Page No.
Date / /

```
main ()
{
    FILE *fp;
    fp = fopen ("abc.txt", "r"); // opening the file
    yyin = fp;                  // Make tokens & do fp's work
    yylex ();
}
```

Regular Expressions

→ The set of all words ending by either 'bb' or 'a'

RE: $(a+b)^* (bb+a)$

→ The set of all strings with even no. of 'a' followed by odd no. of b's.

RE: $(aa)^* (bb)^* b$

→ The set of all words which consist word with exactly two 'aa'

RE: $b^* a b^* a b^*$

→ Find R.E. representing the set of strings – $a^m b^{2m} c^{3p}$, $m, n, p \geq 1$

$a^+ (bb)^+ (ccc)^+$

Scanned with CamScanner

13 / Feb / 18

Imp.

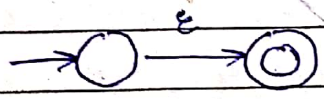
Thompson Algorithm

* To convert any regular expression into an NFA.
* **Input** : A regular expression over alphabet Σ.
* **Output** : An NFA 'N' accepting language $L(r)$.
* **Method** : It starts by parsing 'r' into subexpressions.

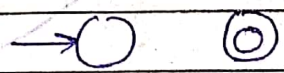
The rule for constructing an NFA consists of basic rule for handling sub-expressions with no operators, an inductive rule for constructing large NFA from the NFA for the immediate sub-expression of a given expression.

* **Induction** : Suppose N(s) and N(t) are NFA for regular expressions 's' and 't'.

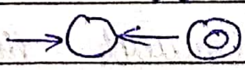
1) ε



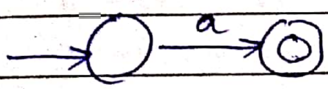
3) ϕ



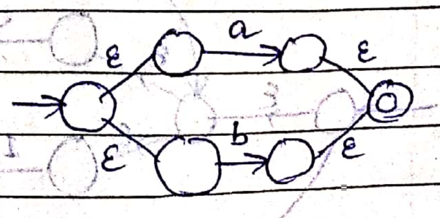
or



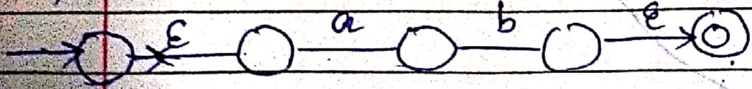
2) {a}



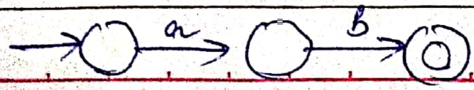
4) (a + b)



