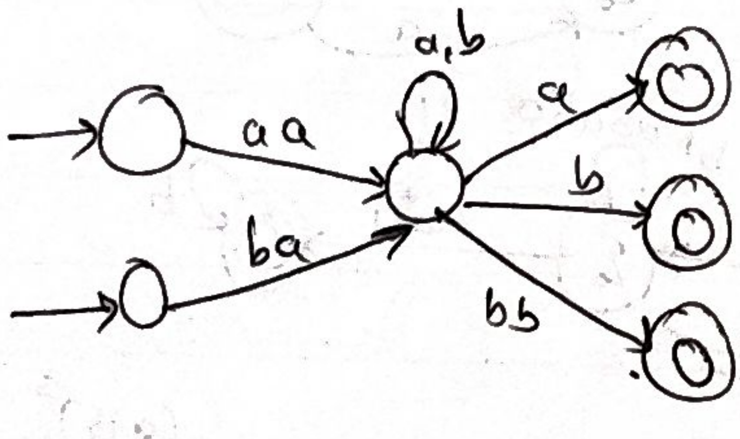
$\Rightarrow$ 

$\Downarrow$

$a(ba + aa + ab)a^*$

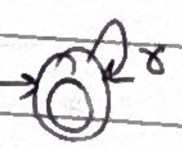
14) 

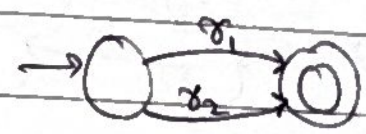
$\Rightarrow (ba + ab)(a+b)^*$

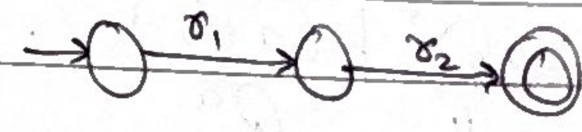
15) 

$\Rightarrow (aa + ba)(a+b)^*(a+b+bb)$

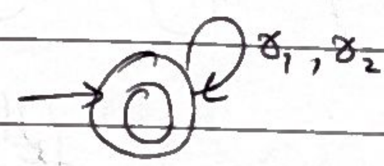
RE → FA (ε-NFA / NFA / DFA)

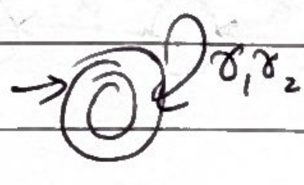
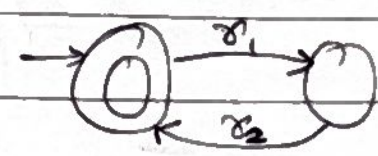
1) Kleen Closure : r^* 

