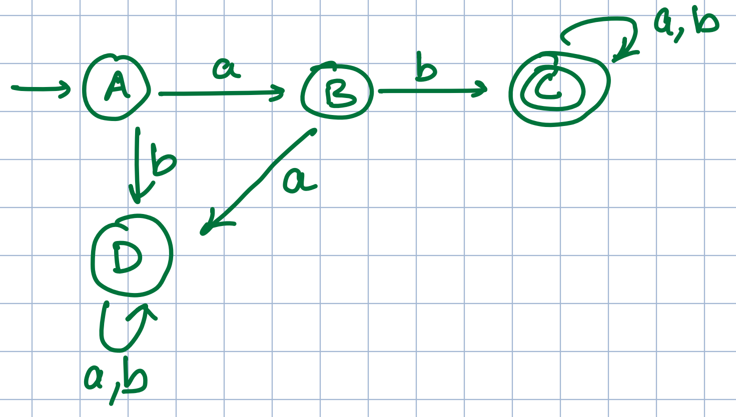
Q: Starts with 'ab'

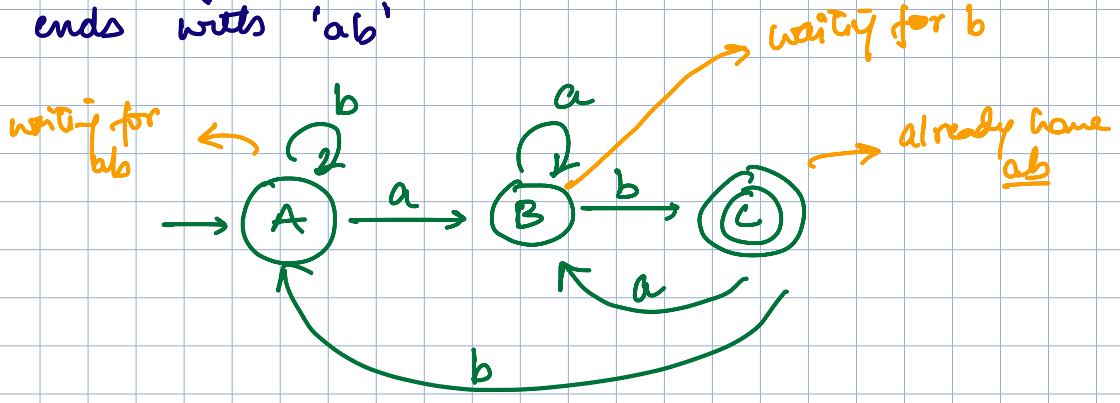


ab✓
aba✓
abb✓
abab✓

aab ✗

ba
bb } ✗

Q: ends with 'ab'



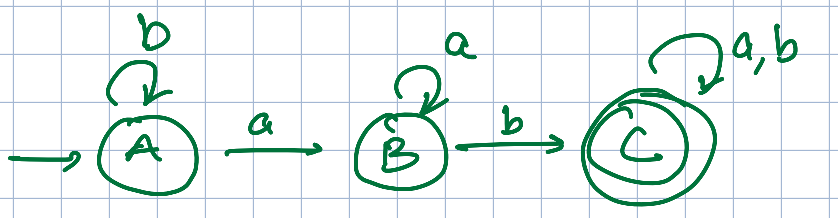
✓ab
aab
baab
aaab

abaaa

abb

$A \xrightarrow{a} B \xrightarrow{b} C \xrightarrow{b} ?$

Q: contains 'ab' ?