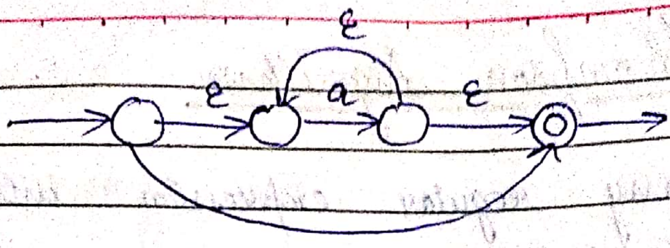
5) ab



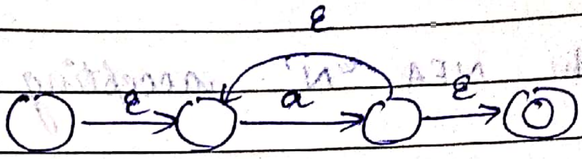
or



6) a^*

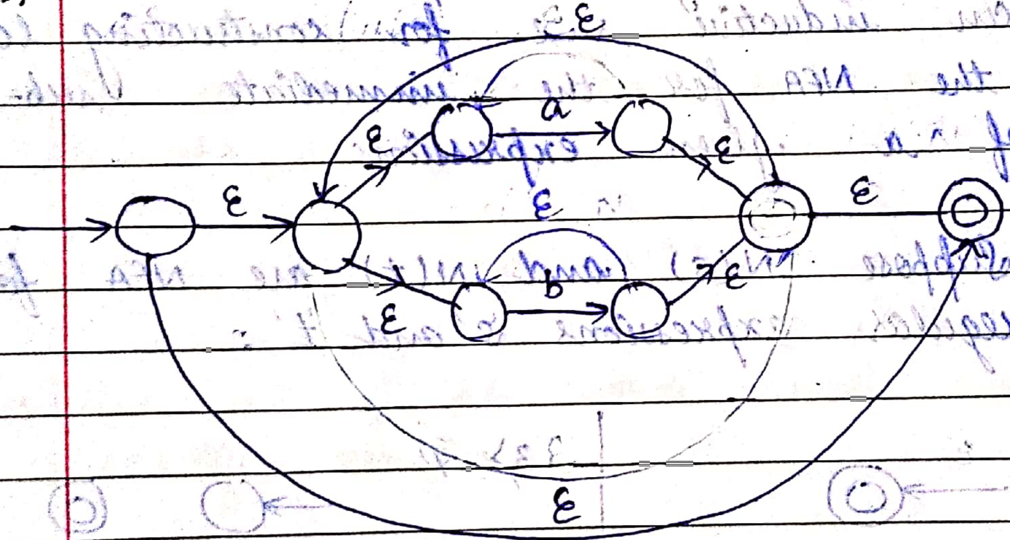


7) a^+

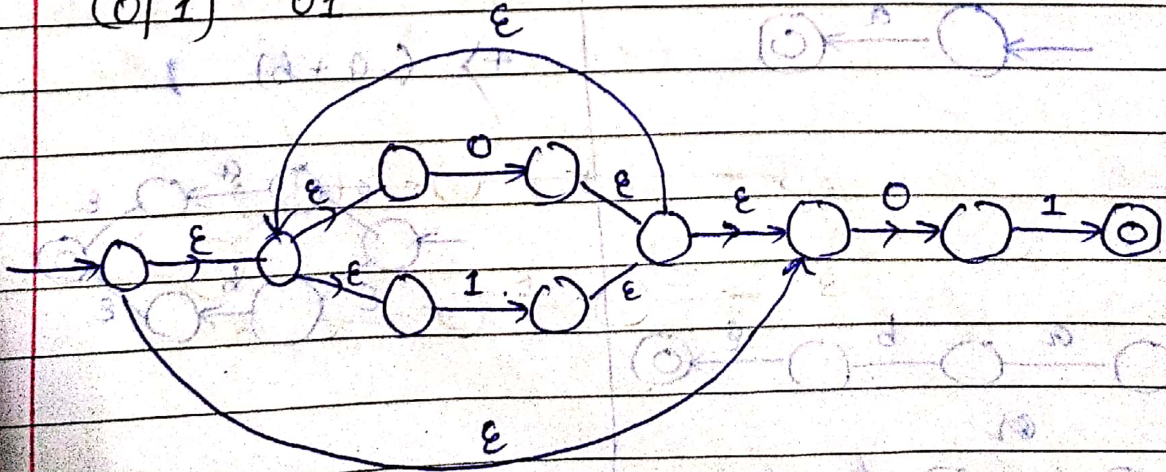


~~8) $(a+b)^*$~~

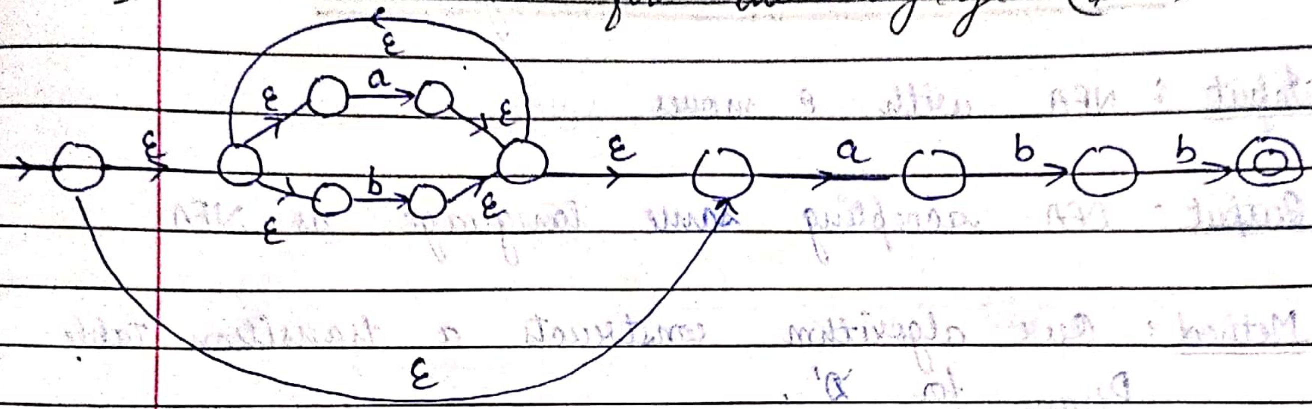
Q- Find the NFA with ε moves for R.E. $(a+b)^*$



