# Regular Grammar

By

NAYAN KUMAR

# Regular Grammar

Regular grammar is a type of grammar that describes a regular language.

A regular grammar is a mathematical object, G, which consists of four components, G = (V, T, P, S), where

- **V:** non-empty, finite set of non-terminal symbols,
- **T:** a finite set of terminal symbols, or alphabet, symbols,
- **P:** a set of grammar rules, each of one having one of the forms
  - $A \rightarrow aB$
  - $A \rightarrow a$
  - $A \rightarrow \in$, Here $\in$=empty string, $A, B \in N$, $a \in \sum$
- **S** $\in$ V is the start symbol.

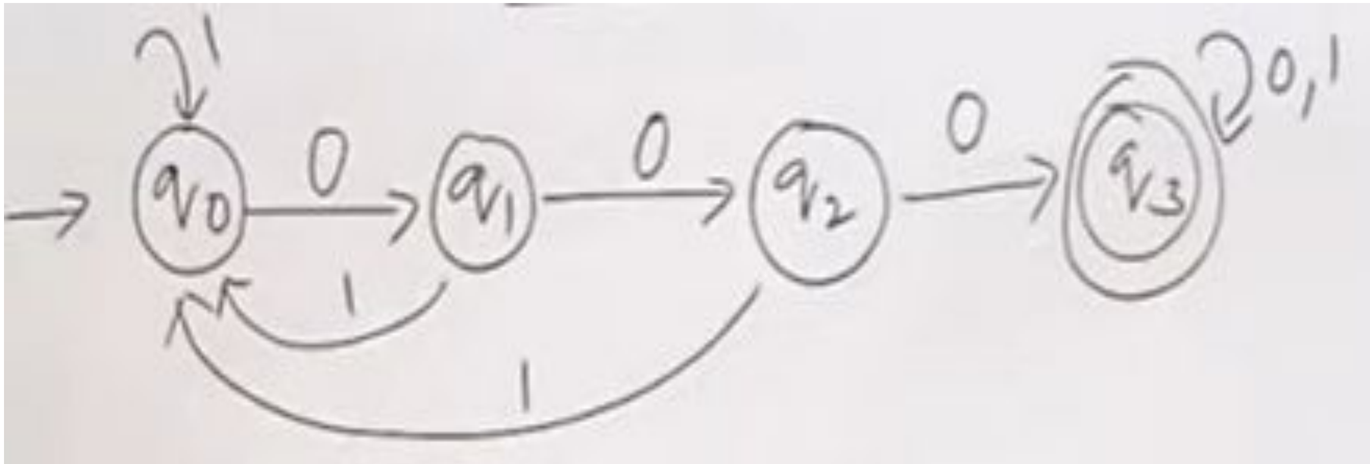
# DFA to REGULAR GRAMMAR

1. ***Ai->aAj*** , if **(qi,a)=qj** where **qj ∉ F**
2. ***Ai->aAj*** and **A->a** are the production rules, if **(qi,a)=qj** where **qj ∈ F**

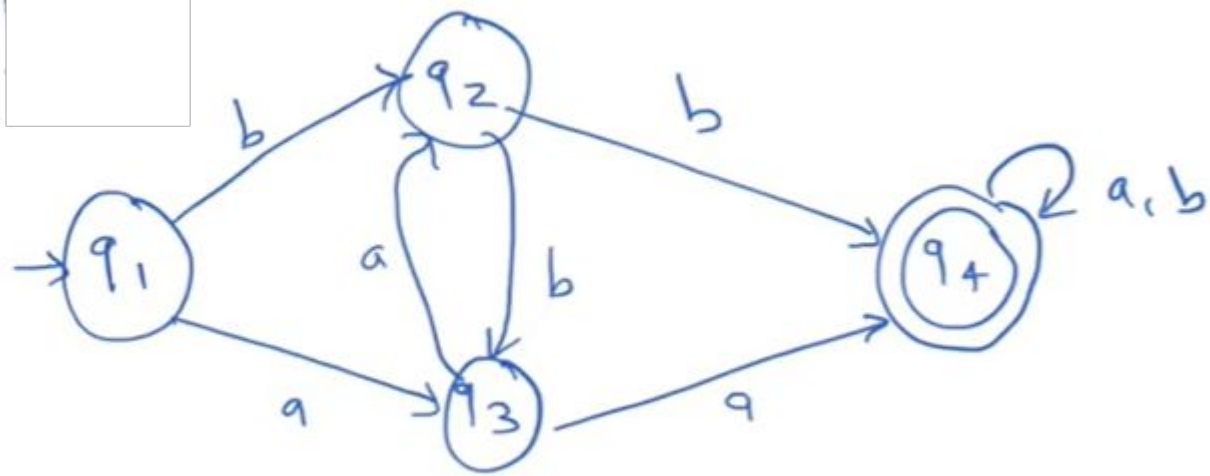
# Example-1

Construct RG from given DFA:

## Example-2



# Example-3



## REGULAR GRAMMAR to DFA

1. Each productions ***Ai->aAj*** , induces a transition from **qi** to **qj** with label **a**.
2. Each production ***Ak->a*** induces a transition from **qk** to **qf** with label **'a'.**

# Example-4

Construct a finite automata recognizing L(G) where G is the grammar

S-> aS | bA | b

A->aA | bS |a

# Example-5

Construct a finite automata recognizing L(G) where G is the grammar

S-> aS | bS | aA

A-> bB

B-> C

C-a

## Left Linear Regular Grammar

In this type of regular grammar, all the non-terminals on the left-hand side exist at the leftmost place, i.e; left ends.ing

A ⇢ a, A ⇢ **B**a, A ⇢ ∈ , where

A and B are non-terminals,

a is terminal, and

∈ is empty string

Example

S ⇢ **B**00 | **S**11
B ⇢ **B**0 | **B**1 | 0 | 1
where
S and B are non-terminals, and
0 and 1 are terminals

# Right Linear Regular Grammar

In this type of regular grammar, all the non-terminals on the right-hand side exist at the rightmost place, i.e; right ends

$A \dashrightarrow a$, $A \dashrightarrow a\mathbf{B}$, $A \dashrightarrow \in$ where,

A and B are non-terminals,

a is terminal, and

$\in$ is empty string

**Example**

$S \dashrightarrow 00\mathbf{B} \mid 11\mathbf{S}$

$B \dashrightarrow 0\mathbf{B} \mid 1\mathbf{B} \mid 0 \mid 1$

where,

S and B are non-terminals, and

0 and 1 are terminals

# Examples-6

Find a right linear grammar (RLG) for the language

$L = \{a^n b^m : n + m \text{ is odd}\}$.

Convert the RLG to left linear grammar (LLG)

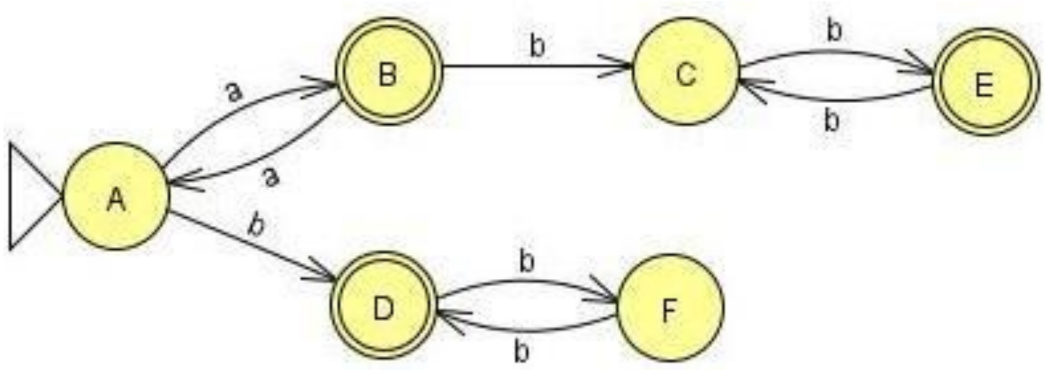
# Examples-6

Find a right linear grammar (RLG) for the language

L = {a^n b^m : n + m is odd}.

Convert the RLG to left linear grammar (LLG)

STEP-1 [Construct the DFA]



# Examples-6

**STEP-2** [Find RLG from the FA]

Right Linear Grammar:

A → aB | bD

B → bC | aA |

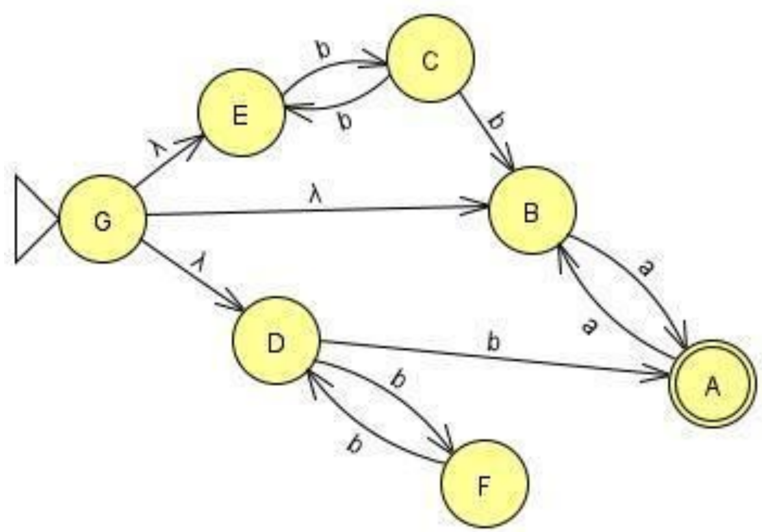
C → bE

D → bF |

E → bC |

F → bD

**STEP-3** [Find the reverse FA]



# Examples-6

**STEP-4** [Right Linear Grammar for the Reverse FA]

:$G \rightarrow B \mid D \mid E$

$B \rightarrow aA$

$D \rightarrow bA \mid bF$

$E \rightarrow bC$

$A \rightarrow aB \mid \in$

$C \rightarrow bE \mid bB$

$F \rightarrow bD$

**STEP-5** [Left Linear Grammar for the given Language]

$G \rightarrow B \mid D \mid E$

$B \rightarrow Aa$

$D \rightarrow Ab \mid Fb$

$E \rightarrow Cb$

$A \rightarrow Ba \mid \in$

$C \rightarrow Eb \mid Bb$

$F \rightarrow Db$

# Examples-7

Construct a DFA that accepts the language generated by the

following grammar. Write its regular expression.