baba

Operations

1. Union

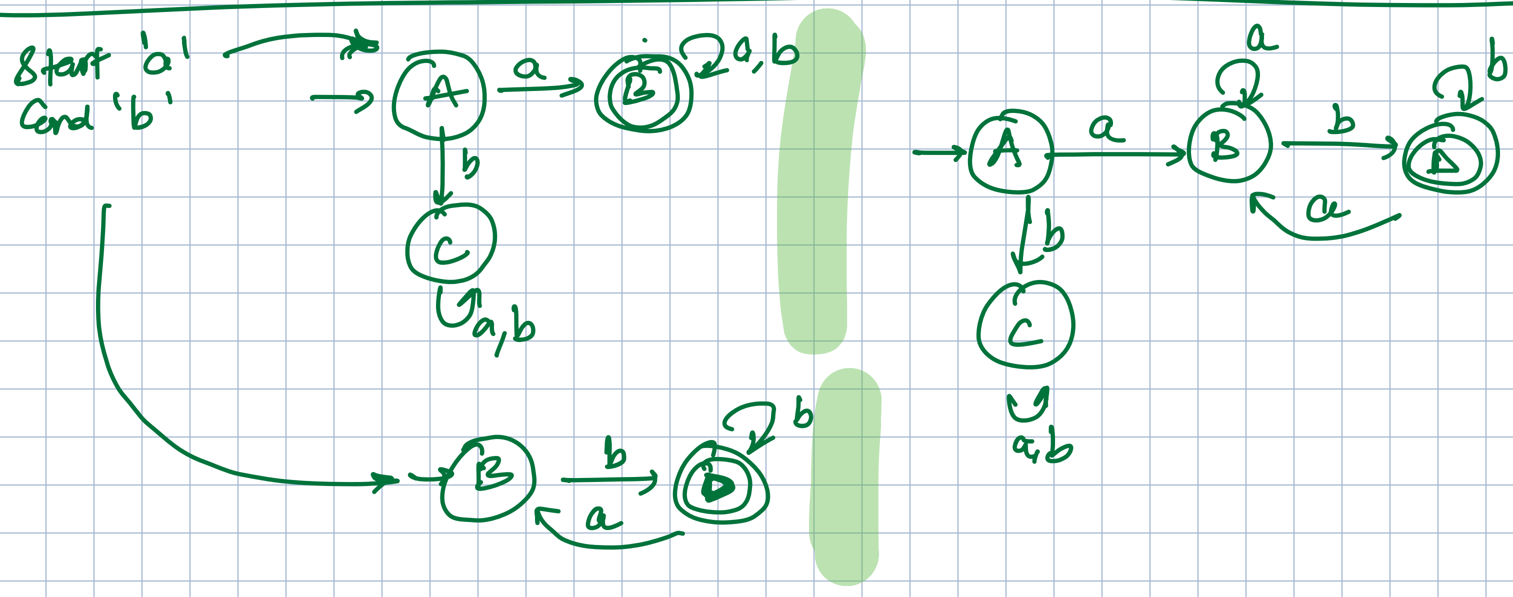
Σ={a,b}

Eg: L: starts & ends with different symbols

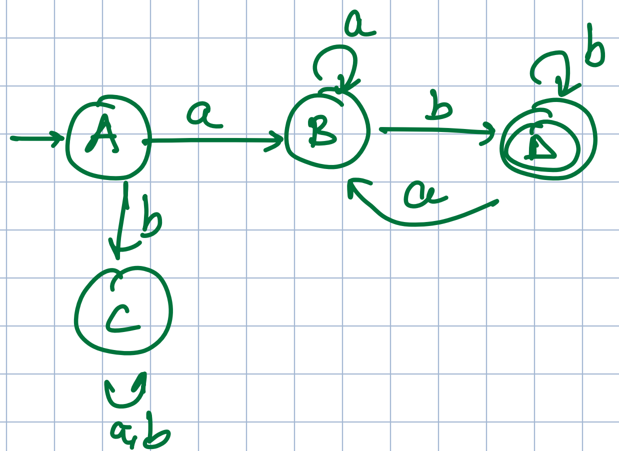
L₁: starts with a & ends with b = {ab, abb, aab, ...}

