Regular Expression

Regular Expressions are used for representing certain sets of strings in an algebraic fashion.

1) Any terminal symbol i.e. symbols $\in \Sigma$ including Λ and Φ are regular expressions. $a, b, c, \ldots \Lambda, \Phi$

2) The Union of two regular expressions is also a regular expression. $R_1, R_2 \quad (R_1 + R_2)$

3) The Concatenation of two regular expressions is also a regular expression. $R_1, R_2 \rightarrow (R_1 . R_2)$

4) The iteration (or Closure) of a regular expression is also a regular expression. $R \rightarrow R^*$ $\quad a^* = \Lambda, a, aa, aaa, \ldots$

5) The regular expression over Σ are precisely those obtained recursively by the application of the above rules once or several times.

Regular Expression - Examples

Describe the following sets as Regular Expressions

1) $\{0,1,2\}$ $\quad$ 0 or 1 or 2

$R = 0 + 1 + 2$

2) $\{\Lambda, ab\}$

$R = \Lambda \; ab$

3) $\{abb, a, b, bba\}$ $\quad$ abb or a or b or bba

$R = abb + a + b + bba$

4) $\{\Lambda, 0, 00, 000, \ldots\ldots\}$ $\quad$ closure of 0

$R = 0^*$

5) $\{1, 11, 111, 1111, \ldots\ldots\}$

$R = 1^+$

Identities of Regular Expression

1) $\emptyset + R = R$

2) $\emptyset R + R\emptyset = \emptyset$ $\quad \Lambda$

3) $\epsilon R = R\epsilon = R$

4) $\epsilon^* = \epsilon$ and $\emptyset^* = \epsilon$

5) $R + R = R$

6) $R^*R^* = R^*$

7) $RR^* = R^*R$ $\quad R^+$

8) $(R^*)^* = R^*$

9) $\epsilon + RR^* = \epsilon + R^*R = R^*$

10) $(PQ)^*P = P(QP)^*$

11) $(P + Q)^* = (P^* Q^*)^* = (P^* + Q^*)^*$

12) $(P + Q) R = PR + QR$ and

$R(P + Q) = RP + RQ$

ARDEN'S THEOREM

If P and Q are two Regular Expressions over Σ, and if P does not contain ϵ, then the following equation in R given by $R = Q + RP$ has a unique solution i.e. $R = QP^*$

$$R = Q + RP \longrightarrow (1)$$

$$R = QP^*$$

$$= Q + QP^*P$$

$$= Q(\epsilon + P^*P) \qquad [\epsilon + R^*R = R^*]$$

$$= QP^* \quad \text{Proved}$$

$$R = Q + RP$$

$$= Q + [Q + RP]P$$

$$= Q + [Q + RP]P$$

$$= Q + QP + RP^2$$

$$= Q + QP + [Q + RP]P^2$$

$$= Q + QP + QP^2 + RP^3$$

$$\vdots$$

$$= Q + QP + QP^2 + \ldots QP^n + RP^{n+1} \qquad [R = QP^*]$$

$$= Q + QP + QP^2 + \ldots QP^n + QP^*P^{n+1}$$

$$= Q[\epsilon + P + P^2 + \ldots P^n + P^*P^{n+1}]$$

$$R = QP^*$$

An Example Proof using Identities of Regular Expressions

Prove that $(1+00^*1) + (1+00^*1)(0+10^*1)^*(0+10^*1)$ is equal to $0^*1(0+10^*1)^*$

$$LHS = (1+00^*1) + (1+00^*1)(0+10^*1)^*(0+10^*1)$$

$$= (1+00^*1)[\epsilon + (0+10^*1)^*(0+10^*1)] \qquad \epsilon + R^*R = R^*$$

$$= (1+00^*1)(0+10^*1)^* \qquad \epsilon \cdot R = R$$

$$= (\epsilon \cdot 1+00^*1)(0+10^*1)^*$$

$$= (\epsilon+00^*)1(0+10^*1)^*$$

$$= 0^*1(0+10^*1)^* = RHS$$

Designing Regular Expressions - Examples (Part-1)

Design Regular Expression for the following languages over {a,b}

1) Language accepting strings of length exactly 2
2) Language accepting strings of length atleast 2
3) Language accepting strings of length atmost 2

