MATHEMATICAL FOUNDATION FOR COMPUTER SCIENCE

Prepared by: Er. Ankit Kharel

Nepal college of information technology

FINITE STATE AUTOMATA

- *Sequential Circuits and Finite state Machine*
- *Finite State Automata*
- *Non-deterministic Finite State Automata*
- *Language and Grammars*
- *Language and Automata*
- *Regular Expression*

GRAMMAR TO FINITE AUTOMATA:

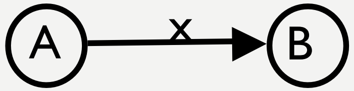
- For each Right Linear Grammar(G_r), there is one finite automata M where L(M) = L(G_R).

1. Right Linear Grammar to NFA:

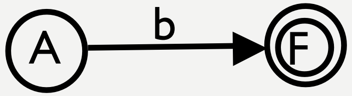
(a) The non- terminals becomes states with σ as an initial state.

(b) The terminal becomes set of alphabets(input)

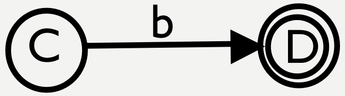
(c) The production of form, A→xB, we draw an edge from state A to B and label it with x.



(d) The production of form C→b is written as C→bF where F is final state.



(e) The production of form C→b, C→bD:



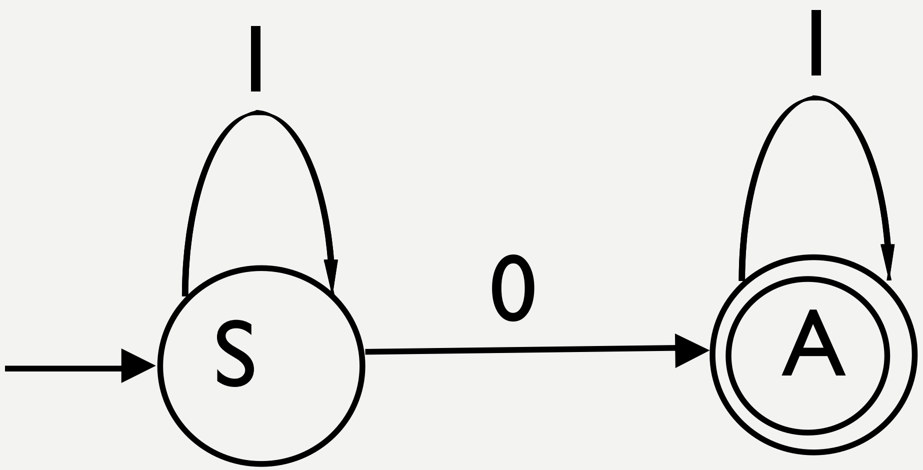
GRAMMAR TO FINITE AUTOMATA:

- Construct a non- deterministic finite automata that recognizes the language generated by the regular grammar, G={N, T, P, σ} where N={A , S} , T={0 , 1}, S is starting symbol and production P are:

  S→1S/0A

  A→1A/1

<u>Solution:</u>



GRAMMAR TO FINITE AUTOMATA:

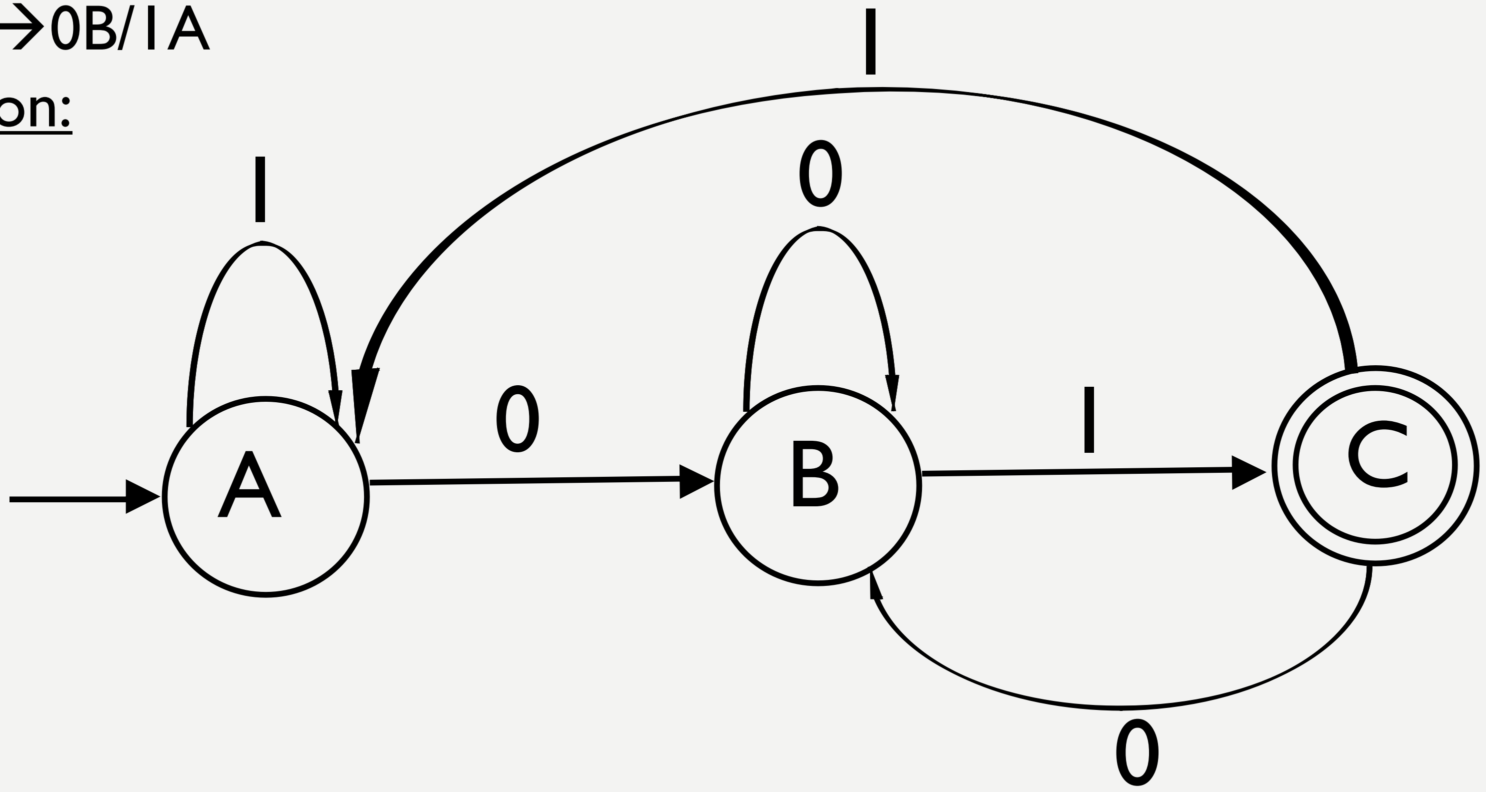
- Construct a non- deterministic finite automata that recognizes the language generated by the regular grammar, G={N, T, P, σ} where N={A , B, C} , T={0 , 1}, A is starting symbol and production P are:

  A→0B/1A

  B→0B/1C/1

  C→0B/1A

Solution:



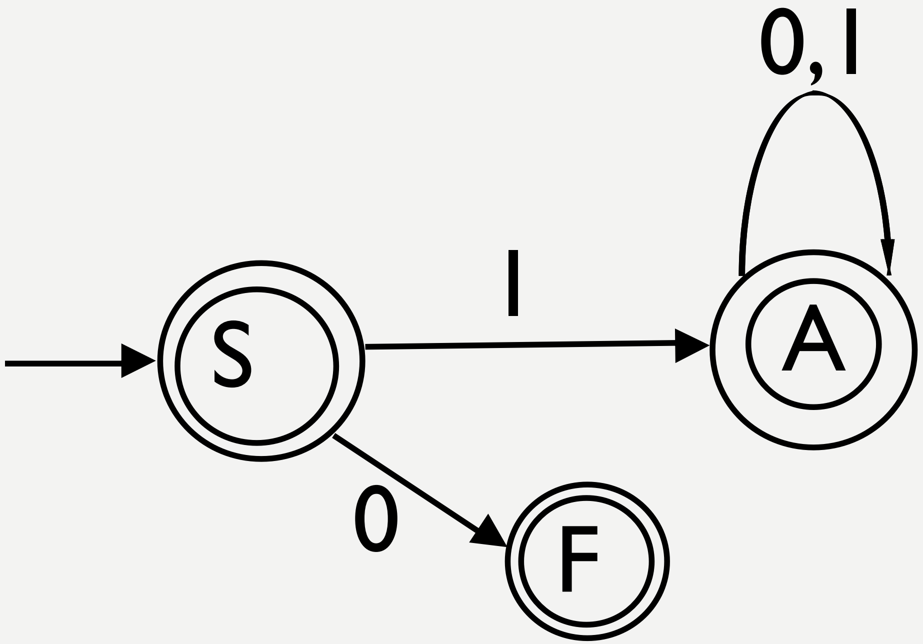
GRAMMAR TO FINITE AUTOMATA:

- Construct a non- deterministic finite automata that recognizes the language generated by the regular grammar, G={N, T, P, σ} where N={A , S} , T={0 , 1}, S is starting symbol and production P are:

  $S \rightarrow 1A/0/\in$

  $A \rightarrow 0A/1A/1$

<u>Solution:</u>



GRAMMAR TO FINITE AUTOMATA:

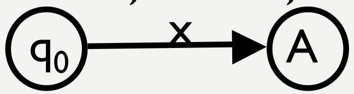
- For a given left linear grammar there is a corresponding finite automata M where $L(M) = L(G_L)$.

2. **LEFT Linear Grammar to NFA:**

(a) Start symbol is the final state.

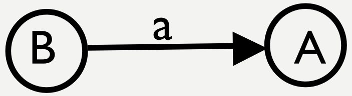
(b) The terminal becomes set of alphabets(input) and non terminal becomes states.

(c) The production of form, $A \rightarrow x$,



where, q_0 is an initial state.

(d) The production of form $A \rightarrow Ba$



GRAMMAR TO FINITE AUTOMATA:

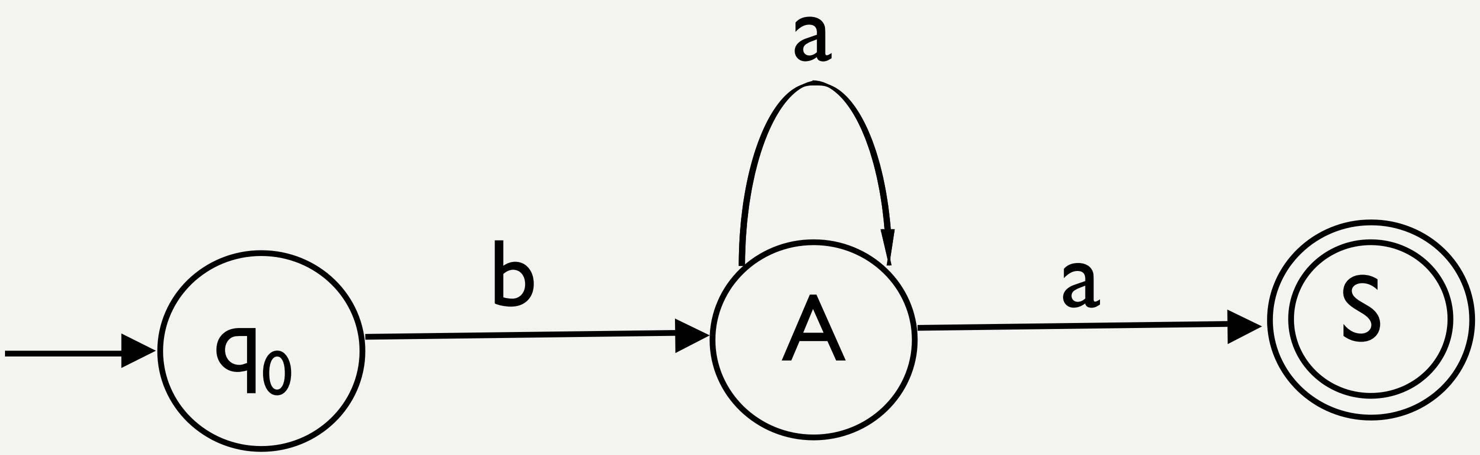
- Construct a non- deterministic finite automata that recognizes the language generated by the regular grammar, G={N, T, P, σ} where N={S , A} , T={a , b}, S is starting symbol and production P are:

  S→Aa

  A→Aa

  A→b

<u>Solution:</u>



GRAMMAR TO FINITE AUTOMATA:

- Construct a non- deterministic finite automata that recognizes the language generated by the regular grammar, G={N, T, P, σ} where N={S , A, B, C} ,T={a , b}, S is starting symbol and production P are:

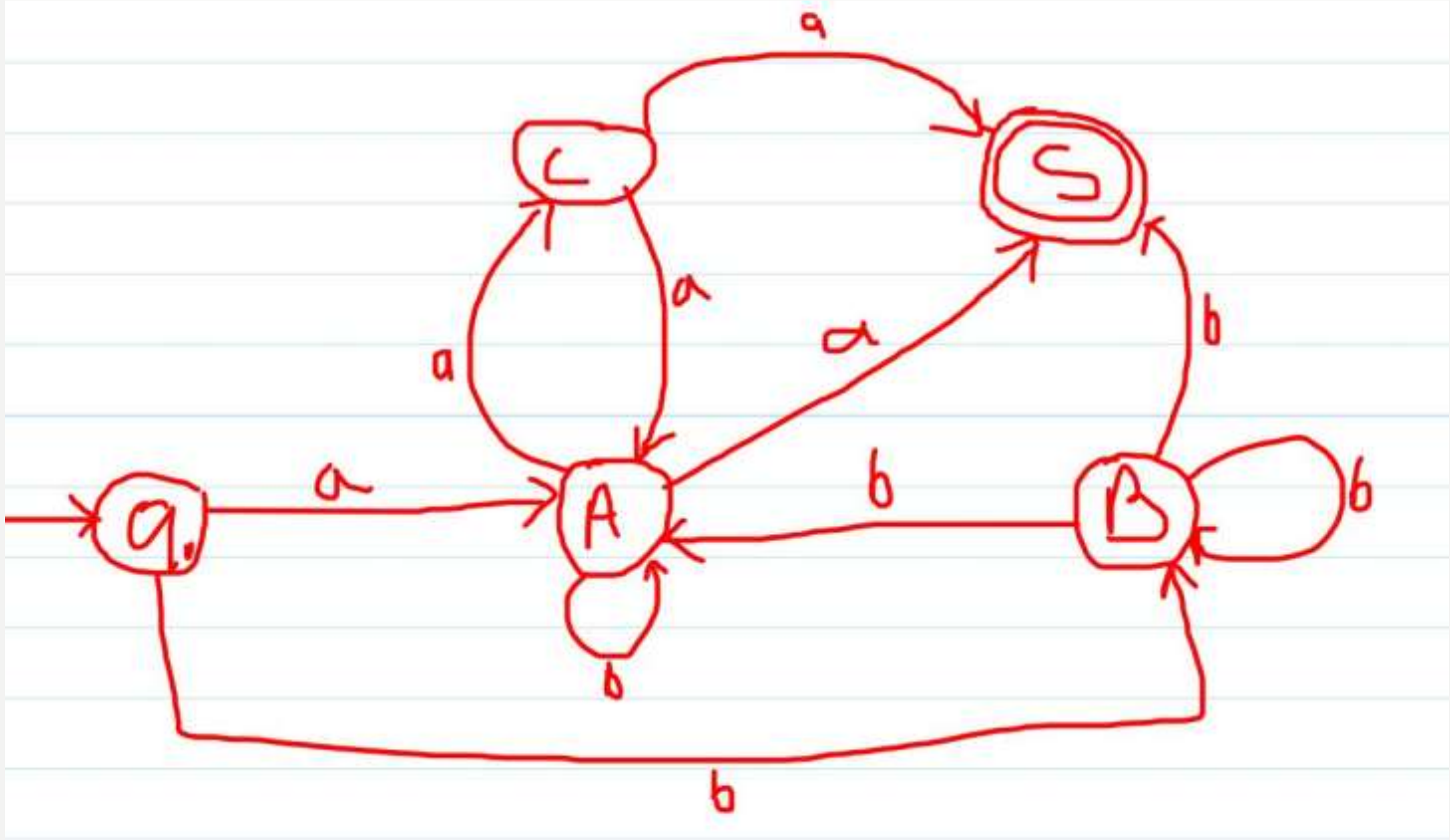
  S→Ca/Aa/Bb

  A→Aa/Ca/Bb/a

  B→Bb/b

  C→Aa

<u>Solution:</u>



FINITE AUTOMATA TO GRAMMAR :

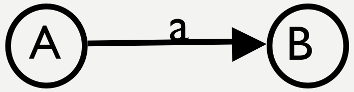
- For each Finite Automata M, there is one right linear grammar G_R where $L(G_R) = L(M)$.

1. **Finite Automata to Right Linear Grammar:**

(a) The set of states becomes non terminal symbols.