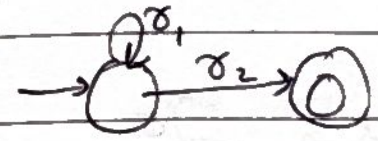
2) Union : $r_1 + r_2$ 

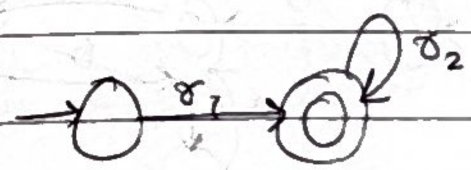
3) Concatenaⁿ : $r_1 r_2$ 

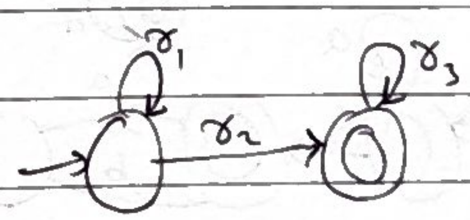
Examples

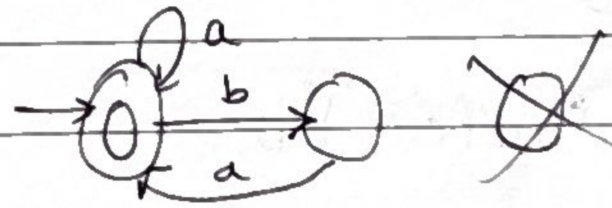
1) $(r_1 + r_2)^*$ ⇒ 

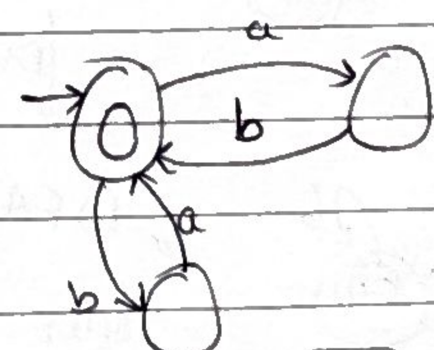
2) $(r_1 r_2)^*$ ⇒  = 

3) $r_1^* r_2$ ⇒ 

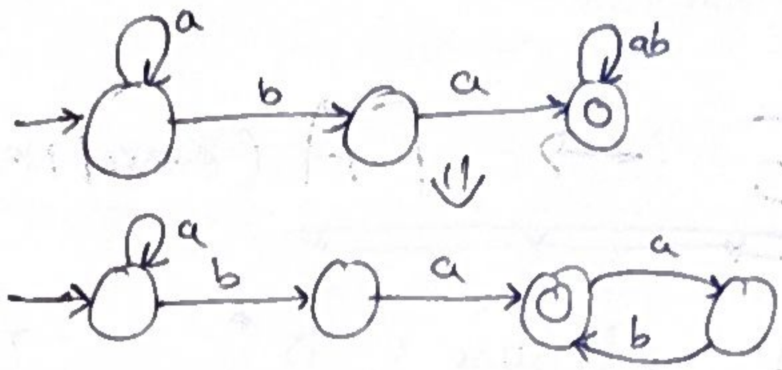
4) $r_1 r_2^*$ ⇒ 

5) $r_1^* r_2 r_3^*$ ⇒ 

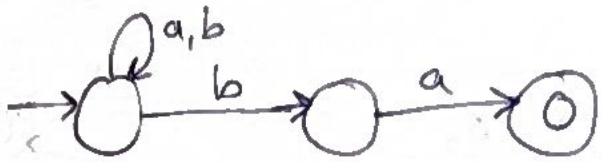
6) $(a^* b a)^*$ ⇒ 

7) $(ab + ba)^*$ ⇒ 

8) $a^*ba(ab)^*$ ⇒

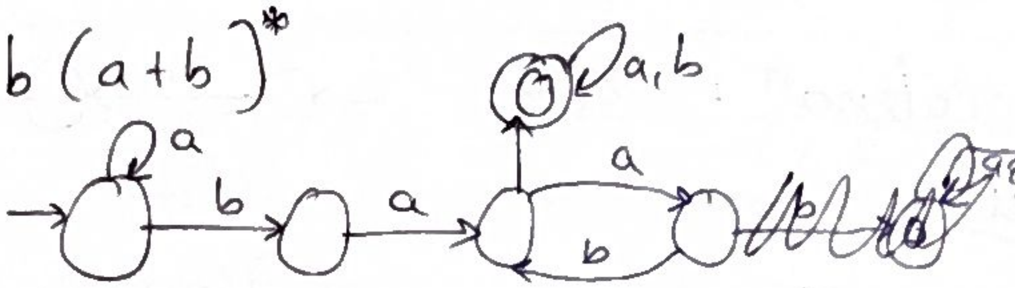


9) $(a+b)^*ba$ ⇒



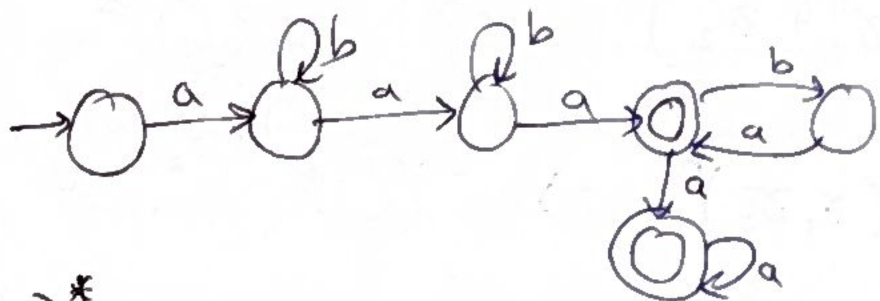
10) $a^*ba(ab)^*b(a+b)^*$

↳



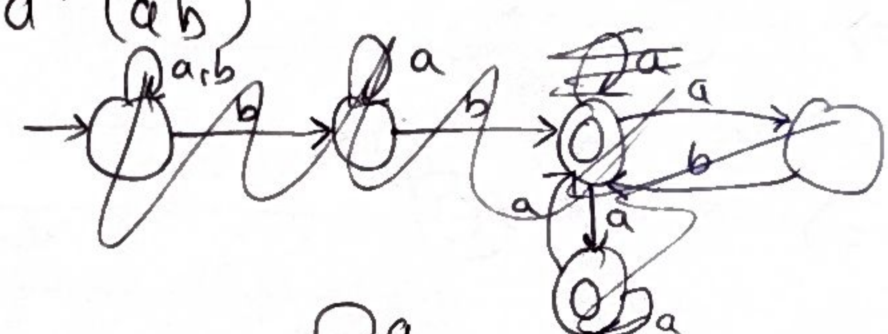
11) $ab^*ab^*a(ba)^*a^*$

⇒

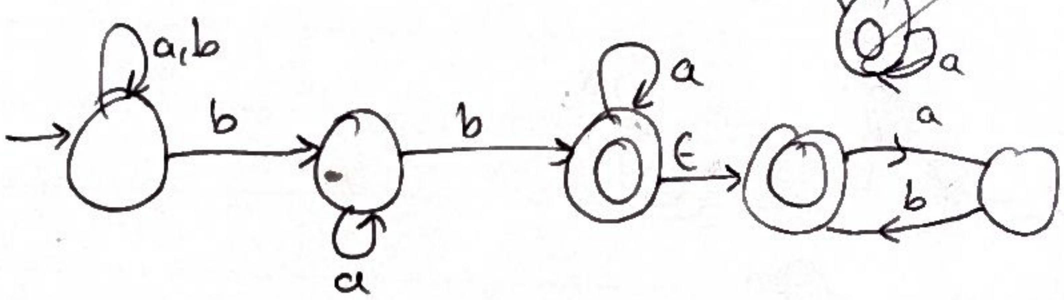


12) $(a+b)^*ba^*ba^*(ab)^*$

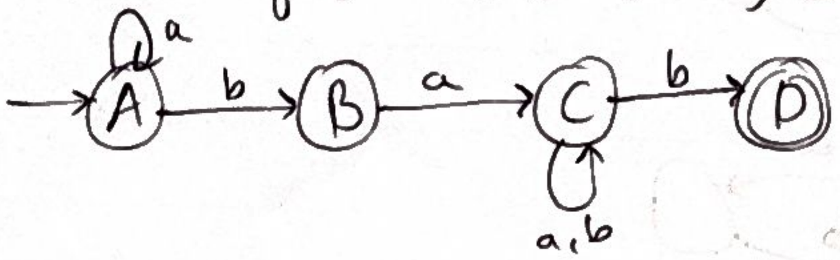
⇒



⇒

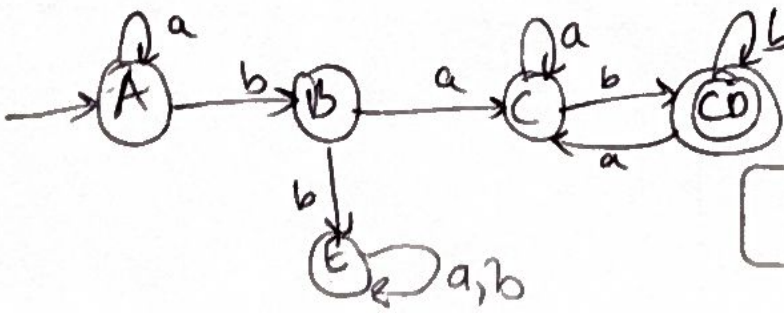


13) Min. DFA for $a^*ba(a+b)^*b$



← NFA

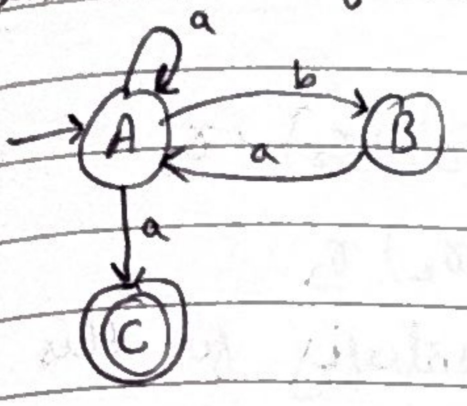
↓ DFA

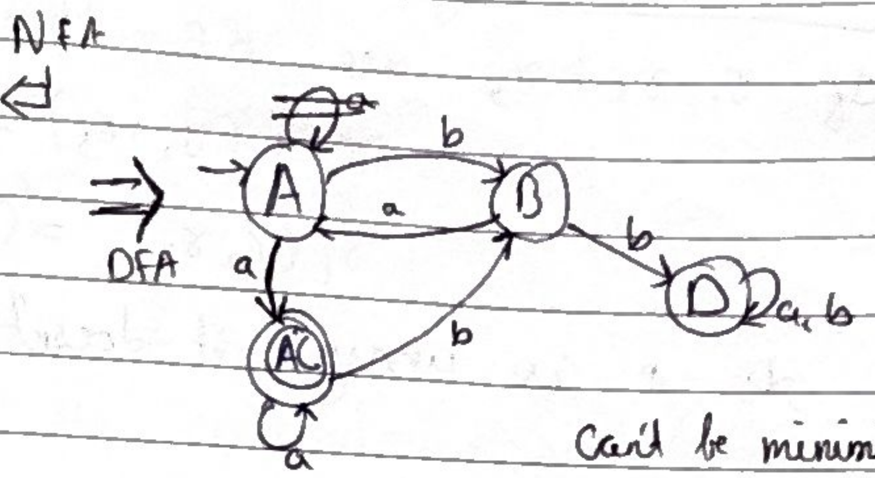


← DFA

After applying equivalence algo. we found that this is the only min DFA

