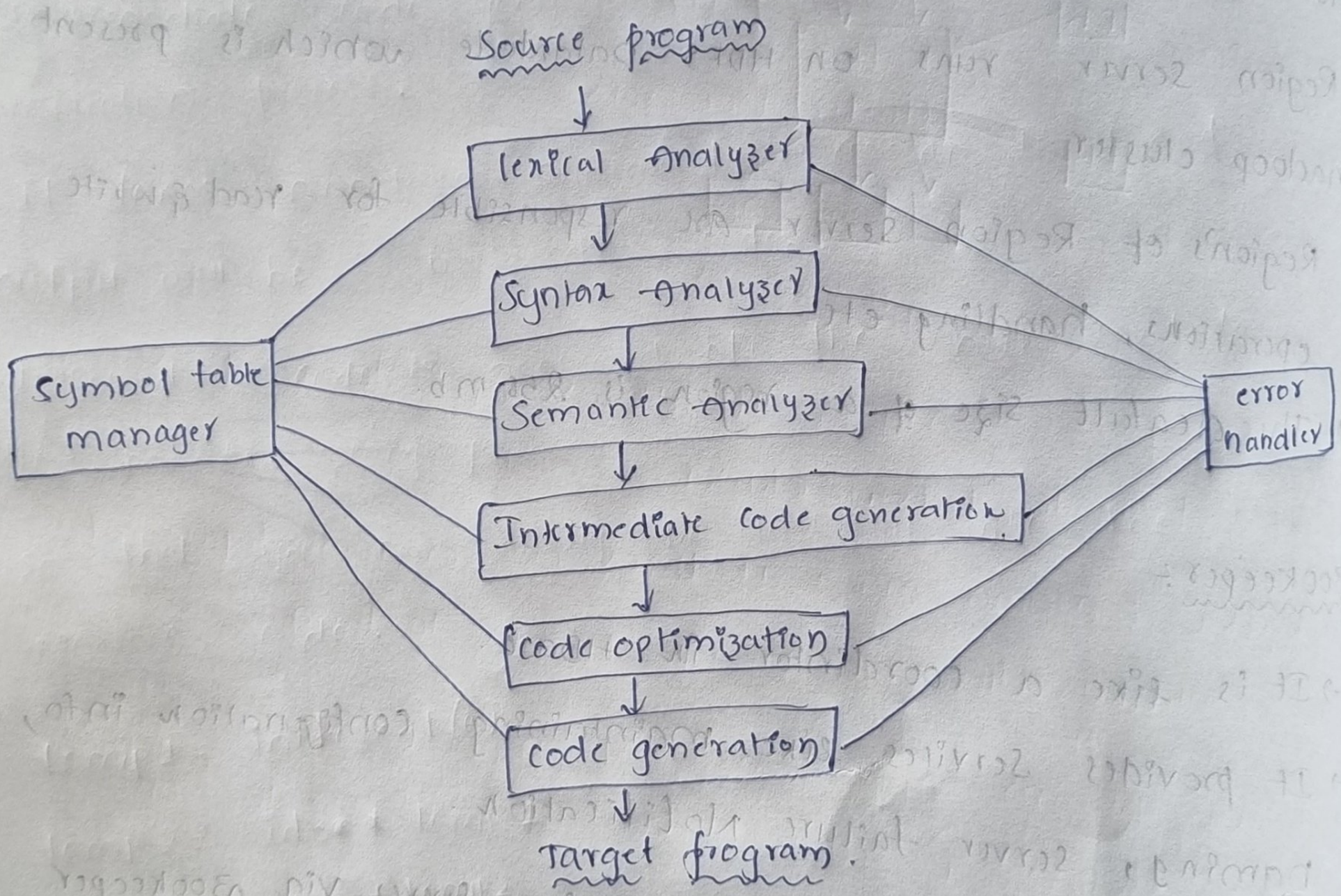
Phases of compiler

→ The compilation process contain the sequence of various phases

→ Each phase takes source program in one representation and produces output in another representation

→ Each phase takes input from its previous stage.