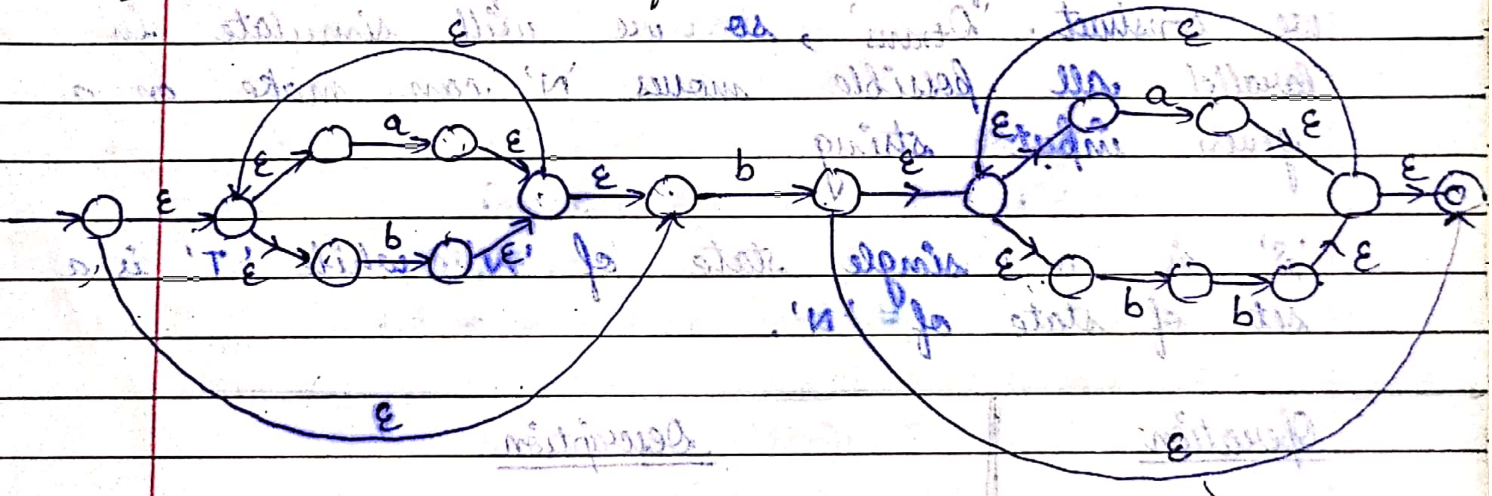
Q- Convert the following RE into NFA.
$(0|1)^*\ 01$



Q- Construct a NFA for the language $(a+b)^*abb$



Q- Construct an NFA for $(a+b)^* b (a+bb)^*$.



Grammar for Semicolon

$S \rightarrow id = E;$

$E \rightarrow E+T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow G \uparrow F \mid G$

$G \rightarrow id \mid num$

Page No.
Date / /

Imp: Subset Set Construction Algorithm

★ **Input** : NFA with ε moves.

★ **Output** : DFA accepting same language as NFA.

★ **Method** : Our algorithm constructs a transition table D_{trans} for 'D'.

Each state of 'D' is a set of NFA state and we construct 'D_{trans}', so we will simulate in parallel all possible moves 'N' can make on a given input string.

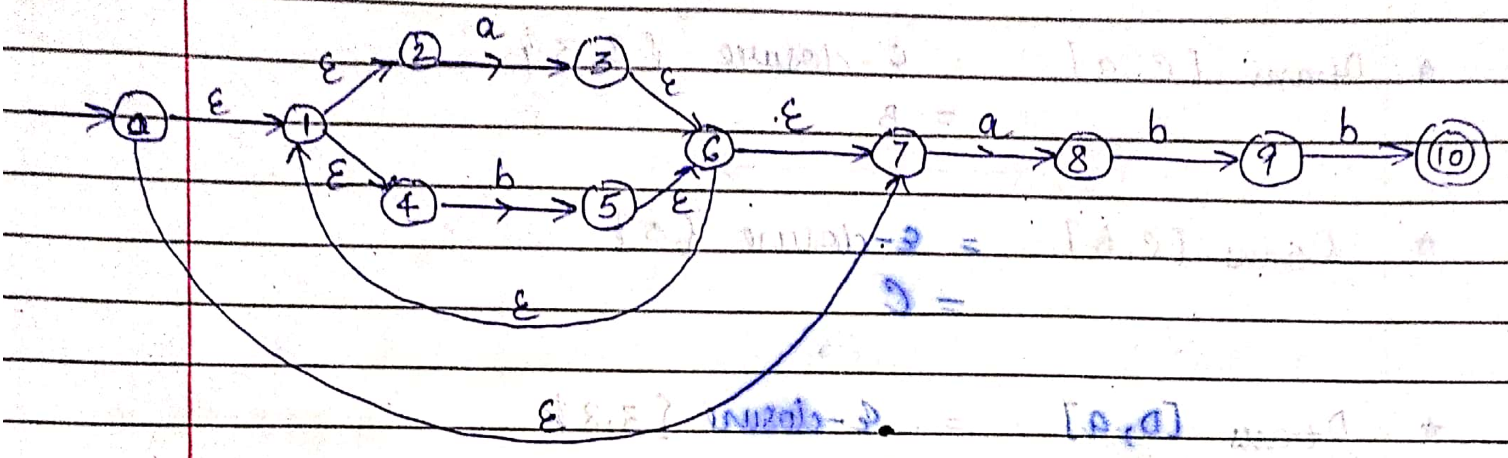
'S' is a single state of 'N' while 'T' is a set of state of 'N'.

Operation	Description
1) ε- Closure (S) $S \rightarrow \{0\}$	Set of NFA states reachable from NFA state 's' on 'ε' - transition alone
2) ε- Closure (T)	Set of NFA states reachable from some NFA state 's' in set 'T' on ε- transition
3) move (T, a) state input	Set of NFA state to which there is 'a' transition on input symbol 'a' from some state 'S' in 'T'.

