1. myfamily = { "child1" : { "name" : "Emil",
                            "year" : 2004 },
               "child2" : { "name" : "Tobias",
                            "year" : 2011 }
             }

print (myfamily ["child 2"] ["year"])

(a under "family", b under "child 2", info under "year")

ফ্যামিলির অনেক info print করছে -

2. string print (a = """lorem ipsum dolar sit met """)

3. index print - a = " Learn teach! "

- print(a[1]) = e
- print(a[-1]) = ! # শেষ থেকে first index
- print (a[1:8]) = earn_te # 1 index থেকে 7 পর্যন্ত
- print (a[1:8:2]) = er e
- print (a[8:1:-2) = ce re .

4. উল্টা print দেখানো -

a = "Hello, World!"
b = a[::-1] = !dlroW, olleH

5. a = input()
b = a[::-1] #b কে উল্টা print করে .
if a==b:
    print ("Yes")
else: print ("No") .

Python file Open

1. f = open("demofile.txt")
2. 'r' = read file

   'w' = write

   'x' = create - the specified file, returns error if the file not exist

   'a' = append - open a file for appending.

   't' = text - Default value, Text Mode

   'b' = binary - mode.

3. read file - f.= open ("file1.txt", "r")
   print (f.read (5))

4. read line -
   f=open ("file1.txt","r"
   a-f.readline()
   print(a)
   f.close()

5. delete file -
   import os
   os.remove("demofile.txt").

folder delete -
import os
os.rmdir("myfolder")

Class and Objects

1. create a class –

```
class Myclass:
- class <classname>
```

2. Object create –

```
<object-name> = <class_name>(<argument>)
msg1 = Myclass("good")
```

3. --init--() function use –

= init() method, we can implement constructor to initialize the object or to assign values for object properties (name).

```
class Person :                          ——— <class name declare>
  def __init__(self, name, age):        ——— <define init() method for object>
    self.name = name                    ——— <variable declaration key>
    self.age = age                      ———
P1 = Person("Asa", 23)                  ——— <value print>
print(P1.name)
print(P1.age)
```

4. Object methods = can also contain methods. Methods in objects are functions the belong to the object.

```
  def myfunc(self):                     <parameter
    print("Name:" + self.name)

P1 = Person("Asa", 23)
P1.myfunc()
= Name : Asa
```

5. del p1.age <for delete object Properties>

6. pass is a Null statement. nothing happen when pass statement is executed. – class Person :
pass

function - a block of code which only runs when it is called.

01. Creating - a function is defined using 'def' keyword.

```
def my-function():
```

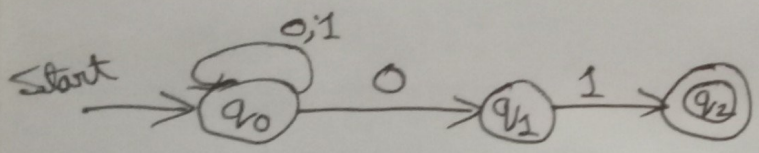
02. calling -

```
def myfunction():
    print (" Hi")
my-function()      < call function >
```

~~Nabil~~ CSE 307

* 2টি Finite automata → Equivalent হবে যদি,
২টি automata র Accepting আর Rejecting string same হবে।

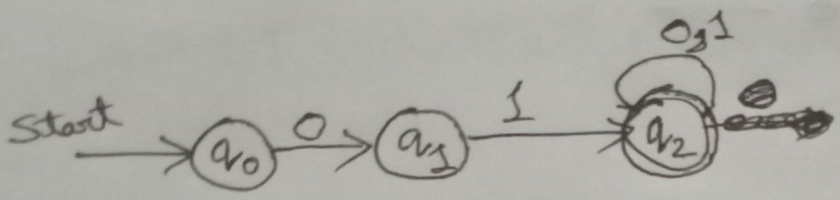
An NFA accepting all strings that end in 01.



Accepting strings → {01}
{001}
{01010011101}
{001010101}

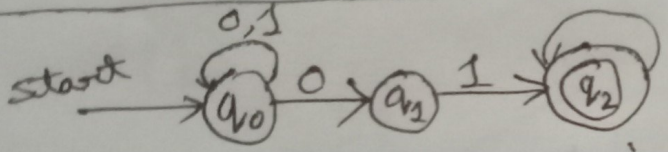
ii)

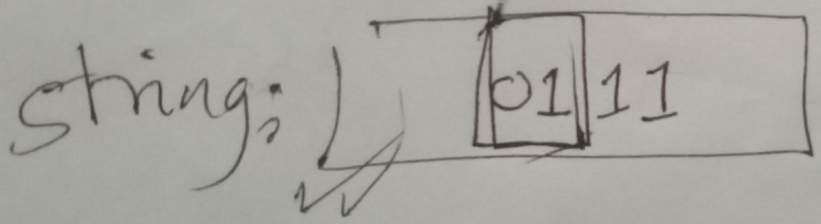
An NFA accepting all starting with 01 has a substring 01.