(4) Min DFA for $(a+ba)^* a$





Can't be minimized more

☆ Algebraic Properties of RE operators.

$*$: Kleen Closure (unary)
$+$: Union } (binary)
$\cdot$: concatenaⁿ

We will check diff properties using/with these operators, and the properties are related to the concept of Algebraic structure {SG, Monoids, groups, abelian group ...}

1) Closure :

If r_1, r_2 are RE then,

r_1^* is RE
$r_1 + r_2$ is RE
$r_1 r_2$ is RE

2. Associativity

If r_1, r_2 & r_3 are RE then,

$$r_1 + (r_2 + r_3) = (r_1 + r_2) + r_3$$

$$r_1 (r_2 r_3) = (r_1 r_2) r_3$$

As * is unary, it doesn't satisfy for this.

3) Identity

As we know, for identity $\forall r \ \exists x$ such that

$r + x = r \Rightarrow x = \phi$ $\quad \underline{r \cup \phi = r}$

Hence $x = \underline{\phi}$ is identity element for +

Similarly for ·

$r_1 \cdot x = r_1 \Rightarrow r_1 \cdot \epsilon = r_1$

$\therefore x = \epsilon$ is identity for ·

- Inverse is not possible for RE
  ↳ as $r + x =$ identity is the condition of inverse
  $x = (-r)$ is the only option which isn't possible

4) Annihilator