14 / Feb / 18

Q- Find the DFA for a regular expression $(a+b)^* abb$.

Step-1) NFA with ε moves



★ A = ε-closure (0)
= {0, 1, 2, 4, 7}

★ D_{trans} [A, a] = ε-closure {move (0,a) ∪ move (1,a) ∪ move (2,a) ∪ move (4,a) ∪ move (7,a)}

ε-closure
= {3, 8}
= {3, 6, 7, 1, 2, 4, 8} = B

★ D_{trans} [A, b] = ε-closure {move (0,b) ∪ move (1,b) ∪ move (2,b) ∪ move (4,b) ∪ move (7,b)}

= ε-closure {5}
= {5, 6, 7, 1, 2, 4} = C

★ D_{trans} [B, a] = ε-closure {move (1,a) ∪ move (2,a) ∪ move (3,a) ∪ move (4,a) ∪ move (6,a) ∪ move (7,a) ∪ move (8,a)}
= ε-closure {3, 8} = B

☆ D_{trans} [B, b] = ε-closure {5, 9}
= {5, 6, 7, 1, 2, 4, 9}
= D

☆ D_{trans} [C, a] = ε-closure {8, 3}
= B

☆ D_{trans} [C, b] = ε-closure {5}
= C

☆ D_{trans} [D, a] = ε-closure {3, 8}
= B

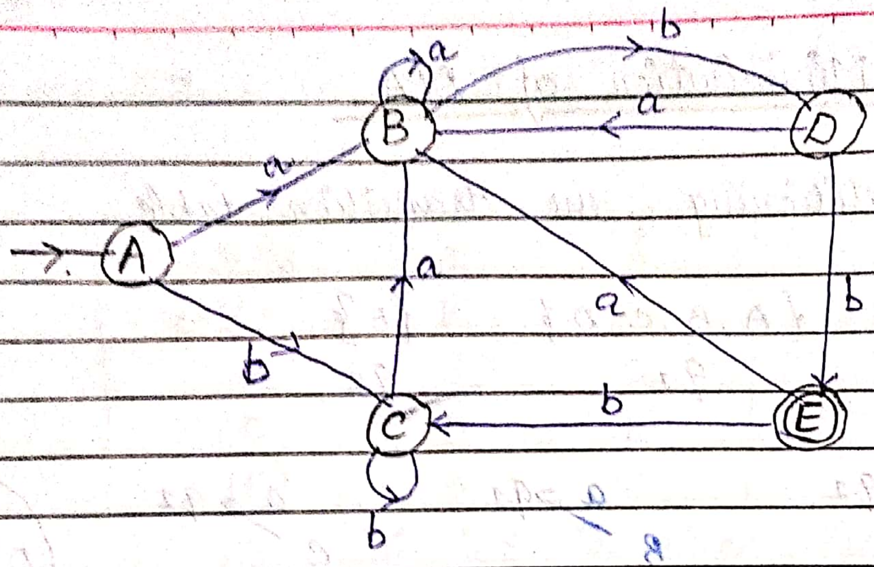
☆ D_{trans} [D, b] = ε-closure {5, 10}
= {5, 6, 1, 2, 4, 7, 10}
= E

☆ D_{trans} [E, a] = ε-closure {3, 8}
= B

☆ D_{trans} [E, b] = ε-closure {5}
= C

States	a	b
→ A	B	C
B	B	D
C	B	C
D	B	E
E	B	C

Fig. New transition table



Algorithm

while (there is an unmarked state - T in D state)

{

marked = T;

for (each I/P terminal 'a')

{

U = ε-closure(move(T, a))

if (U is not in T-state)

{

add U as unmarked state

}

Dtrans (T, a) = U;

}

}

}

Minimisation of DFA

eg: Partitioning the transition table.

$\pi_0 = \{A, B, C, D\} \quad \{E\}$

$\qquad\quad q_1 \qquad\qquad q_2$

$A \xrightarrow{a} q_1$, $A \xrightarrow{b} q_1$ | $B \xrightarrow{a} q_1$, $B \xrightarrow{b} q_1$ | $C \xrightarrow{a} q_1$, $C \xrightarrow{b} q_1$ | $D \xrightarrow{a} q_1$, $D \xrightarrow{a} q_2$

$\pi_1 = \{A, B, C\} \quad \{D\} \quad \{E\}$

$\qquad\quad q_1 \qquad\quad q_2 \qquad q_3$

$A \xrightarrow{a} q_1$, $A \xrightarrow{b} q_1$ | $B \xrightarrow{a} q_1$, $B \xrightarrow{b} q_2$ | $C \xrightarrow{a} q_1$, $C \xrightarrow{b} q_1$

$\pi_2 = \{A, C\} \quad \{B\} \quad \{D\} \quad \{E\}$

$\qquad\quad q_1 \qquad q_2 \qquad q_3 \qquad q_4$

$A \xrightarrow{a} q_2$, $A \xrightarrow{b} q_1$ | $C \xrightarrow{a} q_2$, $C \xrightarrow{b} q_1$

$\pi_3 = \{A, C\} \{B\} \{D\} \{E\}$

Page No.
Date / /

⊛ Since $\pi_2 = \pi_3 = \pi_{final} = \{A, C\} \{B\} \{D\} \{E\}$

states	a	b
→ AC	B	AC
B	B	D
D	B	E
(E)	B	AC

Fig. Minimised Transition Table.