⇒ The Diagramatical representation be like :-



lexical Analysis :

→ It is the first phase of compilation process.

→ It takes source code as input & it reads the source program one character at a time and converts it into meaning-ful lexemes

→ The lexeme's are represented as tokens.

ex:- position := initial + rate * 60

⇒ Position = id

= ⇒ operator

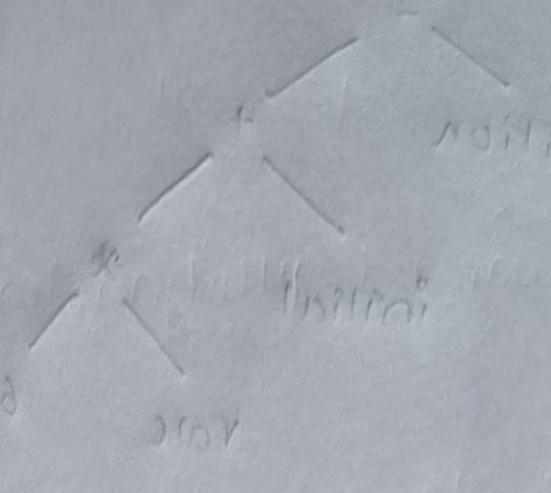
initial = id_2

+ ⇒ operator

rate = id_3

* ⇒ operator

60 = Number



⇒ output = $id_1 := id_2 + id_3 * 60$

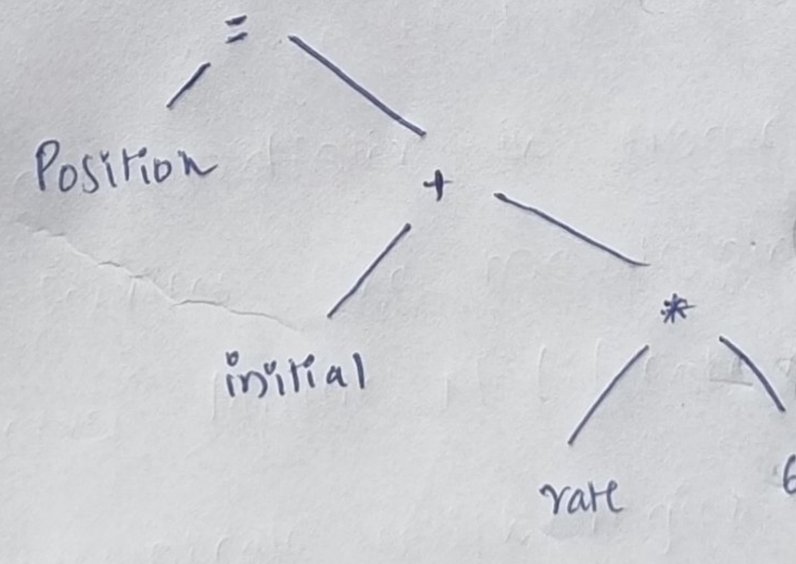
⇒ Syntax Analysis :-

→ It is the 2nd phase of compilation process.

→ It takes tokens as input and generates parse tree as output

→ In syntax Analysis the parser checks the expression made by token is syntactially correct or not.