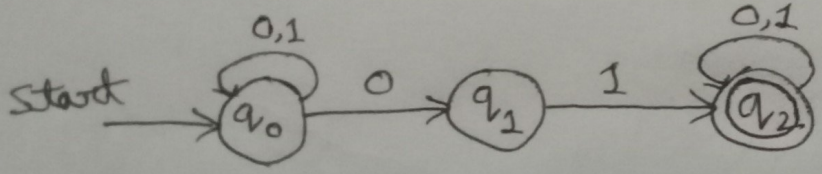


iii)

,, ,, ,,, substring 01







2.9

Transition Table:

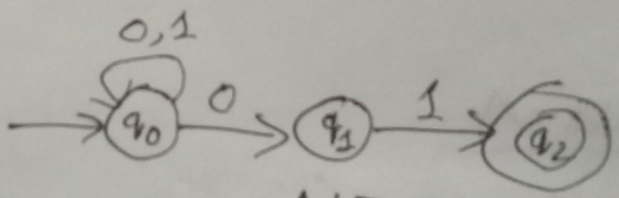
	0	1
→ $\{q_0\}$ A	$\{q_0, q_1\}$ B	$\{q_0\}$ A
$\{q_0, q_1\}$ B	$\{q_0, q_1\}$ B	$\{q_0, q_2\}$ C
* $\{q_0, q_2\}$ C	$\{q_0, q_1\}$ B	$\{q_0\}$ A

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

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

$= \{q_0\} \cup \{q_2\}$

$= \{q_0, q_2\}$



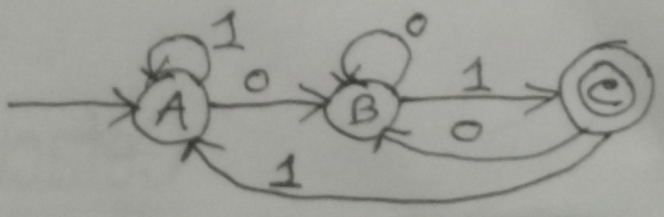
NFA

Equivalent

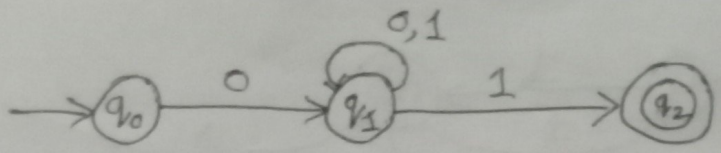
Let, $\{q_0\} = A$

$\{q_0, q_1\} = B$

$\{q_0, q_2\} = C$

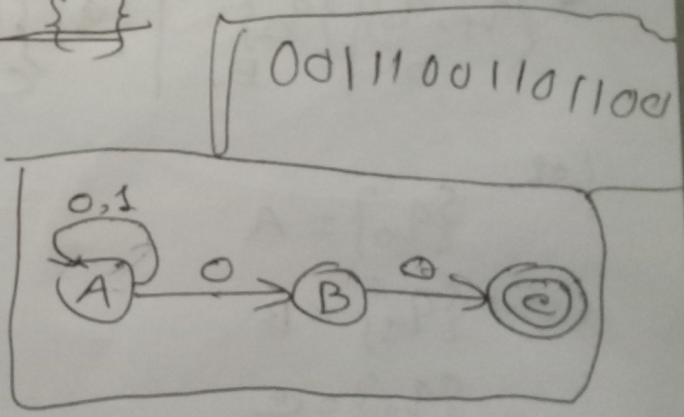


DFA



Transition Table

	0	1
$\rightarrow \{q_0\}_A$	$\{q_1\}_B$	$\{\}$
$\{q_1\}_B$	$\{q_1\}_B$	$\{q_1, q_2\}_C$
$*\{q_1, q_2\}_C$	$\{q_1\}_B$	$\{q_1, q_2\}_C$
~~$*\{q_1, q_2\}_D$~~	$\{\}$	$\{\}$

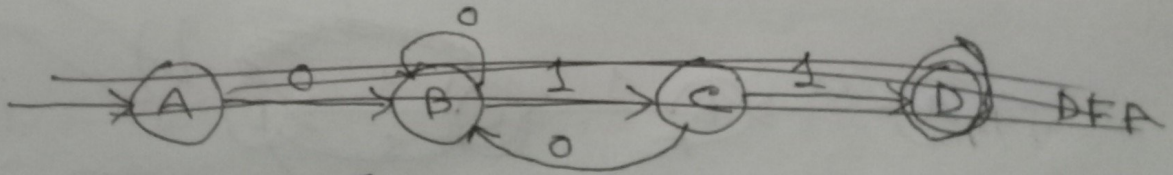


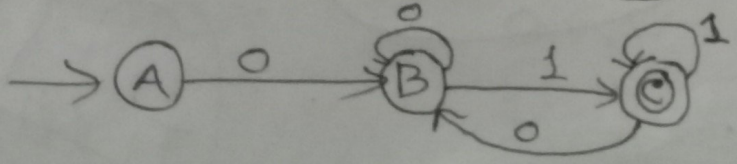
Let, $\{q_0\} = A$

$\{q_1\} = B$

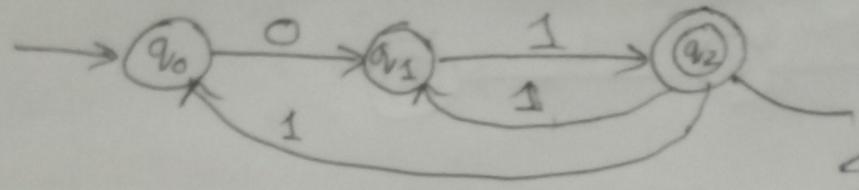
$\{q_1, q_2\} = C$

~~$\{q_1, q_2\} = D$~~





#.