Soln

1) $L_1 = \{aa, ab, ba, bb\}$

$$R = aa + ab + ba + bb$$
$$= a(a+b) + b(a+b)$$
$$= (a+b)(a+b)$$

2) $L_1 = \{aa, ab, ba, bb, aaa, \ldots\}$

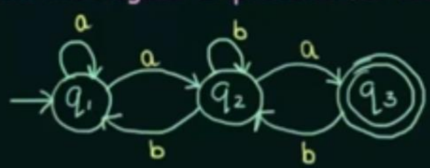
$$R = (a+b)(a+b)(a+b)^*$$

3) $L_1 = \{\epsilon, a, b, aa, ab, ba, bb\}$

$$R = \epsilon + a + b + aa + ab + ba + bb$$
$$= (\epsilon + a + b)(\epsilon + a + b)$$

Designing Regular Expression - Examples (Part-2)

Find the Regular Expression for the following NFA



$q_3 = q_2 a \rightarrow ①$

$q_2 = q_1 a + q_2 b + q_3 b \rightarrow ②$

$q_1 = \epsilon + q_1 a + q_2 b \rightarrow ③$

① ⇒ $q_3 = q_2 a$

$$= (q_1 a + q_2 b + q_3 b) a$$
$$= q_1 aa + q_2 ba + q_3 ba \rightarrow ④$$

② ⇒ $q_2 = q_1 a + q_2 b + q_3 b$ Putting value of q_3 from ①

$$= q_1 a + q_2 b + (q_2 a) b$$

$$= (q_1 a + q_2 b + q_3 b) a$$
$$= q_1 aa + q_2 ba + q_3 ba \rightarrow ④$$

② ⇒ $q_2 = q_1 a + q_2 b + q_3 b$ Putting value of q_3 from ①

$$= q_1 a + q_2 b + (q_2 a) b$$

$$= q_1 a + q_2 b + q_2 ab$$

$$\underbrace{q_2}_{R} = \underbrace{q_1 a}_{Q} + \underbrace{q_2}_{R}\underbrace{(b + ab)}_{P}$$

$R = Q + RP$ Arden's Theorem

$R = QP^*$

$$q_2 = (q_1 a)(b + ab)^* \rightarrow ⑤$$

$$q_2 = (q_1 a)(b+ab)^* \rightarrow (5)$$

(3) $\Rightarrow$ $q_1 = \epsilon + q_1 a + q_2 b$

Putting value of q_2 from (5)

$$q_1 = \epsilon + q_1 a + ((q_1 a)(b+ab)^*) b \qquad R = Q + RP$$

$$\underbrace{q_1}_{R} = \underbrace{\epsilon}_{Q} + \underbrace{q_1}_{R} \underbrace{(a + a(b+ab)^*)b}_{P} \qquad R = QP^*$$

$$q_1 = \epsilon((a + a(b+ab)^*)b)^* \qquad \epsilon \cdot R = R$$

$$q_1 = (a + a(b+ab)^* b)^* \rightarrow (6)$$

$$q_1 = \epsilon((a + a(b+ab)^*)b)^* \qquad \epsilon \cdot R = R$$

$$q_1 = (a + a(b+ab)^* b)^* \rightarrow (6)$$

Final state q_3

$$q_3 = q_2 a$$

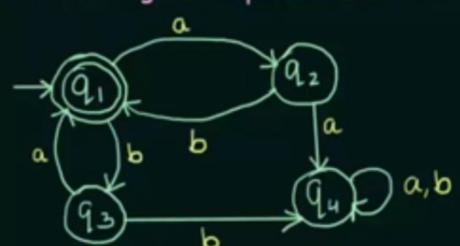
$$= q_1 a (b+ab)^* a \quad \text{Putting value of } q_2 \text{ from (5)}$$

$$q_3 = (a + a(b+ab)^* b)^* a (b+ab)^* a \quad \text{Putting value of } q_1 \text{ from (6)}$$

= Required Regular Expression for the given NFA

Designing Regular Expression - Examples (Part-3)

Find the Regular Expression for the following DFA



$$q_1 = \epsilon + q_2 b + q_3 a \rightarrow (\mathrm{I})$$

$$q_2 = q_1 a \rightarrow (\mathrm{II})$$

$$q_3 = q_1 b \rightarrow (\mathrm{III})$$

$$q_4 = q_2 a + q_3 b + q_4 a + q_4 b \rightarrow (\mathrm{IV})$$

(I) $\Rightarrow$ $q_1 = \epsilon + q_2 b + q_3 a$

Putting values of q_2 and q_3 from (II) and (III)

$$q_1 = \epsilon + q_1 ab + q_1 ba$$

$q_4 = q_2 a + q_3 b + q_4 a + q_4 b \rightarrow$ (IV)