ex:- Position = initial + rate * 60



⇒ Semantic Analysis :-

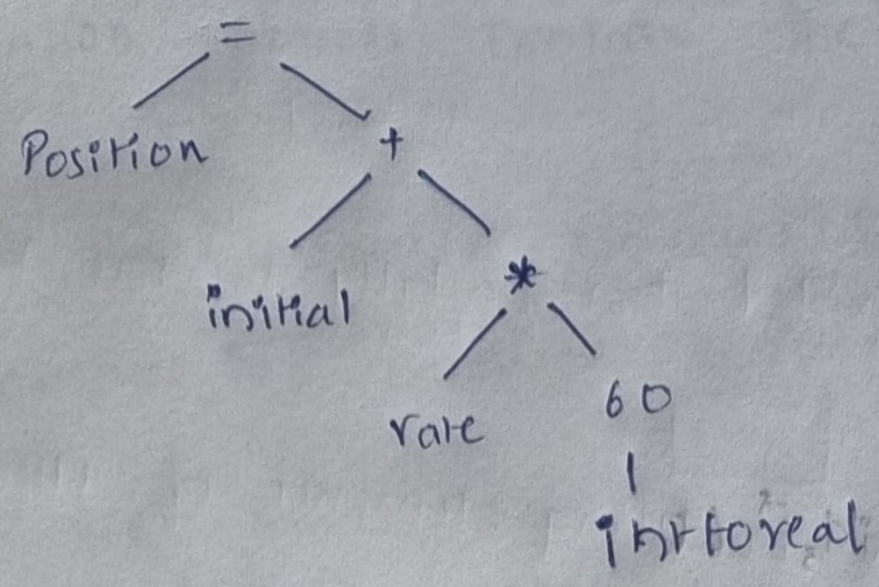
→ Semantic analysis is the 3rd phase of compilation process

→ It takes parse tree as input and generates annotated tree as output

→ It check weather the parse tree follows the rule of language

It keeps track of identifiers, their types & expression.

Semantic analysis :-



⇒ Intermediate code generation :-

→ Intermediate code generation is the 4th phase of compilation process.

→ It takes Annotated syntax tree as an Input and output will be Intermediate code.

→ Intermediate code is generated b/w the high-level language & the machine language

→ so that you can easily translate it into the target machine code.

temp 1 = inttoreal (60).

temp 2 = id 3 * temp 1

temp 3 = id 2 + temp 2

id 1 = temp 3.

Code optimization :-

It is the 5th phase of compilation process.

→ It takes Intermediate code as input and produces optimized code.

→ it optimize the Intermediate code and so that the output of the program could run faster & take less space.

ex:- temp1 = rate * inttoreal(60)

$id_1 = id_2 + temp1$

Target code generator:-

It is the last state of the compilation process.

It takes optimized code as input & convert it into the Assembly language.

⇒ Optimized intermediate code is translated into sequence of machine learning Instruction. ∈

```
MOVE  id3, R2
MUL   #60.0, R2
MOV   id2, R1
ADD   R1, R2
MOV   R1, id1
```

Symbol table manager:- It is an important data structure created & maintained by compiler in order to store information about the occurances of variable entities.

Error handler:- The task of the error handling process are to detect each error, report it to the user and then make some recovery. during the time of this program should not be slow.

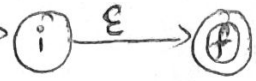
Function in error handlers.

→ error detection
→ error Report
→ error recovery

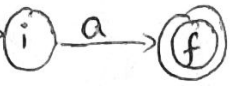
Regular Expression to NFA (Thompson Construction method)

We have 6 Rules to Construct NFA.