	0	1
→ $\{q_0\}_A$	$\{q_1\}_B$	$\{\}$
$\{q_1\}_B$	$\{\}$	$\{q_2\}_C$
* $\{q_2\}_C$	$\{\}$	$\{q_1, q_0\}_D$
* $\{q_1, q_0\}_D$	$\{q_1\}_B$	$\{q_2\}_C$

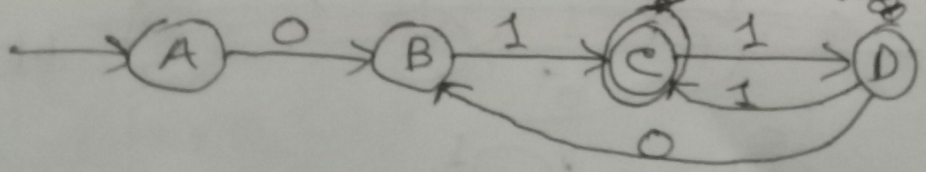
Let,

$\{q_0\} = A$

$\{q_1\} = B$

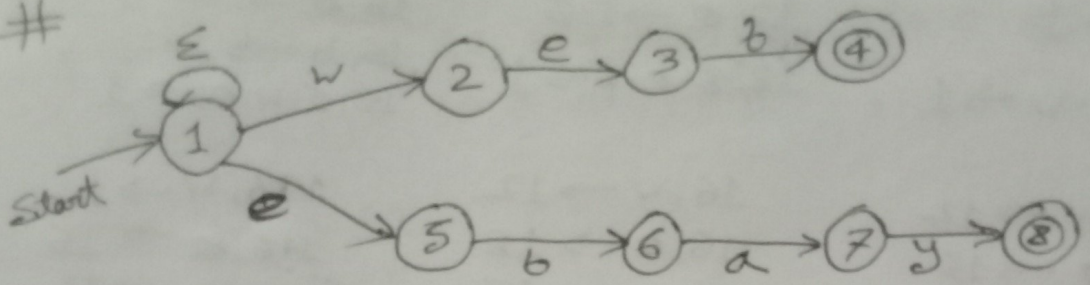
$\{q_2\} = C$

$\{q_1, q_0\} = D$



Final

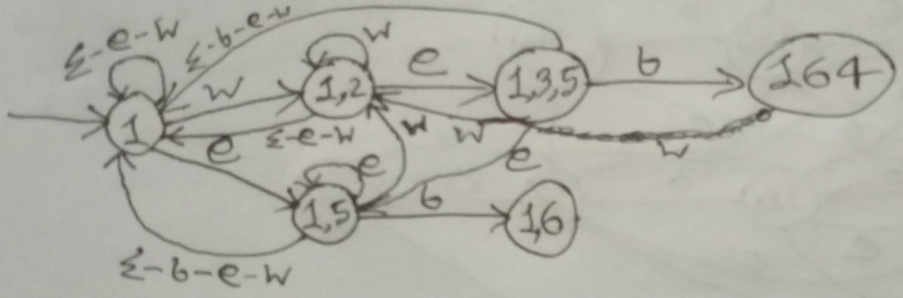
#



DFA

1, w → 1, 2
1, e → 1, 5
1, ε-e-w → 1

12, e → 1, 5, 3
12, w → 1, 2
12, ε-e-w → 1



16, a → 17

1.5, w → 1.2
15, e → 1,5
15, b → 16
15, bε-b-e-w → 1

1 35, b → 164
135, w → 12
135, e → 15
135, ε-b-e-w → 1

1, W → 12

1, e → 15

1, ε-e-W → 1

12, W → 12

12, e → 135

12, ε-e-W → 1

15, W → 12

15, e → 15

15, b → 16

16, ε-W-e-b → 1

135, W → 12

135, e → 15

135, b → 146

135, ε-W-e-b → 1