(I) → $q_1 = \epsilon + q_2 b + q_3 a$

Putting values of q_2 and q_3 from (II) and (III)

$q_1 = \epsilon + q_1 ab + q_1 ba$

$\underbrace{q_1}_{R} = \underbrace{\epsilon}_{Q} + \underbrace{q_1}_{R}\underbrace{(ab + ba)}_{P}$

$R = Q + RP$

$R = QP^*$ Arden's theorem

$q_1 = \epsilon . (ab + ba)^*$

$\epsilon R = R$

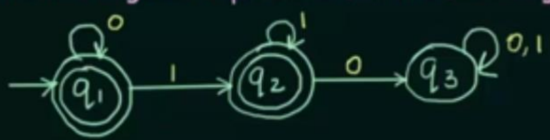
$q_1 = (ab + ba)^*$

↳ Regular Expression.

Designing Regular Expression - Examples (Part-4)

(When there are Multiple Final States)

Find the Regular Expression for the following DFA



$q_1 = \epsilon + q_1 0 \rightarrow$ (I)

$q_2 = q_1 1 + q_2 1 \rightarrow$ (II)

$q_3 = q_2 0 + q_3 0 + q_3 1 \rightarrow$ (III)

Final state q_1

(I) → $\underbrace{q_1}_{R} = \underbrace{\epsilon}_{Q} + \underbrace{q_1}_{R}\underbrace{0}_{P}$

$R = Q + RP$

$R = QP^*$ Arden's theorem

$q_1 = \epsilon . 0^*$

$\epsilon R = R$

$q_1 = 0^* \rightarrow$ (4)

Final state q_2

(II) → $q_2 = q_1 1 + q_2 1$

$\underbrace{q_2}_{R} = \underbrace{0^* 1}_{Q} + \underbrace{q_2}_{R}\underbrace{1}_{P}$ Putting value of q_1 from (4)