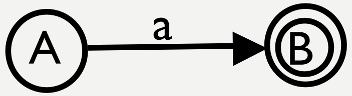
(b) The set of inputs becomes terminal symbols.

(c) Rule 1:



$= A \rightarrow aB$

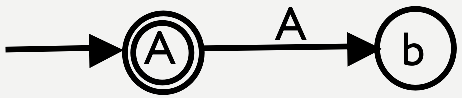
(d) Rule 2:



$= A \rightarrow aB/a$

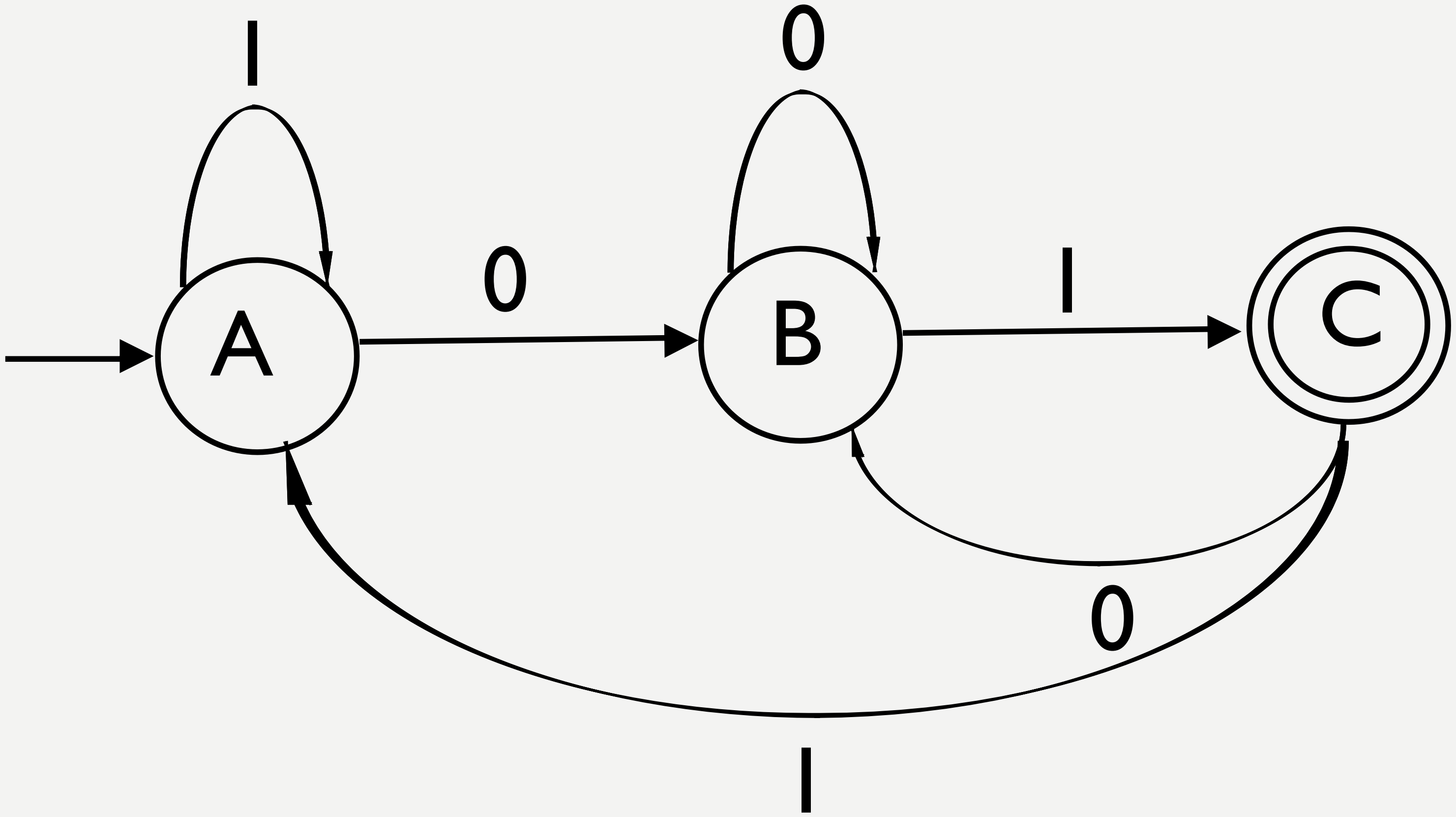
(e) Rule 3:

If initial state is final state then add eplision in a production



$= A \rightarrow aB/ \in$

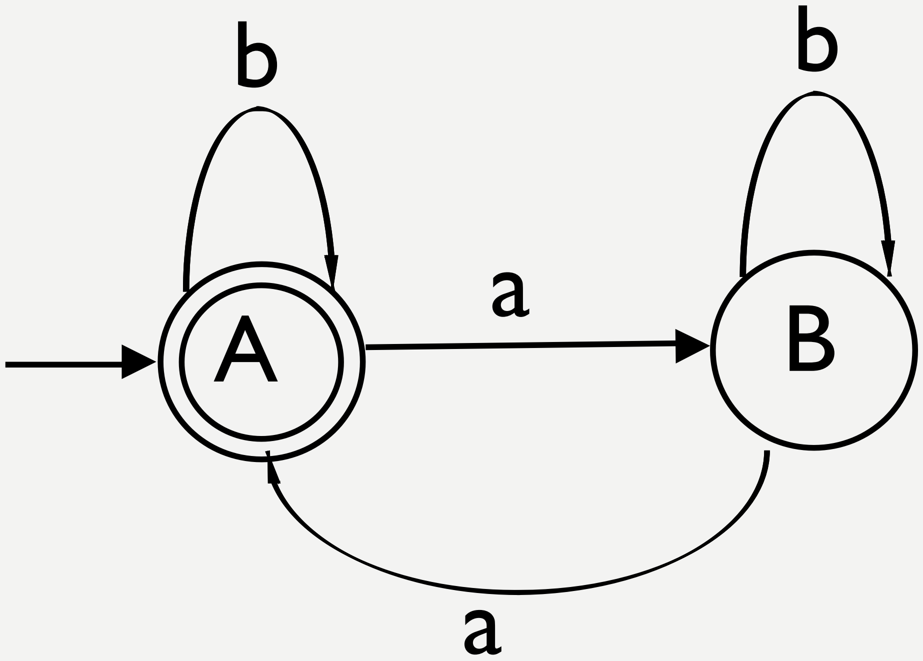
GRAMMAR TO FINITE AUTOMATA:



A→0B/1A
B→0B/1C/1
C→0B/1A

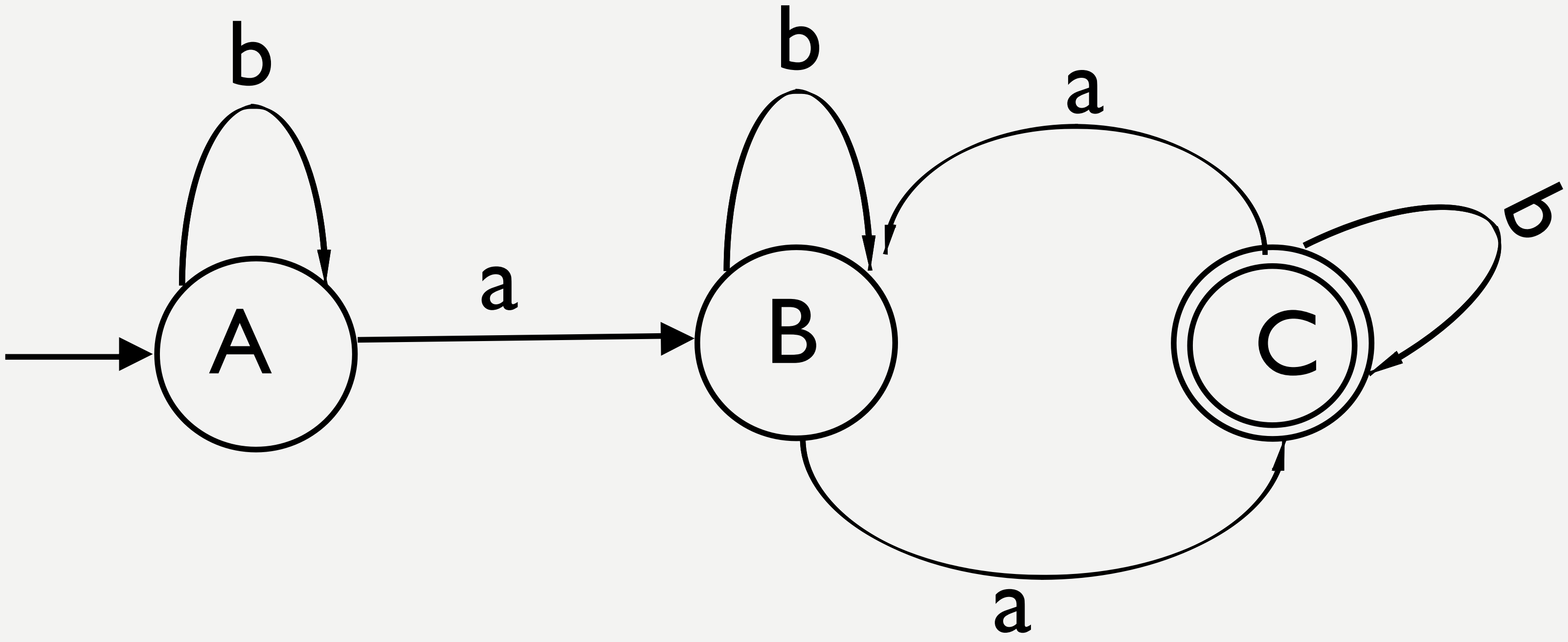
(A is starting symbol)