16, W → 12

16, e → 15

16, a → 17

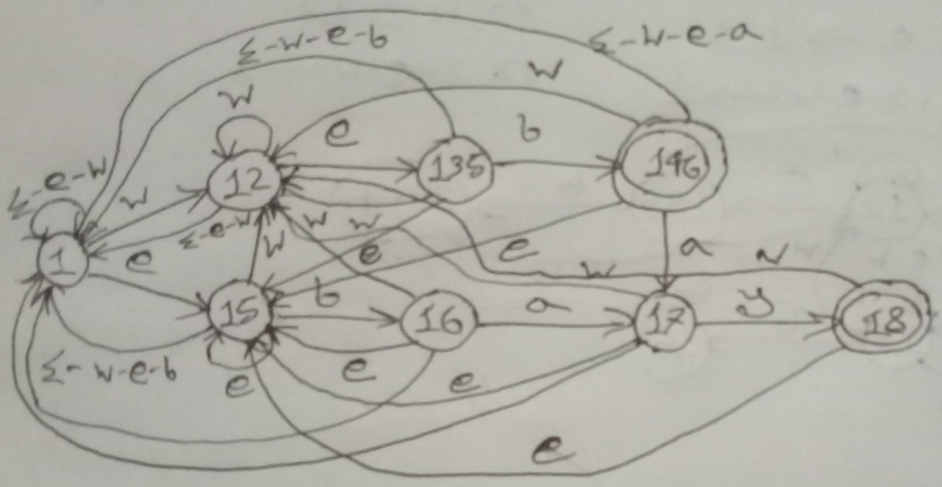
16, ε-W-e-a → 1

146, W → 12

146, e → 15

146, a → 17

146, ε-W-e-a → 1



17, e → 15

17, W → 12

17, y → 18

17, ε-e-W-y → 1

18, e → 15

18, W → 12

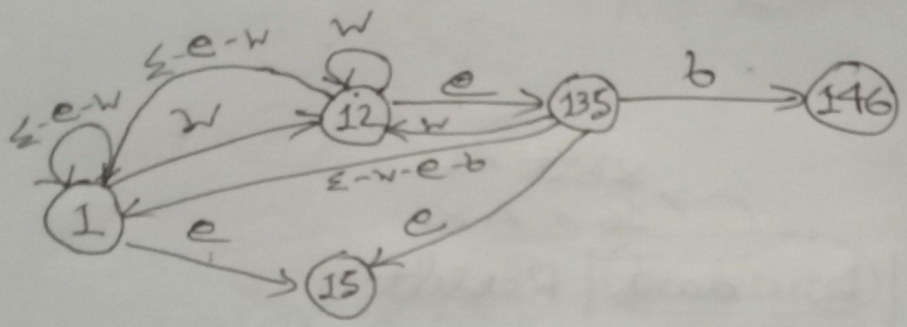
18, ε-e-W → 1

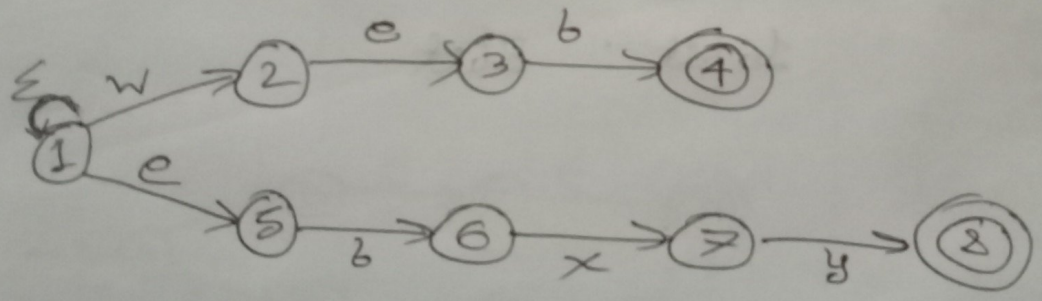
1, w → 12
1, e → 15
1, ε-e-w → 1

12, w → 12
12, e → 135
12, ε-e-w → 1

15, w → 12
15, e → 15

135, w → 12
135, e → 15
135, b → 146
135, ε-w-e-b → 1





w, e b b
↑ ↑ ↑
1 3 5

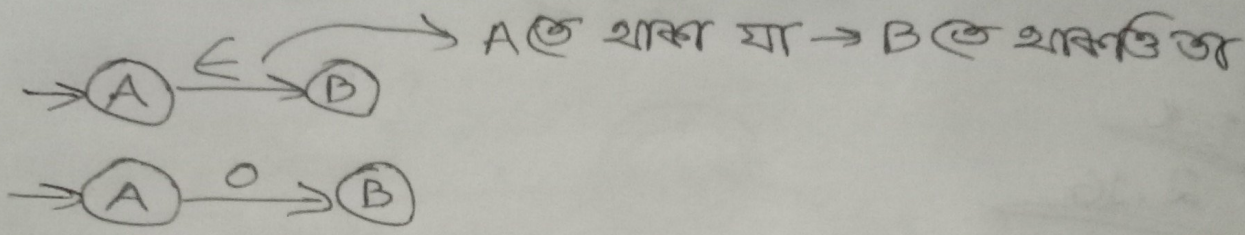
w, e, b

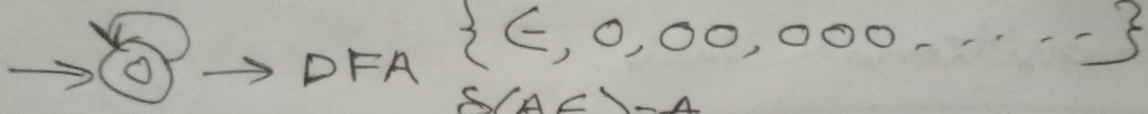
CSE Nadeem Sir

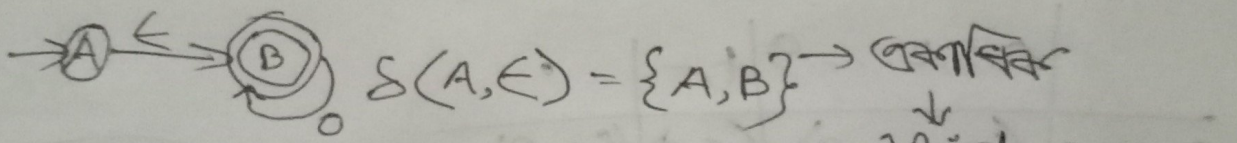
ϵ-NFA → DFA

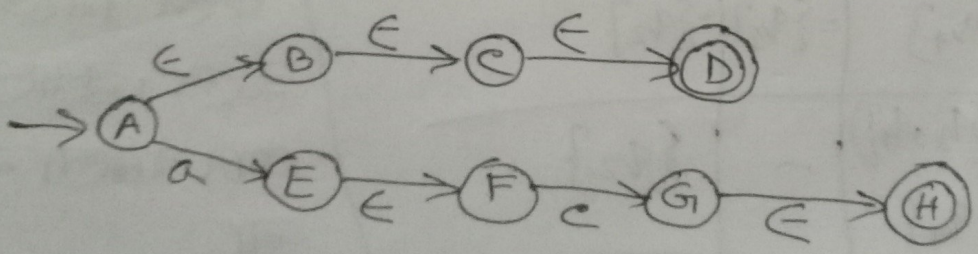
NFA DFA

ϵ-transition



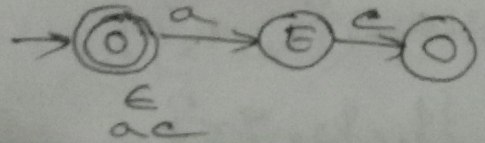