$R = Q + RP$

$R = QP^*$

$q_2 = 0^* 1 (1)^*$

R = union of both Final states

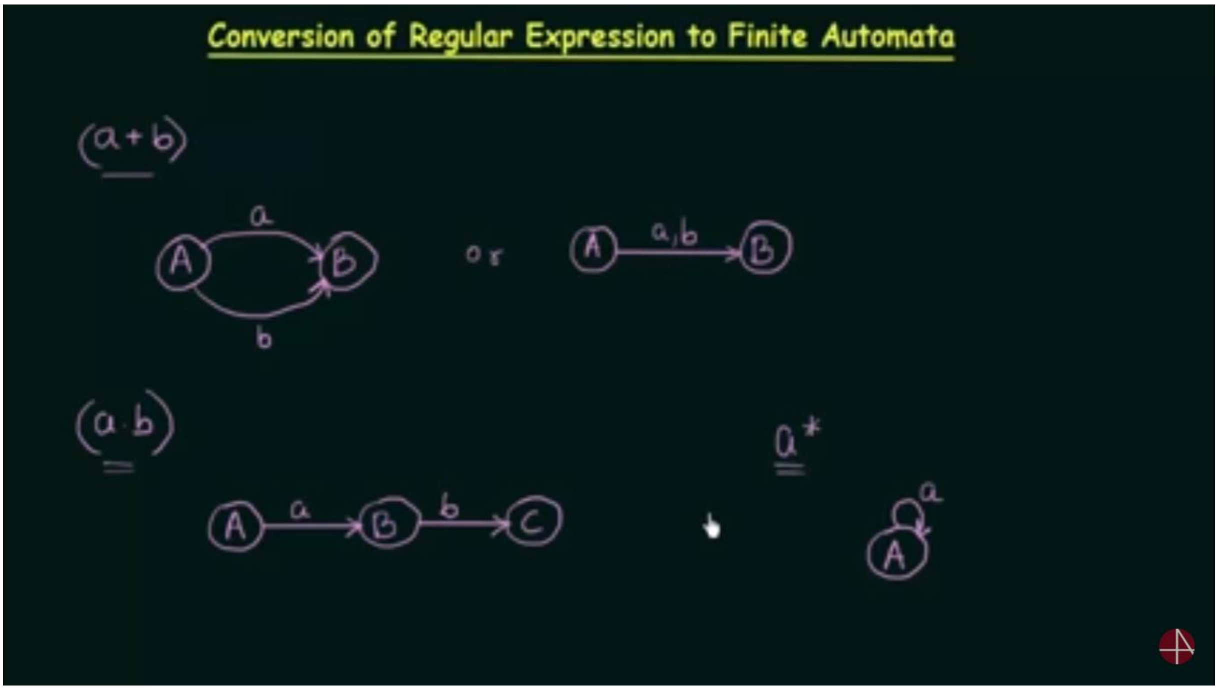
$= 0^* + 0^* 1 1^*$

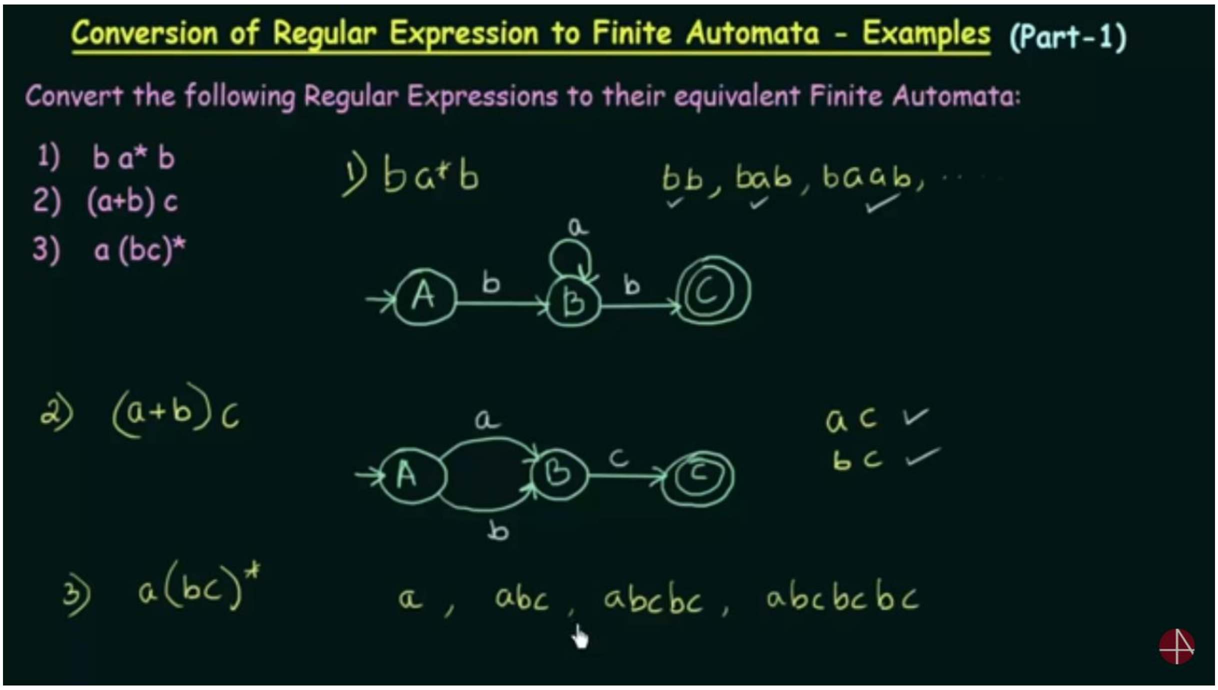
$= 0^* (\epsilon + 1 1^*)$

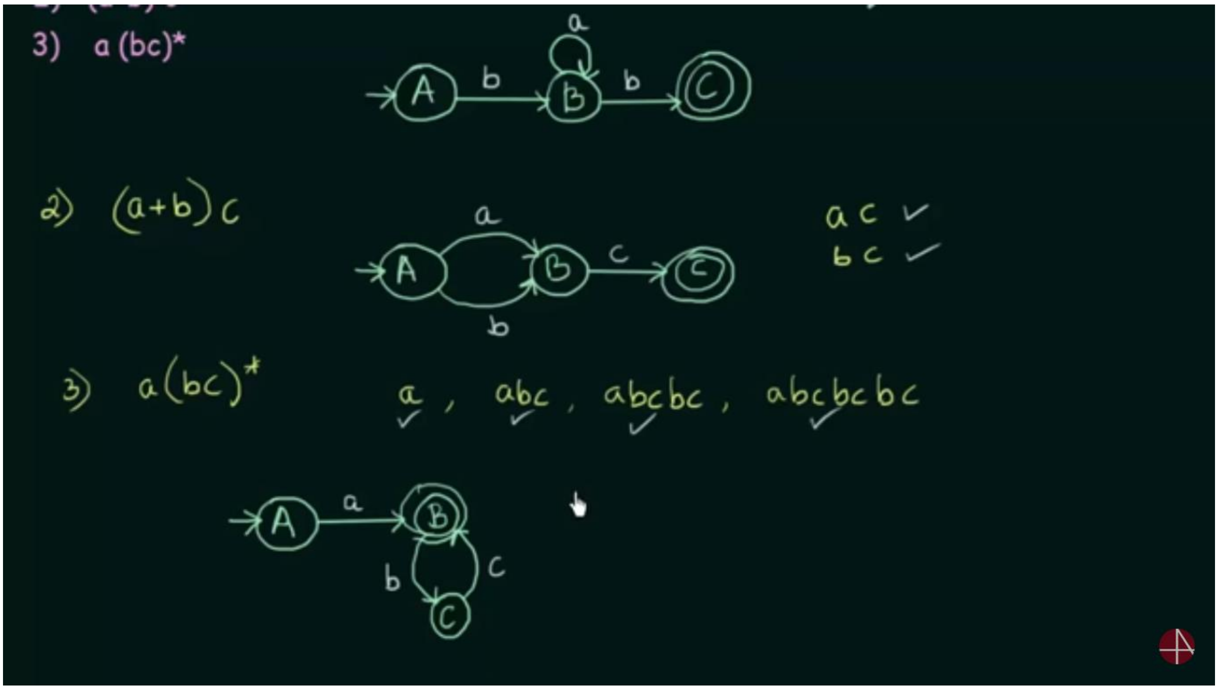
$\epsilon + RR^* = R^*$

$= 0^* 1^*$

↳ Regular Expression


Conversion of Regular Expression to Finite Automata
(a+b)
A
B
a
b
or
A
B
a,b
(a.b)
A
B
C
a
b
a*
A
a

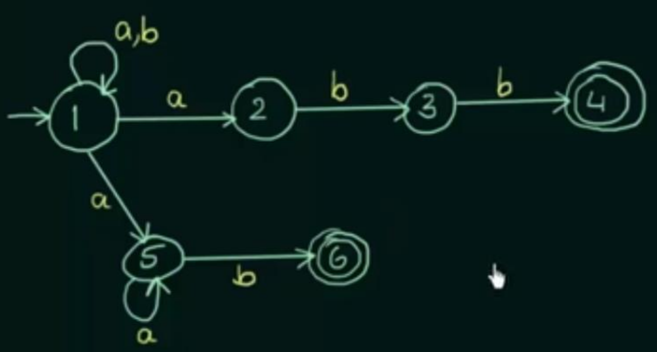

Conversion of Regular Expression to Finite Automata - Examples (Part-1)
Convert the following Regular Expressions to their equivalent Finite Automata:
1) b a* b
2) (a+b) c
3) a (bc)*
1) ba*b
bb, bab, baab, ...
A
B
C
a
b
b
2) (a+b) c
A
B
C
a
b
c
a c
b c
3) a(bc)*
a , abc , abcbc , abcbcbc


3) a (bc)*
A
B
C
a
b
b
2) (a+b) c
A
B
C
a
b
c
a c
b c
3) a(bc)*
a , abc , abcbc , abcbcbc
A
B
C
a
b
c

Conversion of Regular Expression to Finite Automata - Examples (Part-2)

Convert the following Regular Expression to its equivalent Finite Automata:

(a|b)* (abb|a⁺b)