L₂: starts with b & ends with a = {ba, bba, ...}

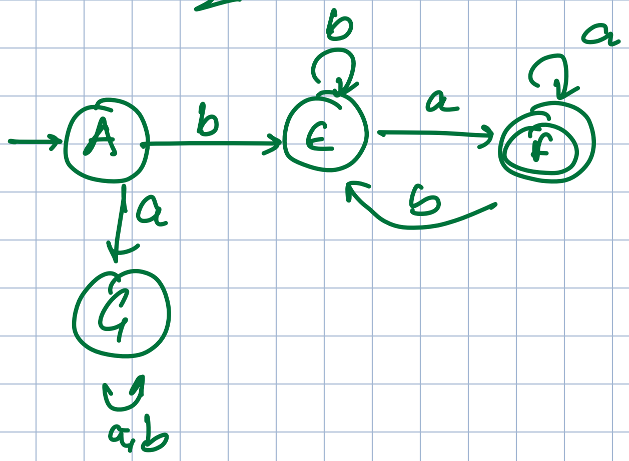
DFA for L₁:

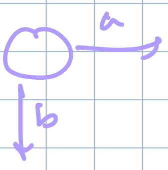


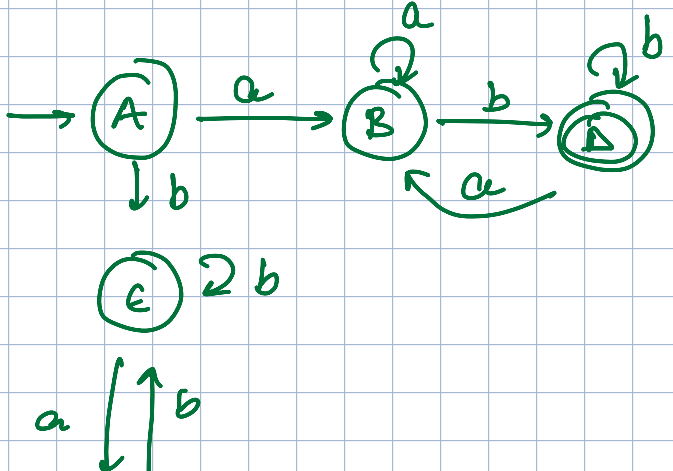
L₁:



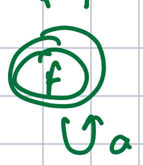
L₂







bab



2. Concatenation

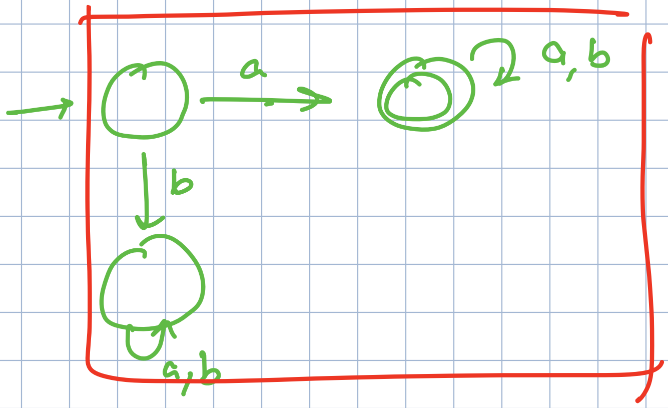
DFA: $\underbrace{\text{starts with 'a'}}_{L_1}$ and $\underbrace{\text{ends with 'b'}}_{L_2}$

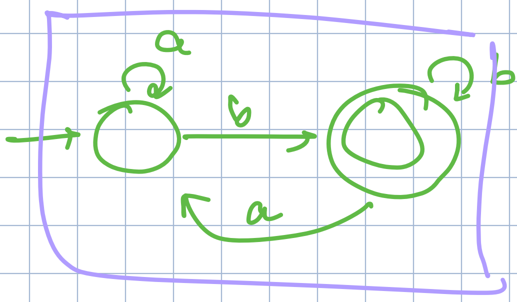
L_1: starts with 'a'