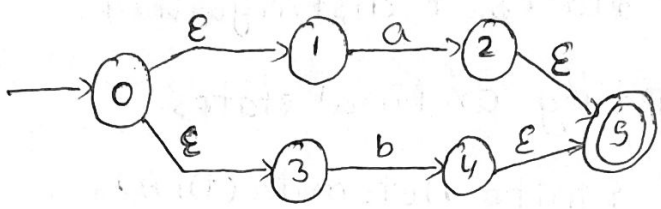
ε -



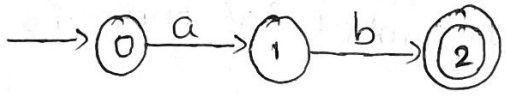
a -



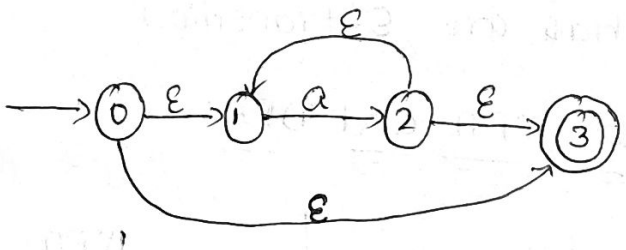
a/b -



a.b -



a* -

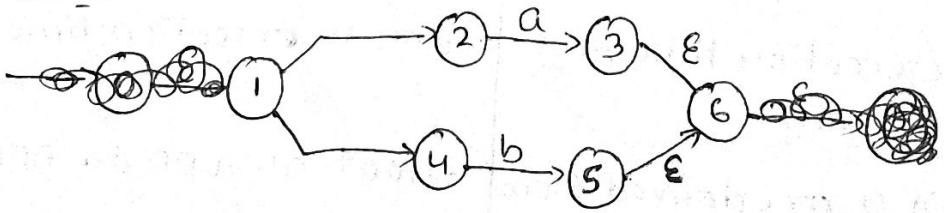


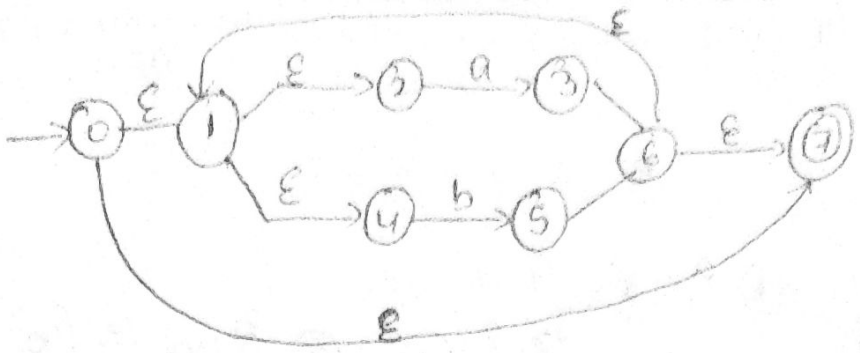