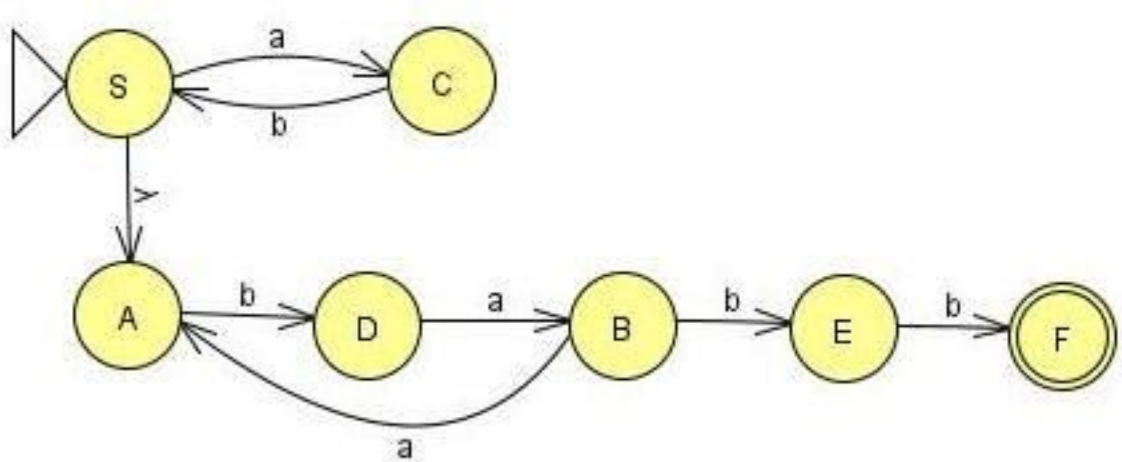
$S \rightarrow abS|A$

$A \rightarrow baB$

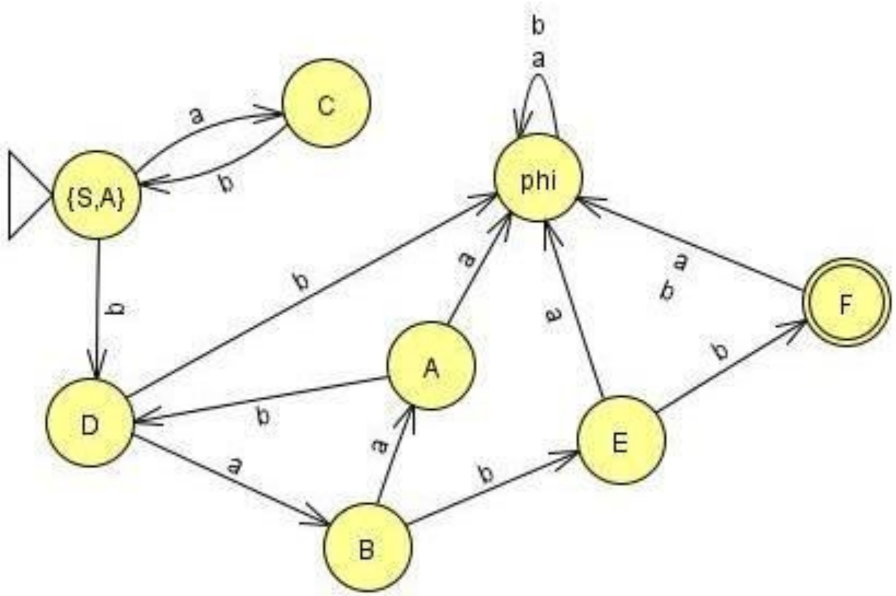
$B \rightarrow aA|bb$

# Examples-7

## STEP-1 [The constructed NFA is]




## STEP-2 [The constructed DFA]




## STEP-3 [Regular Exp ]

The desired RE is:

**(ab)*(baa)*babb**

## Left Linear Regular Grammar

In this type of regular grammar, all the non-terminals on the left-hand side exist at the leftmost place, i.e; left ends.ing

A ⇢ a, A ⇢ **B**a, A ⇢ ∈ , where

A and B are non-terminals,

a is terminal, and

∈ is empty string

Example

S ⇢ **B**00 | **S**11
B ⇢ **B**0 | **B**1 | 0 | 1
where
S and B are non-terminals, and
0 and 1 are terminals