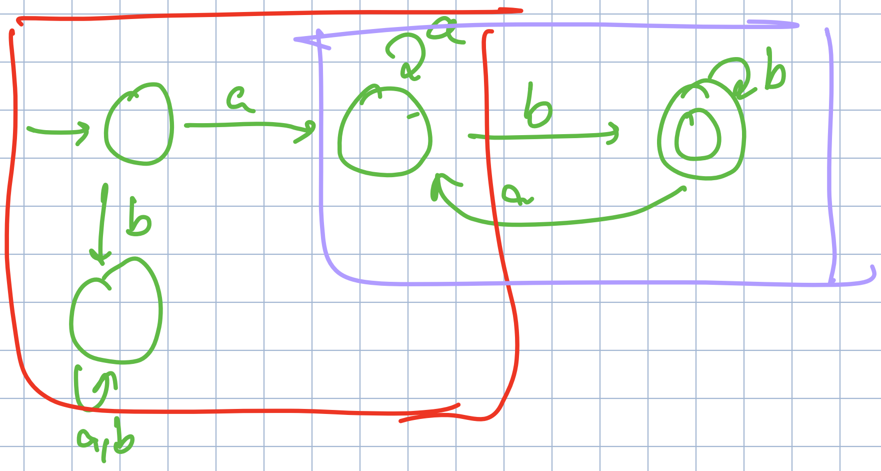
$= \{a, aa, ab, aab \ldots\}$

L_2: ends with b

$= \{b, bb, abb \ldots\}$







$L_1 \longrightarrow D_1$

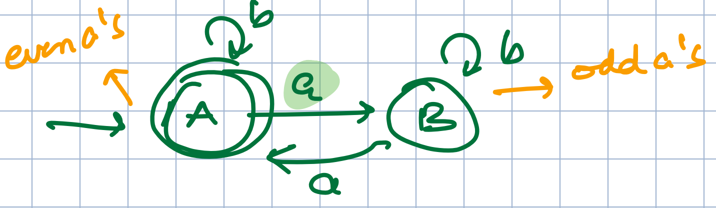
$L_2 \longrightarrow D_2$

$L_1 . L_2 \longrightarrow D_1 . D_2$

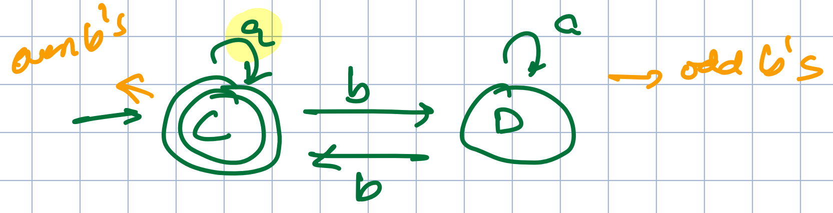
3. Cross Product

Eg: $\underbrace{\text{Even no of a's}}_{L_1}$ and $\underbrace{\text{Even no of b's}}_{L_2}$

DFA for L_1:



DFA for L_2:



$\{A, B\} \times \{C, D\}$

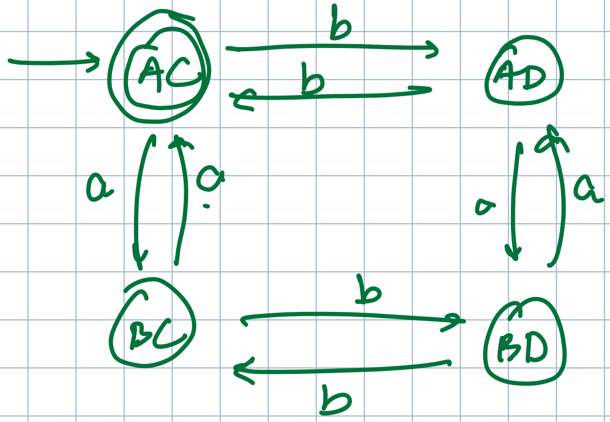
$\{AC, BC, AD, BD\}$

$AC \xrightarrow{a} BC$, $AC \xrightarrow{b} AD$

$BC \xrightarrow{a} AC$, $BC \xrightarrow{b} BD$

$AD \xrightarrow{a} BD$, $AD \xrightarrow{b} AC$

$BD \xrightarrow{a} AD$, $BD \xrightarrow{b} BC$

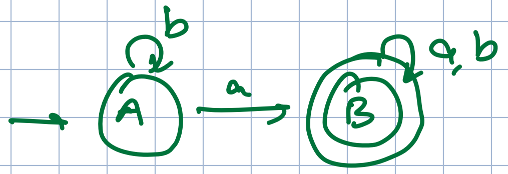


4. Complementation

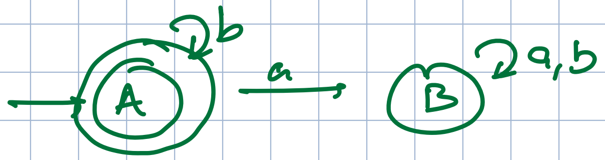
Eg: L_1 = contains 'a' = {a, aa, ba, aab . . . . }