Ex:-

① Convert the following RE to NFA-ε

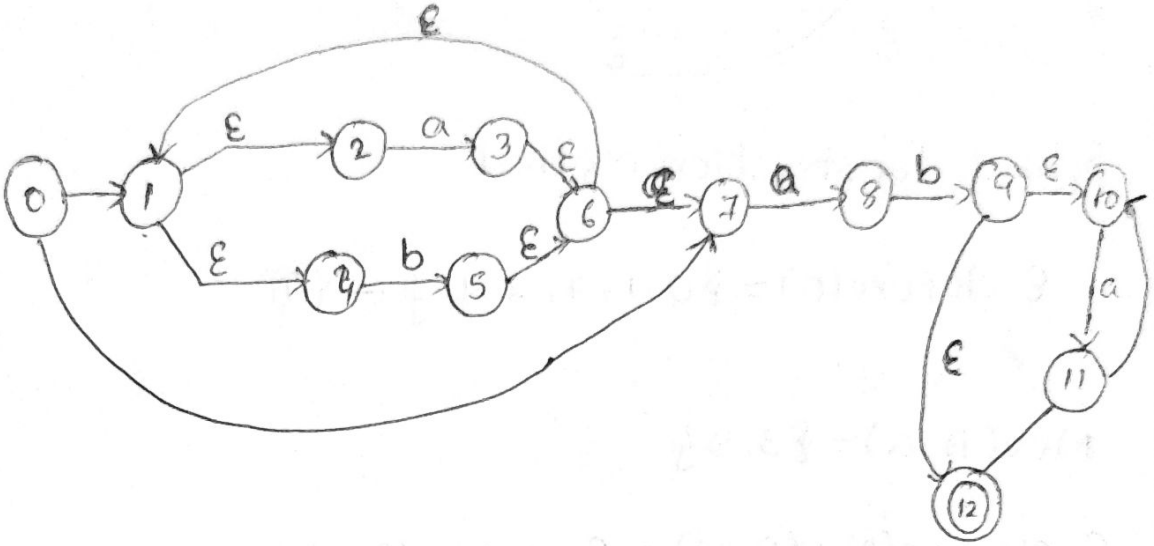
, (a/b)*

Step-1

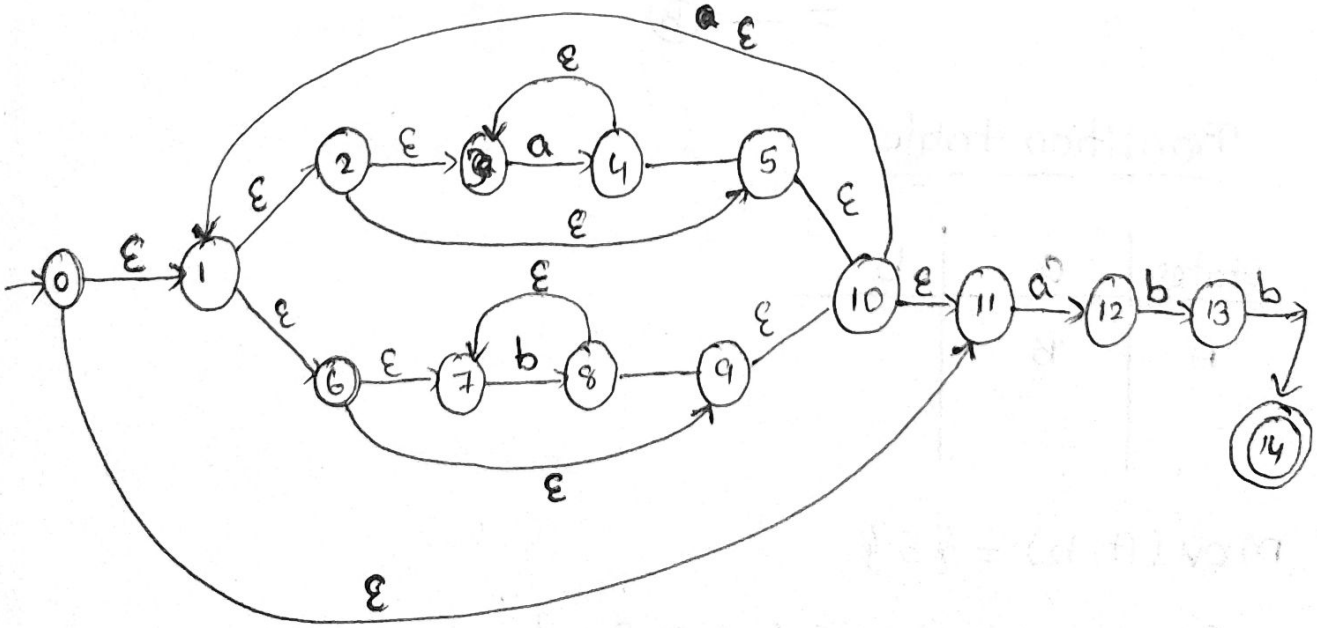




(2) $(a/b)^*abb(a)^*$

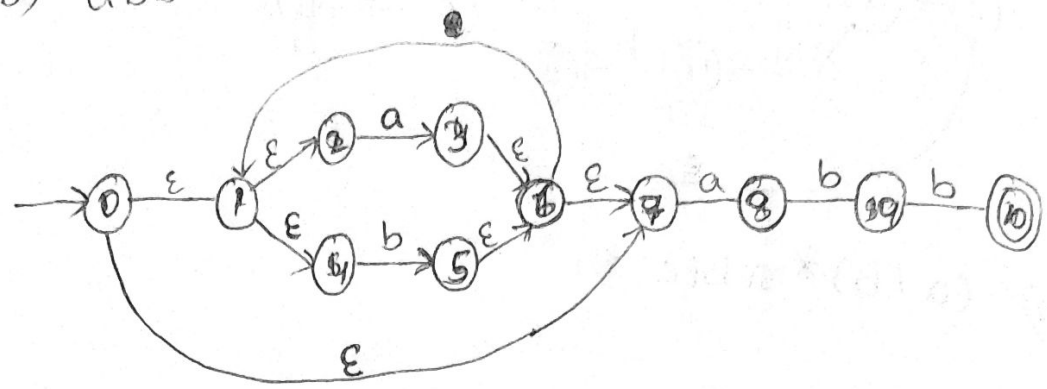


(3) $(a^*/b^*)^*$ a b b.