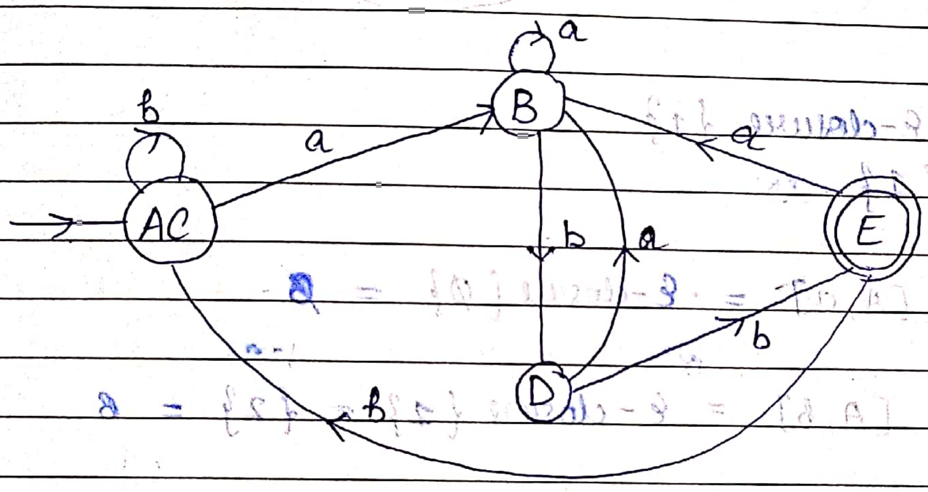
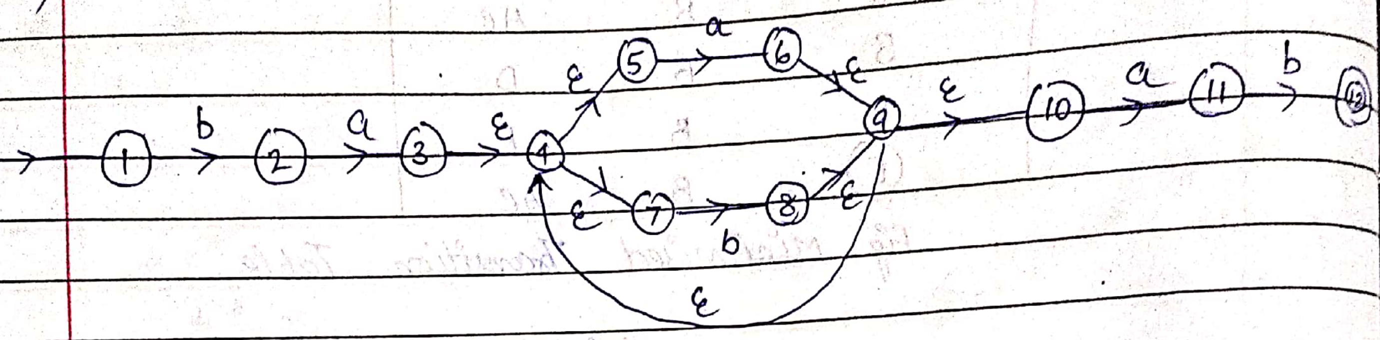


Fig. Minimised DFA.

Date / /

Q. Construct a minimised DFA for a regular expression ba $(a+b)^+$ ab

Step-1)



☆ A = ε-closure {1}
= {1}

☆ Dtrans [A, a] = ε-closure {∅} = ∅

☆ Dtrans [A, b] = ε-closure {2} = {2} = B

☆ Dtrans [B, a] = ε-closure {3} = {3, 4, 5, 7} = C

☆ Dtrans [B, b] = ε-closure {∅} = ∅

☆ Dtrans [C, a] = ε-closure {6} = {6, 9, 10, 4, 5, 7}
= D

Dtrans [C, b] = ε-closure {8} = {8, 9, 10, 4, 5, 7}
= E

☆ Dtrans [D, a] = ε-closure {11, 6} = {11, 6, 9, 10, 4, 5, 7}
= F

Page No.
Date / /

$D_{trans}[D, b] = \epsilon\text{-closure}\{8\} = \{8, 9, 10, 4, 5, 7\}$
$= E$

☆ $D_{trans}[E, a] = \epsilon\text{-closure}\{11, 6\} = F$

$D_{trans}[E, b] = \epsilon\text{-closure}\{8\} = E$

☆ $D_{trans}[F, a] = \epsilon\text{-closure}\{11, 6\} = F$

$D_{trans}[F, b] = \epsilon\text{-closure}\{12, 8\} = \{12, 8, 9, 10, 4, 5, 7\}$
$= G$