$\overline{L_1}$ = not containing 'a' = { ε, b, bb, ~~bb~~b .... }

DFA for L_1 =



DFA for $\overline{L_1}$ =



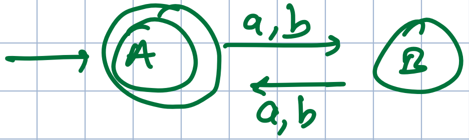
make final state → non final state

non final state → final

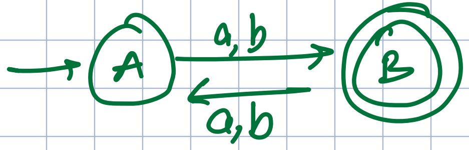
$L_1 = \{Q, \Sigma, \delta, q_0, F\}$

$\overline{L_1} = \{Q, \Sigma, \delta, q_0, Q-F\}$

Eg: L_1 = set of all even length strings

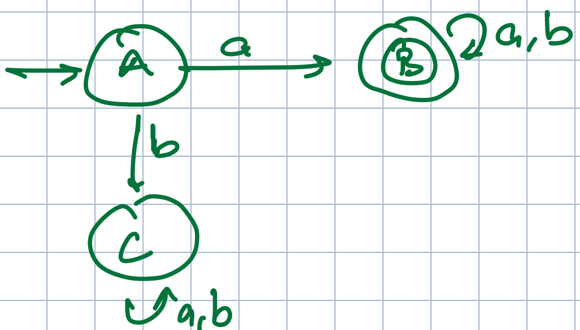


$\overline{L_1}$:



DFA for odd length string

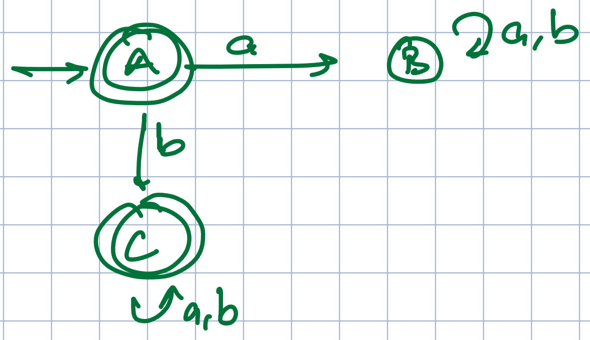
Eg: L_1 = starts with 'a'



Q = {A, B, C}

F = {B}

$\overline{L_1}$ =



F = {A, C}

5. Reversal

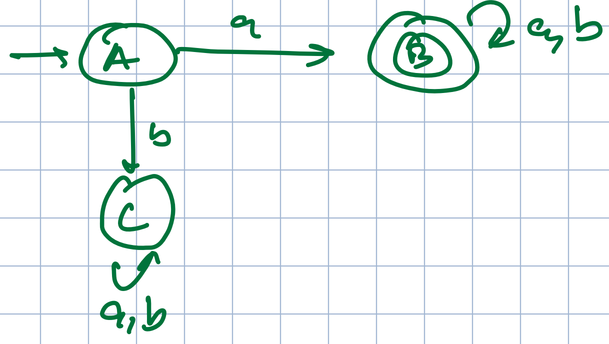
L_1 = starts with 'a'