① Convert the following regular expression to a minimized DFA.

$(a/b)^* abb$



Subset Construction method.

ε-closure(0) = {0, 1, 7, 2, 4} → (A)

Mov(A, a) = {3, 8}

ε-closure(Mov(A, a)) = ε-closure(3, 8)

= {3, 6, 1, 7, 2, 4, 8}

= {1, 2, 3, 4, 6, 7, 8}

= → (B)

Transition table:

States	a	b
A	B	

Mov(A, b) = {5}

ε-closure(5) = {5, 6, 1, 7, 2, 4}

= {1, 2, 4, 5, 6, 7} ⇒ → (C)

States	a	b
A	B	C
B		
C		

$MOV(B, a) = \{3, 8\}$

$= \{3, 6, 1, 7, 2, 4, 8\}$

$= \{1, 2, 3, 4, 6, 7, 8\} \rightarrow (B)$

States	a	b
A	B	C
B	B	

$MOV(B, b) = \{5, 9\}$

$\varepsilon\text{-closure}(MOV(B,b)) = \{5, 6, 1, 7, 2, 4, 9\} \rightarrow (D)$

$= \{1, 2, 4, 5, 6, 7, 9\} \rightarrow D$

	a	b
A	B	C
B	B	D
C		

$MOV(C, a) = \{3, 8\}$

$= \rightarrow (B)$

	a	b
A	B	C
B	B	D
C	B	C
D	B	

$MOV(C, b) = \{5\}$

$\varepsilon\text{-closure}(MOV(C,b)) = \{5, 6, 1, 2, 4, 7\}$

$= \{1, 2, 4, 5, 6, 7\}$

$= \{\} \rightarrow (C)$

$MOV(D, a) = \{3, 8\}$

$= \rightarrow (B)$

$MOV(D, b) = \{5, 10\}$

$\varepsilon\text{-closure}(MOV(D,b)) = \{6, 1, 7, 2, 4, 10\}$

$= \{1, 2, 4, 6, 7, 10\} \rightarrow (E)$

	a	b
A	B	C
B	B	D
C	B	C
D	B	E

Mov(E, a) = {3, 8}

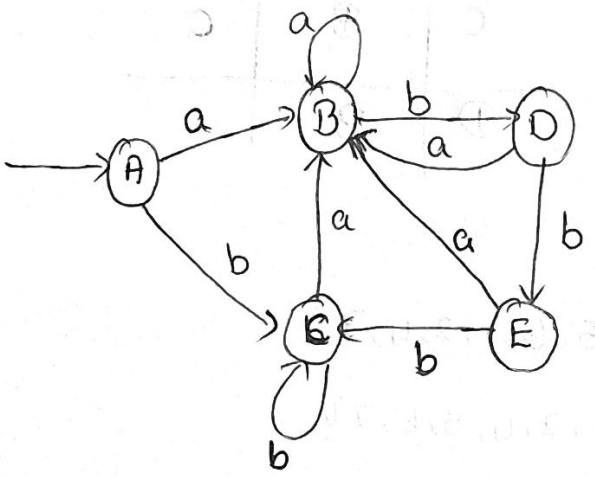
= → (B)

	a	b
A	B	C
B	B	D
C	B	C
D	B	E
E	B	

Mov(E, b) = {5}

= → (C)

	a	b
A	B	C
B	B	D
C	B	C
D	B	E
E	B	C



Minimization of DFA

Group the final and non-final states separately.