☆ $D_{trans}[G, a] = \epsilon\text{-closure}\{11, 6\} = F$

$D_{trans}[G, b] = \epsilon\text{-closure}\{8\} = E$

state	a	b
→ A	∅	B
B	C	∅
C	D	E
D	F	E
E	F	E
F	F	G
G	F	E

Fig. New transition table

Scanned with CamScanner

Minimisation

$\pi_0 = \{A, B, C, D, E, F\} \quad \{G\}$

$\qquad\qquad q_1 \qquad\qquad\qquad q_2$

$A \begin{cases} \phi \\ q_1 \end{cases} \qquad B \begin{cases} q_1 \\ \phi \end{cases} \qquad C \begin{cases} q_1 \\ q_1 \end{cases}$

$D \begin{cases} q_1 \\ q_1 \end{cases} \qquad E \begin{cases} q_1 \\ q_1 \end{cases} \qquad F \begin{cases} q_1 \\ q_2 \end{cases}$

$\pi_1 = \{A\} \quad \{B\} \quad \{C, D, E\} \quad \{F\} \quad \{G\}$

$\qquad\quad q_1 \quad\;\; q_2 \qquad\quad q_3 \qquad\quad q_4 \quad\; q_5$

$C \begin{cases} q_3 \\ q_3 \end{cases} \qquad D \begin{cases} q_4 \\ q_3 \end{cases} \qquad E \begin{cases} q_4 \\ q_3 \end{cases}$

$\pi_2 = \{A\} \quad \{B\} \quad \{C\} \quad \{D, E\} \quad \{F\} \quad \{G\}$

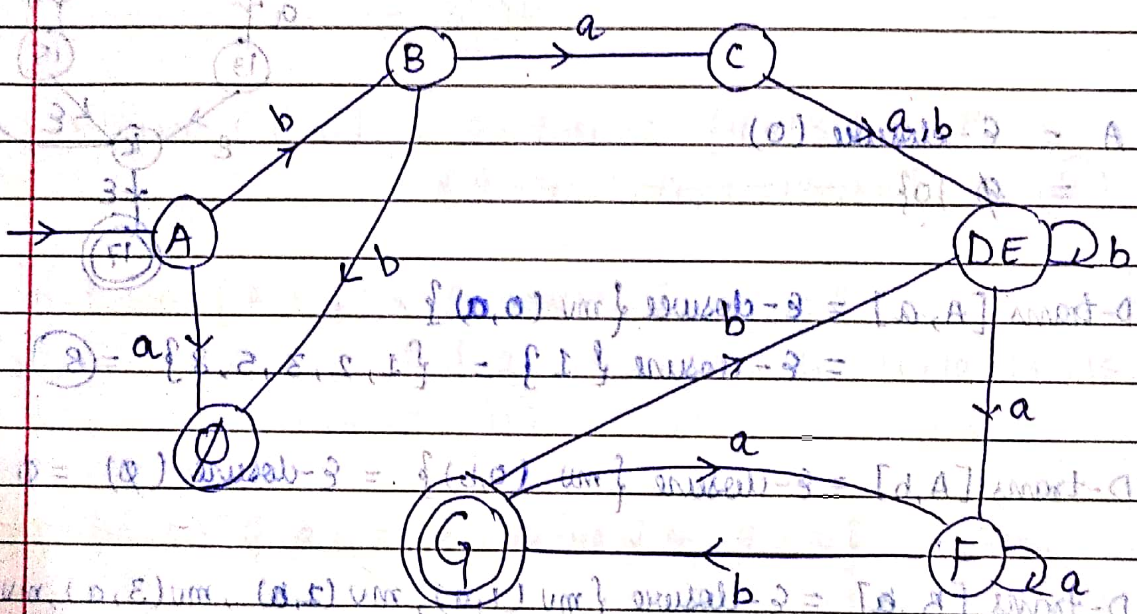
$\qquad\quad q_1 \quad\;\; q_4 \quad\;\; q_5 \qquad q_6 \qquad\; q_7 \quad\;\; q_8$

$D \begin{cases} q_7 \\ q_6 \end{cases} \qquad E \begin{cases} q_7 \\ q_6 \end{cases}$