= {a, aa, aab, ab . . . . }

L_1^R = take each string of L_1 and reverse if

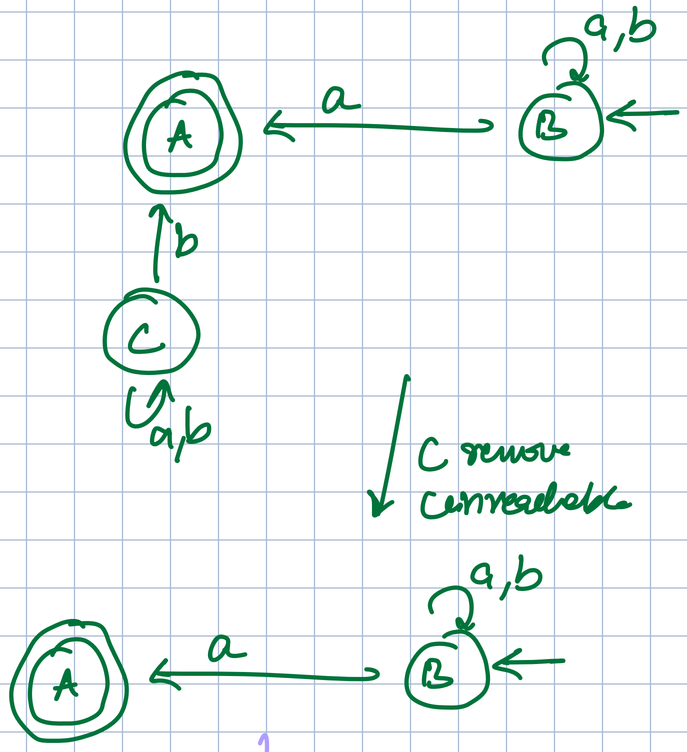
= {a, aa, baa, ba . . . . }

DFA for L_1 =



DFA for L_1^R:

- Draw the states as if is
- final state → initial state
- initial state → final state
- reverse the edges



NFA ↙

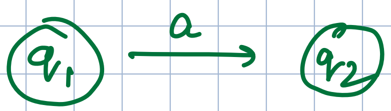
Non Deterministic finite Automata

$L_1 \longrightarrow DFA$

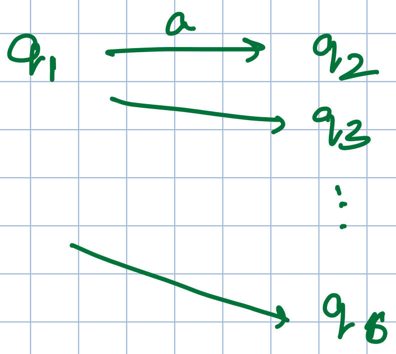
$(DFA)^R \longrightarrow L_1^R \longrightarrow$ DFA or NFA

Non Deterministic finite Automata:

DFA :



NFA :



$(Q, \Sigma, \delta, q_0, F)$

initial state

final state

Set of all States

input alphabet

transition fxn

DFA : $\delta : Q \times \Sigma \longrightarrow Q$

NFA : $\delta : Q \times \Sigma \longrightarrow 2^Q$

$Q : \{A, B\}$

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

$Q \times \Sigma$	2^Q

Subset

$\{1,2,3\}$ $\qquad 2^3$

$\{\}$ $\quad \{12\}$ $\quad \{123\}$

$\{1\}$ $\quad \{13\}$

$\{2\}$

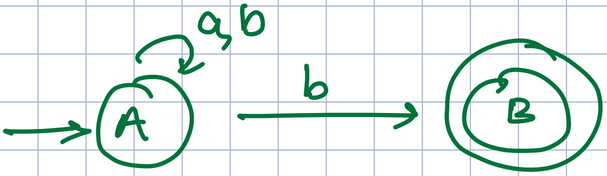
$\{3\}$ $\quad \{22\}$

(A, a)
(A, b)
(B, a)
(B, b)

{Φ} → no transition
{A} goes to A
{B} go to B
{AB} goto both AB

Eg: NFA $\Sigma = \{a, b\}$

strings ends with 'b'



$Q \times \Sigma$	
(A, a)	A
(A, b)	AB
(B, a)	Φ
(B, b)	Φ

$\delta(B, a) = \Phi$

String Acceptence

ab

$A \xrightarrow{a} A \xrightarrow{b} A$ → not final state

$\xrightarrow{b}$ B → final state

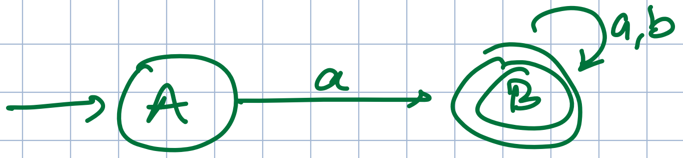
Accept

aa

$A \xrightarrow{a} A \xrightarrow{a} A$ non final state

string is not accepted.