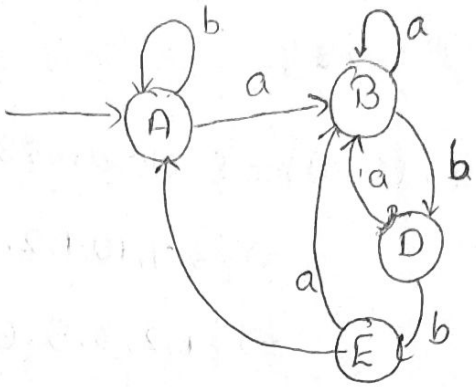
(ABCD) (E)

(ABC) (D) (E)

(AC) (B)

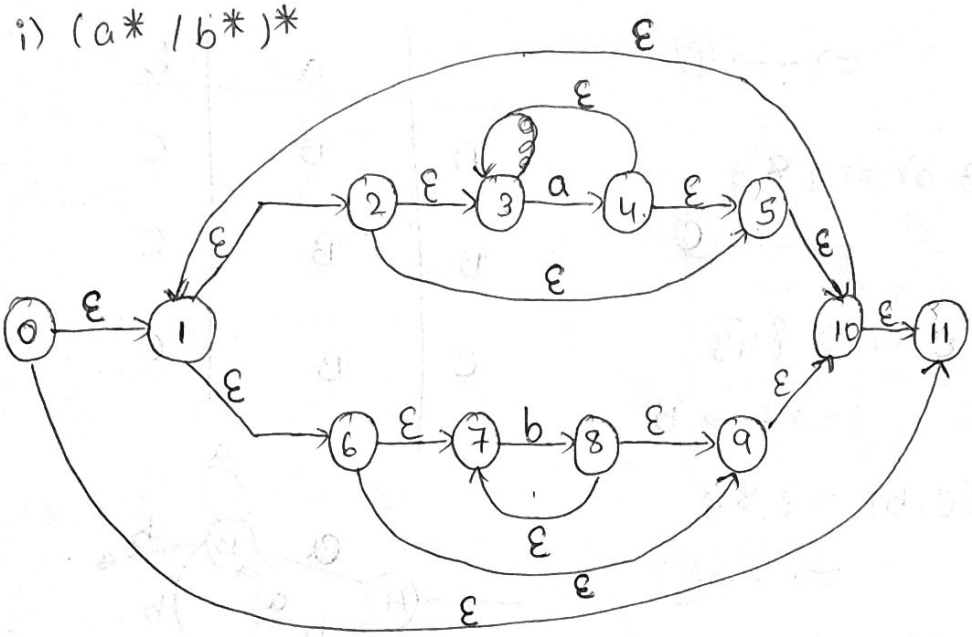
States	a	b
A	B	∉ A,
B	B	D
~~E~~	~~B~~	~~E~~
D	B	E
E	B	∉ A.

	a	b
A	B	A
B	B	D
D	B	E
E	B	A



2. Construct minimized DFA's for the following regular expression and show that they are equivalent.

i) $(a^* | b^*)^*$



Subset Construction method.

ε-closure(0) = {0, 1, 11, 2, 6, 3, 7, 5, 9, 10}

= {0, 1, 2, 3, 5, 6, 7, 9, 10, 11}

→ (A)

mov(A, a) = {4}

ε-closure(mov(A, a)) = ε-closure{4}

= ~~{4, 5, 3, 10, 1, 11, 2, 6, 3, 7, 5, 9}~~

⇒ {4, 5, 3, 10, 1, 11, 2, 6, 3, 9, 7}

⇒ {1, 2, 3, 4, 5, 6, 7, 9, 10, 11}

→ (B)

~~ε-closure~~ {Mov(A,b)

$\Rightarrow$ {8}

ε-closure(Mov(A,b)) = ε-closure{8}.