GRAMMAR TO FINITE AUTOMATA:



A→aB/bA/b/∈ (A is starting symbol)
B→bB/aA/a

GRAMMAR TO FINITE AUTOMATA:



A→aB/bA (A is starting symbol)
B→bB/aC/a
C→bC/aB/b

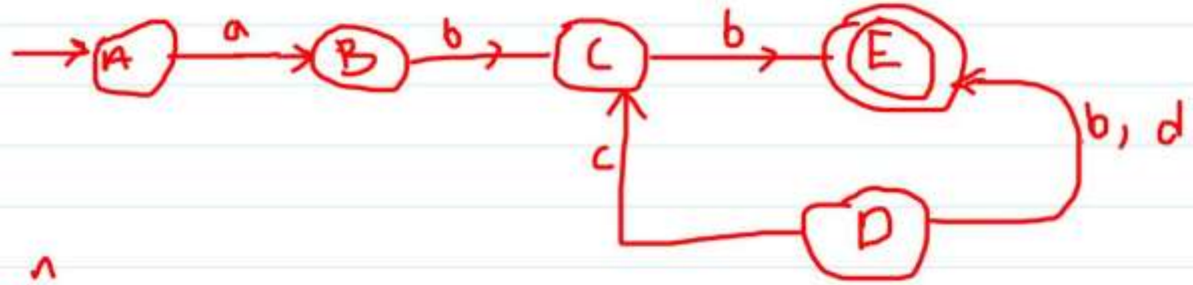
FINITE AUTOMATA TO GRAMMAR :

- For each Finite Automata M, there is one left linear grammar G_L where $L(G_L) = L(M)$.

1. **Finite Automata to Left Linear Grammar:**

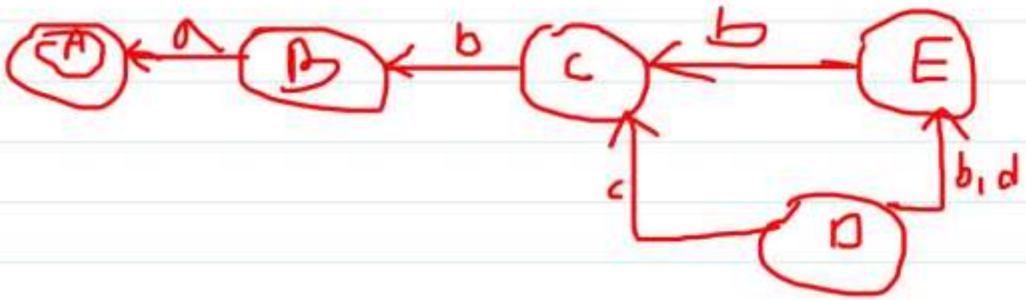
(a) The set of states becomes non terminal symbols.
(b) The set of inputs becomes terminal symbols.
(c)Reverse the edges of NFA and exchange initial and Final state.
(d)Construct Right Linear Grammar
(e)Reverse the production and obtain left linear grammar.

(B)



Sol^n

First Reverse the edges and initial and final state



Obtain right linear grammer

$E \rightarrow bC / bD / dD$
$C \rightarrow bB$
$B \rightarrow aA | a$
$D \rightarrow cC$

Reverse

$E \rightarrow Cb / Db / Dd$
$C \rightarrow Bb$
$B \rightarrow Aa / a$
$D \rightarrow Cc$