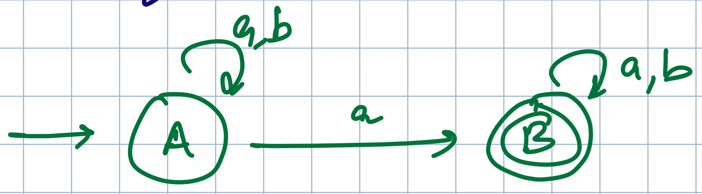
Eg: starts with a



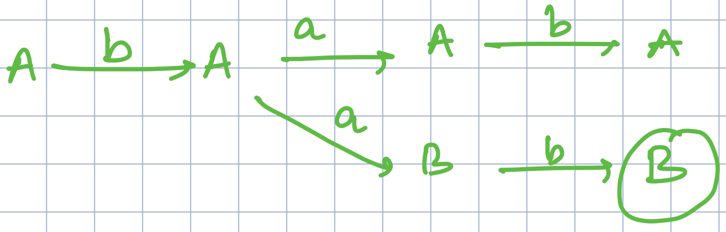
ab

$A \xrightarrow{a} B \xrightarrow{b} \underline{\underline{B}} \rightarrow$ final state

Eg: contains a

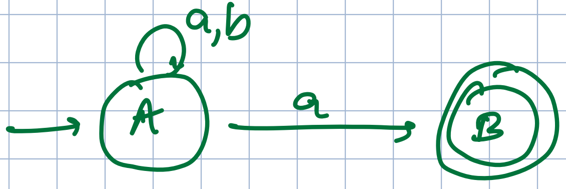


bab



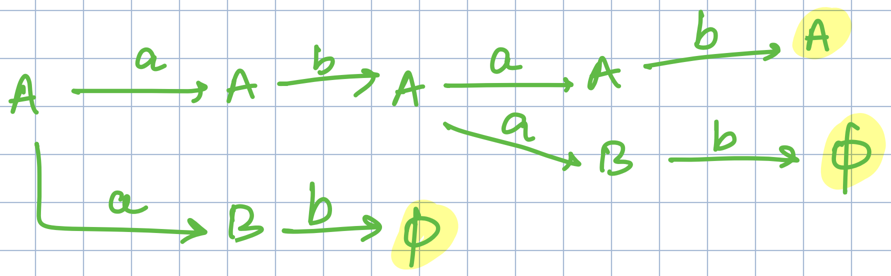
final state
string accept.

Eg: ends with a



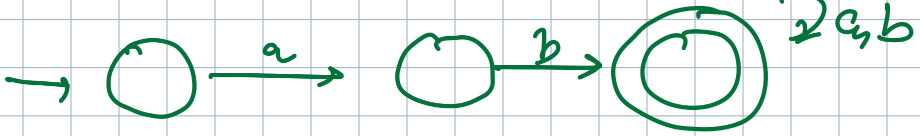
$\underline{ababbba}\ \underline{a}$

abab

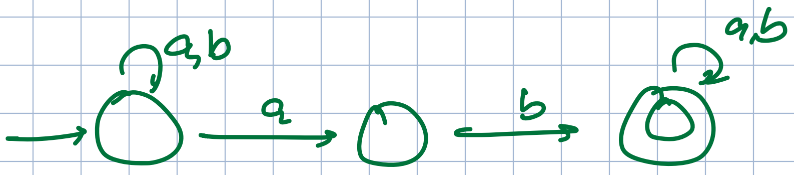


not accepted.

Eg: Starts with 'ab' $\Sigma = \{a,b\}$



Eg: contains 'ab' $\Sigma = \{a,b\}$



Eg: ends 'ab' $\Sigma = \{a,b\}$