$\Rightarrow$ {8,9,10,1,2,11,6,3,7,5}

$\Rightarrow$ {1,2,3,5,6,7,8,9,10,11}

$\Rightarrow \rightarrow$ (C)

mov(B,a) $\Rightarrow$ {4}

$\Rightarrow \rightarrow$ (B)

mov(B,b) $\Rightarrow$ {8}

$\Rightarrow \rightarrow$ (B)

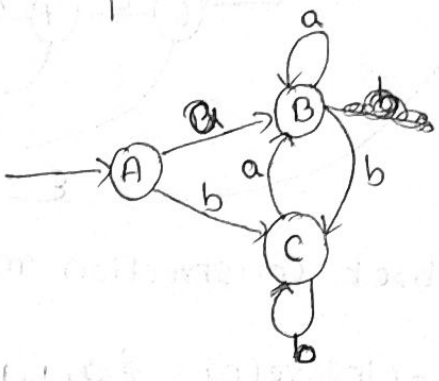
mov(C,a) $\Rightarrow$ {4}

$\Rightarrow \rightarrow$ B

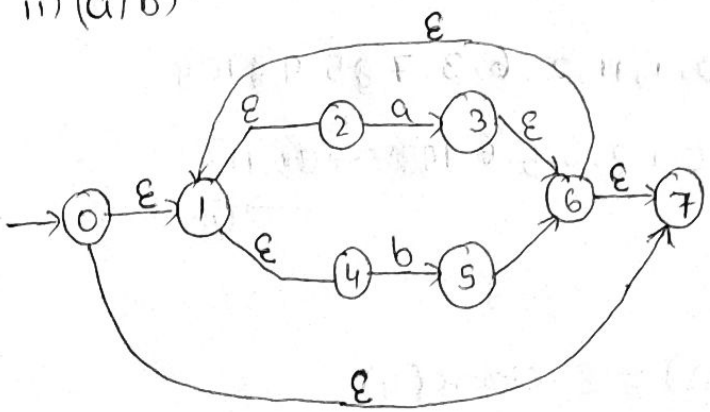
mov(C,b) = {8}

$\Rightarrow \rightarrow$ (C)

	a	b
A	B	C
B	B	C
C	B	C



ii) (a/b)*



NFA.

& DFA

ε-closure {0} $\Rightarrow$ {0,1,7,2,4}

$\Rightarrow$ {0,1,2,4,7}

$\rightarrow$ (A)

$MOV(A,a) = \{3\}$

ε-closure $(MOV(A,a)) \Rightarrow \varepsilon$-closure $\{3\}$

$\Rightarrow \{3,6,1,7,2,4\}$

$\Rightarrow \{1,2,3,4,6,7\}$

$\Rightarrow \rightarrow$ (B).

$MOV(A,b) = \{5\}$.

ε-closure $(MOV(A,b)) \Rightarrow \varepsilon$-closure $\{4\}$

$\Rightarrow \{4,5,6,7,1,2,4\}$

$\Rightarrow \{1,2,4,5,6,7\}$. $\rightarrow$ (C)

$MOV(B,a) = \{3\}$

$\Rightarrow \rightarrow$ (B)

$MOV(B,b) = \{5\}$

$\Rightarrow \rightarrow$ (C)

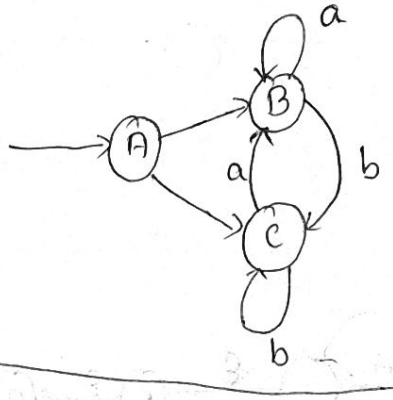
$MOV(C,a) = \{3\}$

$\Rightarrow \rightarrow$ (B)

$MOV(C,b) = \{5\}$.

$\Rightarrow \rightarrow$ (C)

States	a	b
A	B	C
B	B	C
C	B	C



Minimization of DFA

i) $(a^*/b^*)^*$

States	a	b
A	A	C
B	B	C
C	A	C

ii) $(a/b)^*$