aεcε = ac

↓ concatenated



ECLOSE(A) = {A, B, C, D}

ECLOSE(B) = {B, C, D}

ECLOSE(C) = {C, D}

ECLOSE(D) = {D}

ECLOSE(E) = {E, F}

ECLOSE(F) = {F}

ECLOSE(G) = {G, H}

ECLOSE(H) = {H}

E.X 2.16

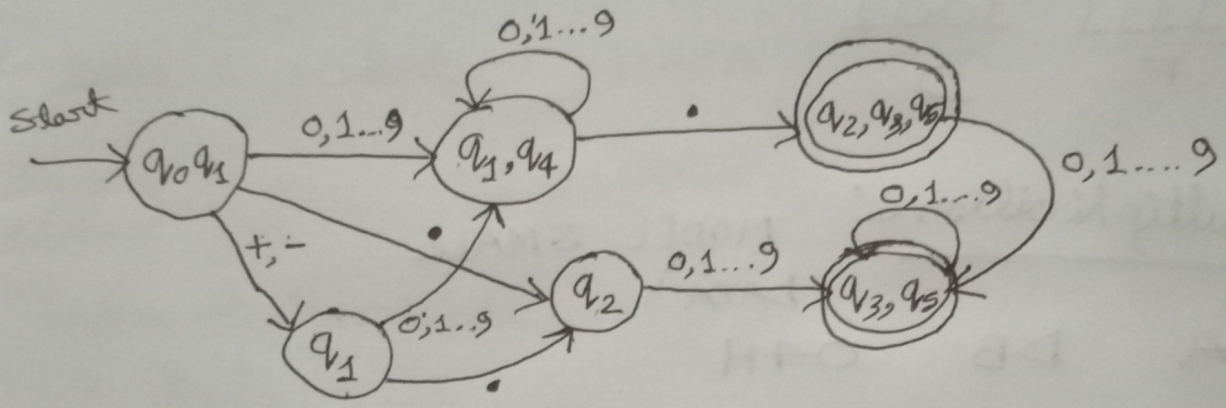
DFA

Transitional Table:

	0, 1, 2 9	+, −	.
→ $\{q_0, q_1\}$	$\{q_1, q_4\}$ → ECLOSE = $\{q_1, q_4\}$	$\{q_1\}$ → ECLOSE = $\{q_1\}$	$\{q_2\}$ → ECLOSE = $\{q_2\}$
$\{q_1\}$	ECLOSE($\{q_1, q_4\}$) = $\{q_1, q_4\}$	−	$\{q_2\}$
$\{q_2\}$	ECLOSE($\{q_3\}$) = $\{q_3, q_5\}$	−	−
$\{q_1, q_4\}$	$\{q_1, q_4\}$	−	ECLOSE($\{q_2, q_3\}$) = $\{q_2, q_3, q_5\}$

সবসময় ECLOSE করে নিতে হবে। বুঝলে আর বসানো লাগবে না।

	0, 1, . . . 9	+, −	.
* $\{q_3, q_5\}$	$\{q_3, q_5\}$	—	—
* $\{q_2, q_3, q_5\}$	$\{q_3, q_5\}$	—	—