$r + x = x \Rightarrow r \cup x = x$ (Not possible)
Annihilator isn't possible for +

$r \cdot x = x \Rightarrow r \cdot \phi = \phi$

$\therefore \phi$ is the annihilator for ·

5) Commutative property :

If r_1, r_2 are RE then

$$r_1 + r_2 = r_2 + r_1$$

$$r_1 r_2 \neq r_2 r_1$$

Hence, + satisfies the commutative property & · doesn't.

6) Distributive property :

$$(r_1 + r_2) \cdot r_3 = r_1 \cdot r_3 + r_2 \cdot r_3 \quad \text{(Right)}$$
$$r_1 \cdot (r_2 + r_3) = r_1 \cdot r_2 + r_1 \cdot r_3 \quad \text{(Left)}$$

$$(r_1 \cdot r_2) + r_3 \neq (r_1 + r_3) \cdot (r_2 + r_3)$$
$$r_1 + (r_2 \cdot r_3) \neq (r_1 + r_2) \cdot (r_1 + r_3)$$

7) Idempotent property :

$r + r = r \Rightarrow$ + satisfies Idempotent property

$r \cdot r \neq r \Rightarrow$ · (concat.) doesn't.

{ if $r = \epsilon$ only, $\epsilon \cdot \epsilon = \epsilon$ (but not always) }

Set : Group of elements / RE
Algebraic structure : Set, Operaⁿ (closure)
Semigroup : closure, Associative
Monoid : Closure, Associative, Identity
Group : Closure, Associative, Identity & Inverse
Abelian group :: Group + Commutative