$\pi_3 = \{A\} \quad \{B\} \quad \{C\} \quad \{D\ E\} \quad \{F\} \quad \{G\}$

$\boxed{\pi_2 = \pi_3 = \pi_{final}}$

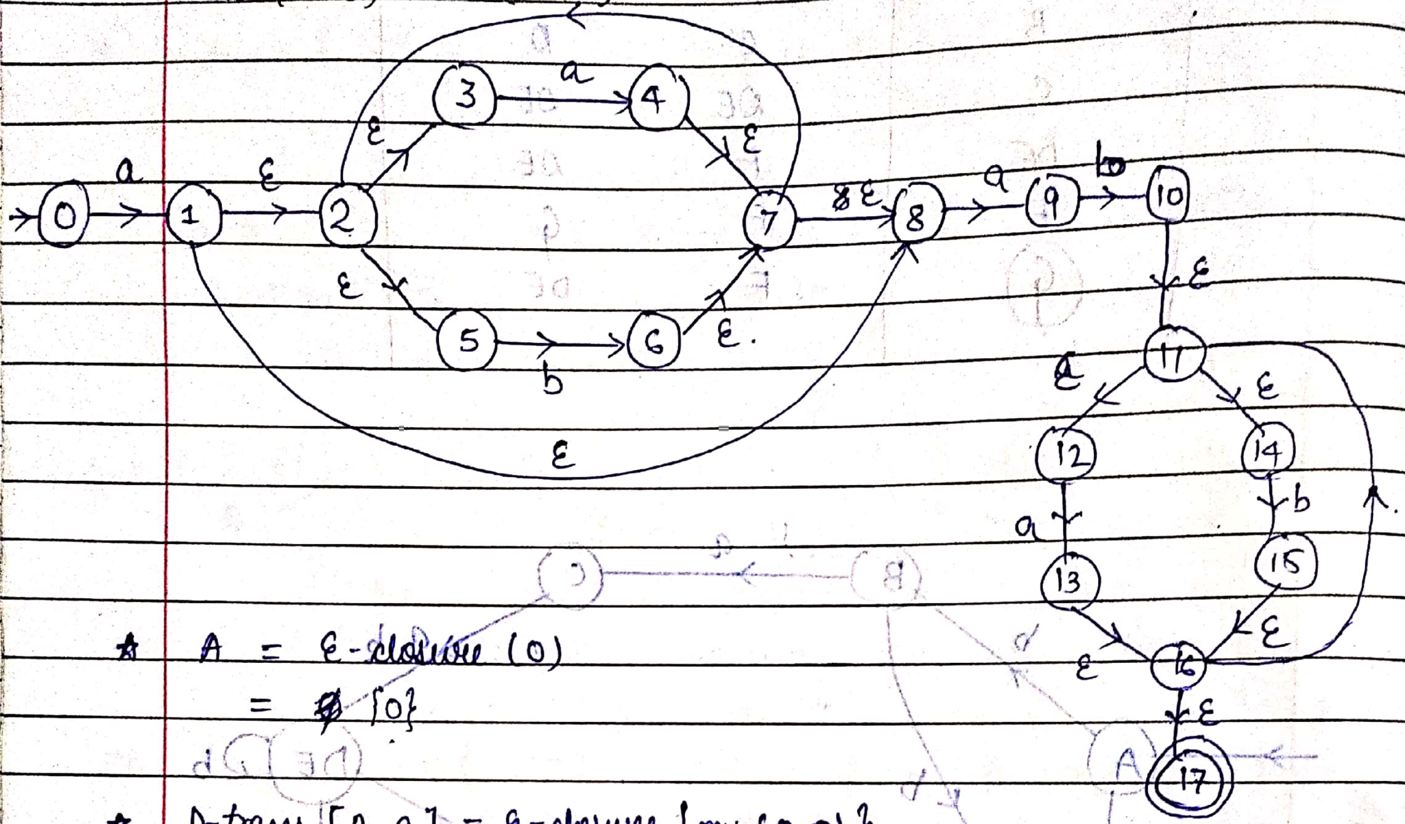
states	a	b
→ A	∅	B
B	C	∅
C	DE	DE
DE	F	DE
F	F	G
G	F	DE



Minimised DFA

15/ Feb /18

Q- Construct the minimised DFA for a RE
$a(a+b)^* ab (a+b)^+$



★ A = ε-closure (0)
= {0}

★ D-trans [A, a] = ε-closure {mv (0,a)}
= ε-closure {1} = {1, 2, 3, 5, 8} = B

★ D-trans [A, b] = ε-closure {mv (0,b)} = ε-closure (∅) = ∅

★ D-trans [B, a] = ε-closure {mv (1,a), mv (2,a), mv(3,a), mv(5,a)
mv (8,a)}
= ε-closure {4, 9}
= {4, 7, 8, 2, 3, 5, 9} = C

★ D-trans [B, b] = ε-closure {6} = {6, 7, 2, 3, 5, 8} = D

★ D-trans [C, a] = ε-closure {4, 9} = C

★ D-trans [C, b] = ε-closure {6, 10}
= {6, 7, 2, 3, 5, 8, 10, 11, 12, 14} = E

Page No.
Date / /

★ D-trans [D, a] = ε-closure {4, 9} = C

D-trans [D, b] = ε-closure {6} = D

★ D-trans [E, a] = ε-closure {4, 9, 13}
= {4, 7, 8, 2, 3, 5, 9, 13, 16, 17, 11, 12, 14} = F

D-trans [E, b] = ε-closure {6, 15}
= {6, 7, 2, 3, 5, 8, 15, 16, 17, 11, 12, 14}
= G

★ D-trans [F, a] = ε-closure {9, 4, 13} = F
= {~~9, 13, 16, 17, 11, 12, 14~~} = H