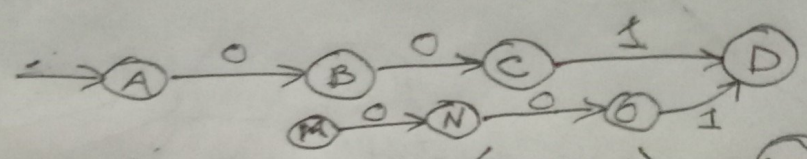


DFA

CSE307

4.4

Equivalent and minimization



$\delta(A, 0) = B$

$\delta(B, 0) = C$

$\delta(C, 1) = D$

$\delta(A, \underbrace{001}_{\text{string}}) = D$

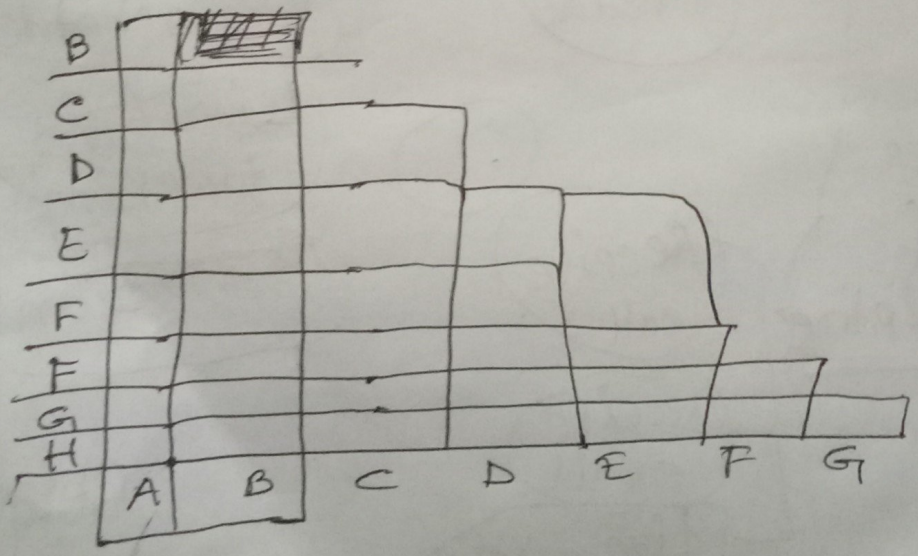
$\delta(M, 001) = D$

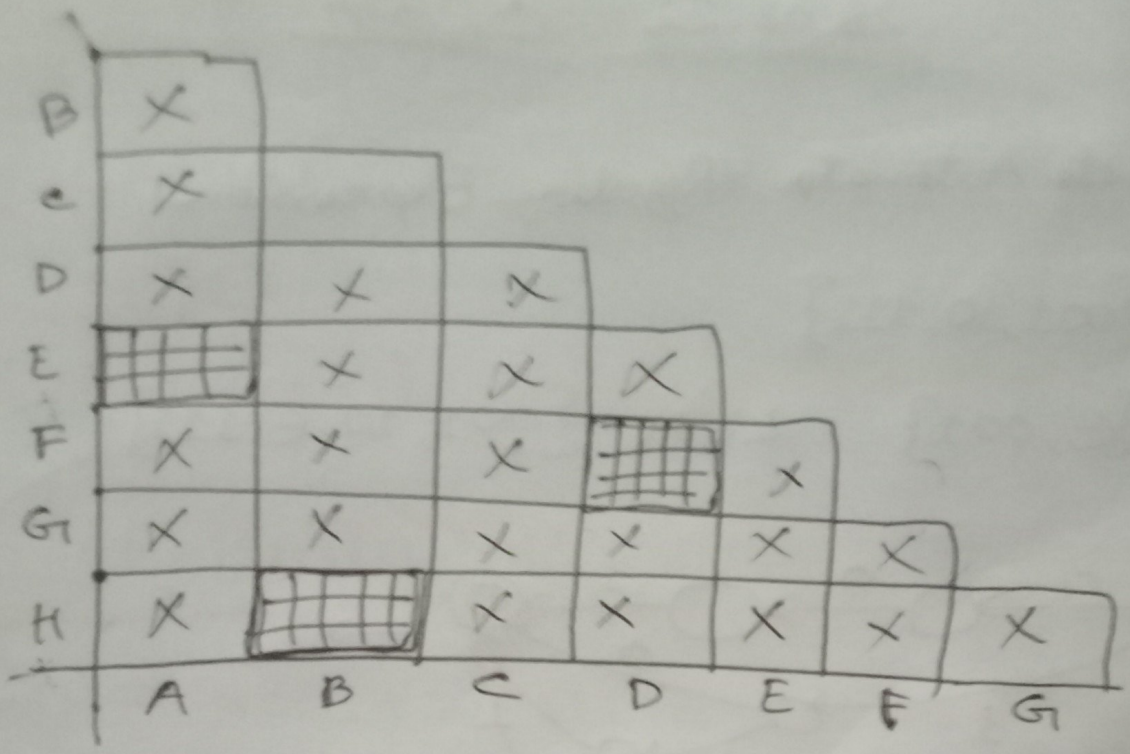
Equivalent হবে A, M

Final State, Final State } → Equivalent

Non " " , Non " " } → "

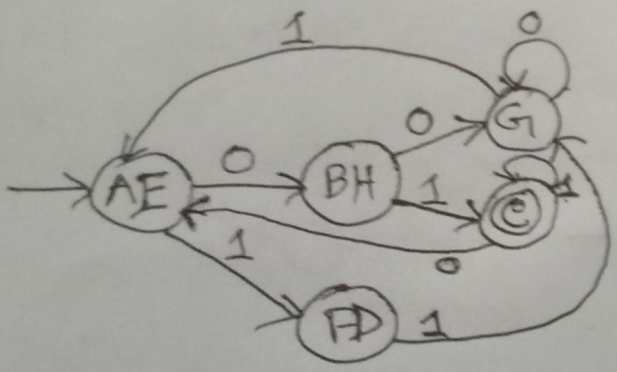
Non " " , Final " } " হবে না





মিনিমাইজড DFA:

A, 0 → B A, 1 → F
E, 0 → H E, 1 → F
B, 0 → G B, 1 → C
H, 0 → G H, 1 → G



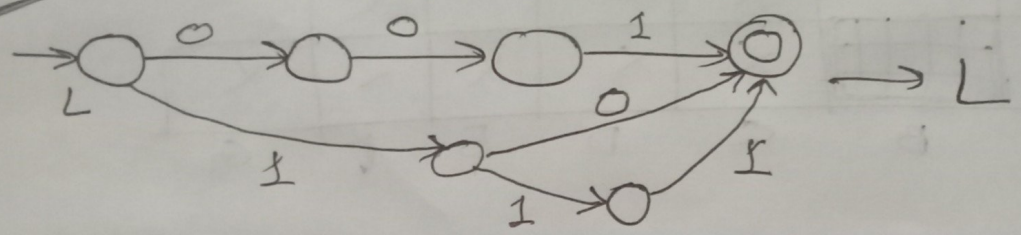
28.08.22 CSE307

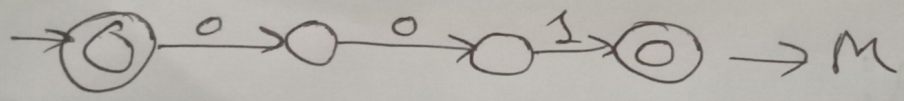
Finite Automata ≡ Regular Expression

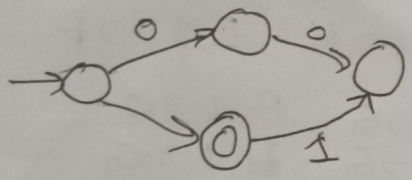
$L = \{001, 10, 111\}$

$M = \{\in, 001\}$ $L \cup M = \{001, 10, \in, 111\}$

Union:







Concatenation

$L.M = LM \neq ML$

001 ∈
001 001
10 ∈
10 001
111 ∈
111 001

CLOSURE

$L = \{0, 11\}$ $\quad L^2 = L.L = \{00, 011, 110, 1111\}$

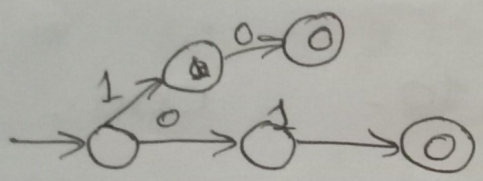
$L^0 = \{\in\}$

$L^1 = L = \{0, 11\}$

L^2

$L^* = L^0 \cup L^1 \cup L^2 \cup L^3$

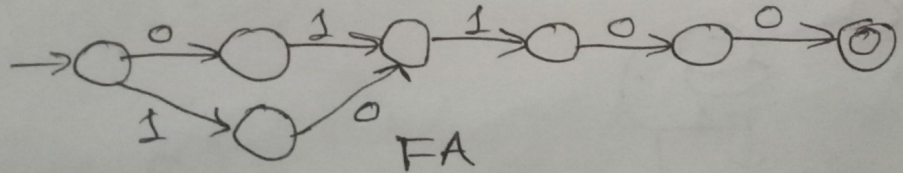
→ RE = 001 + 10 + 111 + ∈



RE = 01 + 10

RE = (01+10)100

$L = \{01100, 10100\}$

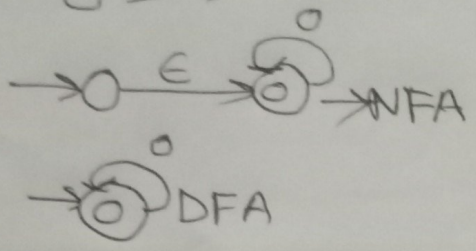


FA

RE = a^*

String = ϵ, a, aa, aaa - - - - -

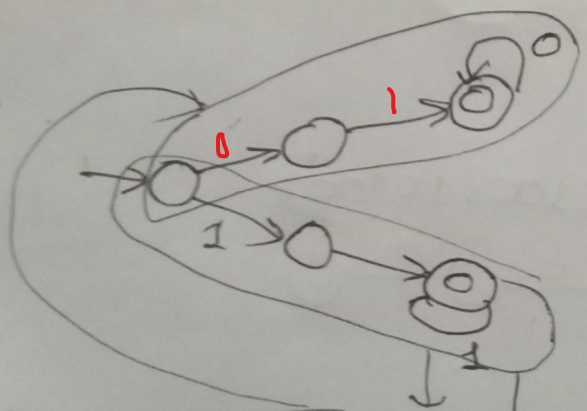
$0^* \rightarrow$ RE



String : $\epsilon, 0, 00, \ldots$

Strings : $\epsilon, 0, 1, 00, 01, 10 \ldots 000 \ldots$

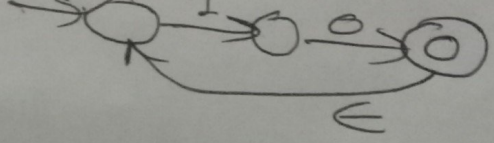
RE = $(0+1)^*$



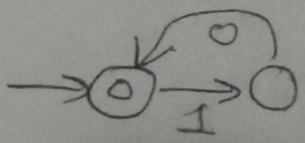
RE = $01\boxed{0^*} + 10\boxed{1^*}$

RE = $(10)^*$

$\boxed{a = 10}$

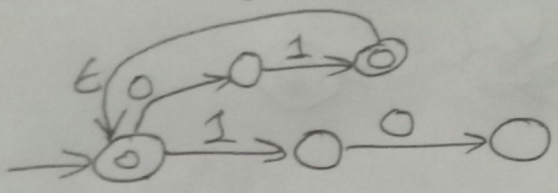


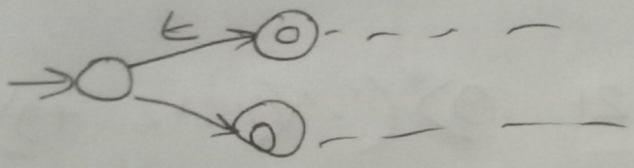
$10\epsilon 10\epsilon 10$



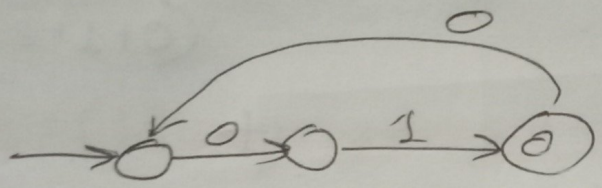
String : $\epsilon, 10, 1010, 101010 \ldots$