D-trans [F, b] = ε-closure {6, 10, 15}
= {2, 3, 5, 6, 7, 8, 10, 11, 12, 14, 15, 16, 17}
= H

★ D-trans [G, a] = ε-closure {4, 9, 13} = F

D-trans [G, b] = ε-closure {6, 15} = G

★ D-trans [H, a] = ε-closure {4, 9, 13} = F

D-trans [H, b] = ε-closure {6, 15} = G

Scanned with CamScanner

states	a	b
→ A	B	∅
B	C	D
C	C	E
D	C	D
E	F	G
F	F	H
G	F	G
H	F	G

Minimised DFA

✡ π_0 = {A, B, C, D, E} {F, G, H}
q_1 q_2

A: a / q_1, b / ∅
B: a / q_1, b / q_1
C: a / q_1, b / q_1
D: a / q_1, b / q_1

E: a / q_2, b / q_2
F: a / q_2, b / q_2
G: a / q_2, b / q_2
H: a / q_2, b / q_2

✡ π_1 = {A} {B, C, D} {E} {F, G, H}
q_1 q_2 q_3 q_4

B: a / q_2, b / q_2
C: a / q_2, b / q_3
D: a / q_2, b / q_2

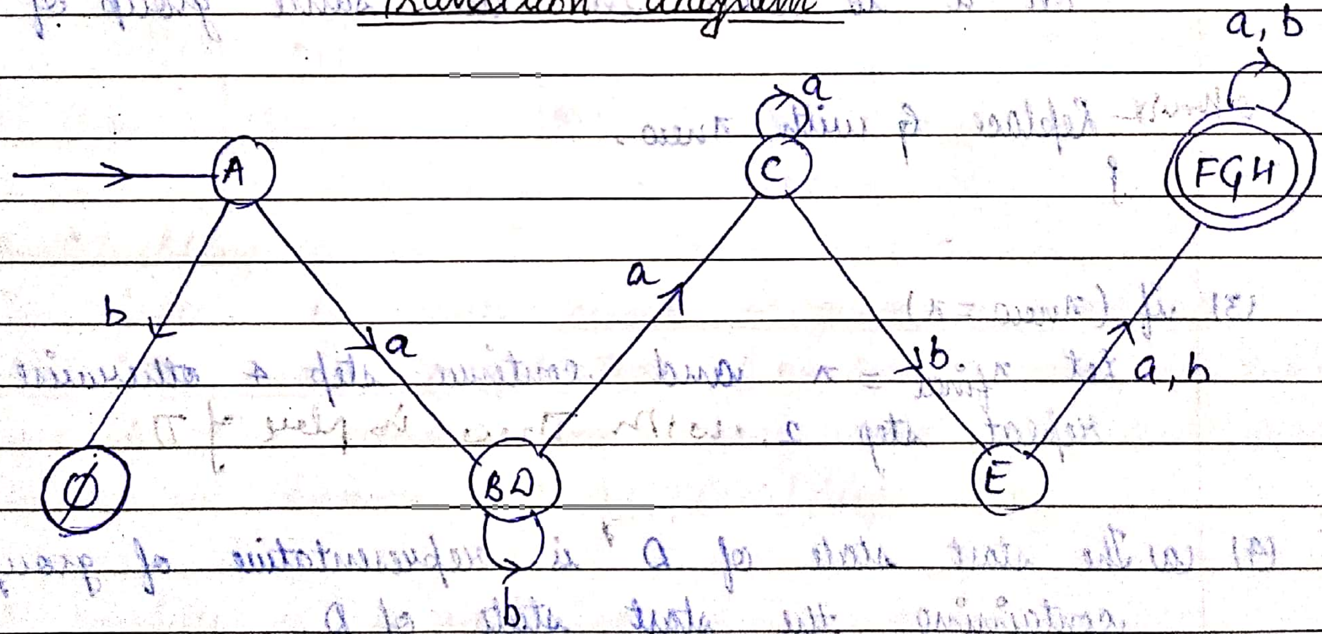
✡ π_2 = {A} {B, D} {C} {E} {F, G, H}

Final stage.

Transition Table

states	a	b
→ A	BD	ϕ
BD	C	BD
C	C	E
E	FGH	FGH
FGH	FGH	FGH

Transition diagram



Algorithm for Minimisation of DFA

(1) Start with the initial partition of π with 2 groups F and S-F. F = final state, S = Non-final state — i.e. Accepting & Non-Accepting state of D.

(2) Apply the following procedure to construct a new partition π_{new}.

Initially let $\pi_{new} = \pi$

for (each group G of π)

{

Partition G into subgroup such that if two state s and t are in same subgroup iff for all I/P symbol 'a' - state s and t have transition on 'a' to state in the same group of π.

Otherwise Replace G with π_{new}.

}

(3) if ($\pi_{new} = \pi$)

Let $\pi_{final} = \pi$ and continue step 4 otherwise repeat step 2 with π_{new} in place of π

(4) (a) The start state of D' is representative of group containing the start state of D.

(b) The accepting state of D' are the representative of those group that contain an accepting state of D.