RE = (10)* + (01)*





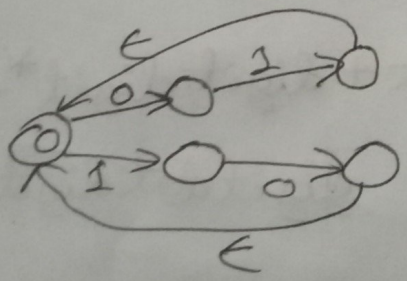
#



String → 01
01001
01001001

RE = 01(001)*

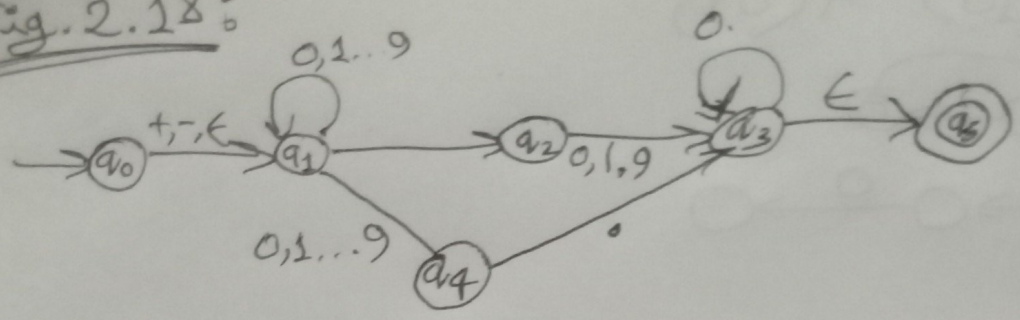
L=(01+10)*



* $a^+ = aa^* = a, aa, aaa$

* $a? = \epsilon + a$

Fig. 2.18:



$$RE = (\in + '+' + '-')(0+1+2+\ldots 9)^*('.'(0+1+2\ldots+9)(0+1+2\ldots+9)^*\in + (0+1+\ldots 9)'.'(0+1+\ldots 9)^*\in$$

समाधान

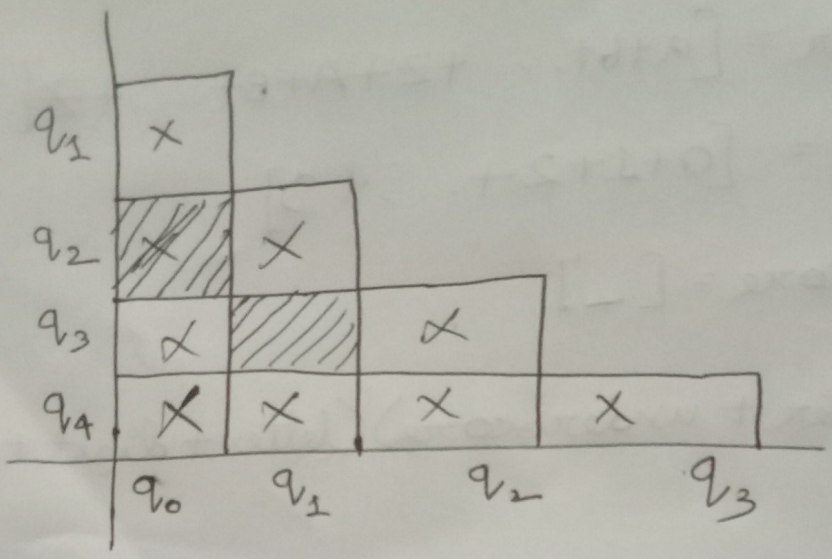
symbol = ('+' + '−')

digit = (0 + 1 + ... 9)

dot = •

$$RE = (\in + symbol)\,digit^*(dot + digit\,digit^* + digit\,dot\,digit^*)$$

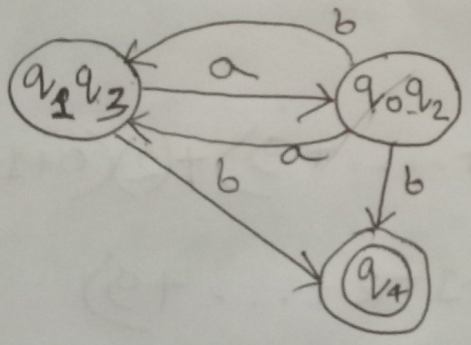
3. ii)



ছোট DFA :

$q_1, a = q_2$ $q_1, b = q_4$

$q_3, a = q_2$ $q_3, b = q_4$



$q_0, a = q_1$ $q_0, b = q_3$

$q_2, a = q_1$ $q_2, b = q_4$

1.b) $\Sigma = \{a, t, i, k\}$