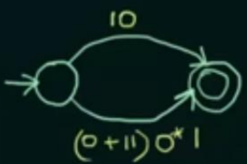
$a^+ = \{a, aa, aaa, \ldots\}$

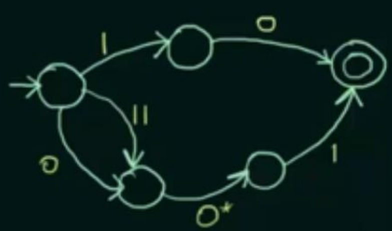
$a^* = \{\epsilon, a, aa, \ldots\}$

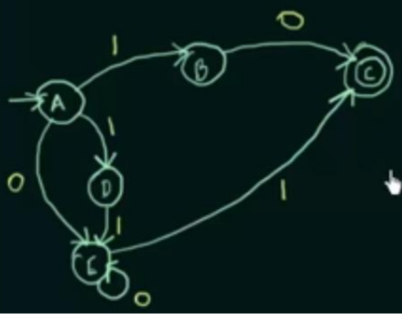
Conversion of Regular Expression to Finite Automata - Examples (Part-3)

Convert the following Regular Expression to its equivalent Finite Automata:

10 + (0 + 11) 0*1







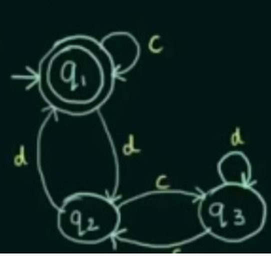
Equivalence of two Finite Automata

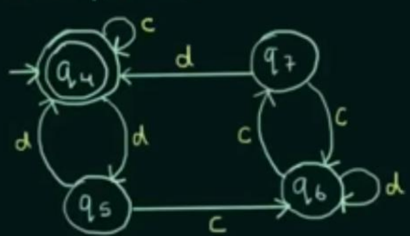
Steps to identify equivalence

1) For any pair of states $\{q_i, q_j\}$ the transition for input $a \in \Sigma$ is defined by $\{q_a, q_b\}$ where $\delta\{q_i, a\} = q_a$ and $\delta\{q_j, a\} = q_b$
The two automata are not equivalent if for a pair $\{q_a, q_b\}$ one is INTERMEDIATE State and the other is FINAL State.

2) If Initial State is Final State of one automaton, then in second automaton also Initial State must be Final State for them to be equivalent.

Eg





A

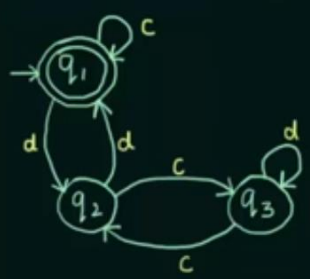
B

States	c	d
(q_1, q_4)	(q_1, q_4) FS FS	(q_2, q_5) IS IS
(q_2, q_5)	(q_3, q_6) IS IS	(q_1, q_4) FS FS
(q_3, q_6)	(q_2, q_7) IS IS	(q_3, q_6) IS IS
(q_2, q_7)	(q_3, q_6) IS IS	(q_1, q_4) FS FS

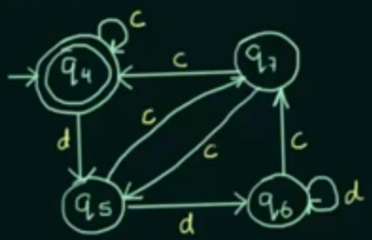
A and B are equivalent

Equivalence of two Finite Automata (Example)

Find out whether the following automata are equivalent or not

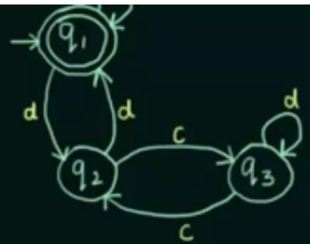


A

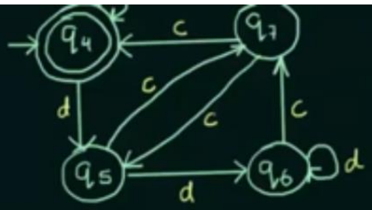


B

States	c	d
$\{q_1, q_4\}$	$\{q_1, q_4\}$ FS FS	$\{q_2, q_5\}$ IS IS



A



B

States	c	d
$\{q_1, q_4\}$	$\{q_1, q_4\}$ FS FS	$\{q_2, q_5\}$ IS IS
$\{q_2, q_5\}$	$\{q_3, q_7\}$ IS IS	$\{q_1, q_6\}$ FS IS

A and B are not equivalent

Pumping Lemma (For Regular Languages)

>> Pumping Lemma is used to prove that a Language is NOT REGULAR

>> It cannot be used to prove that a Language is Regular

If A is a Regular Language, then A has a Pumping Length 'P' such that any string 'S' where $|S| \geq P$ may be divided into 3 parts $S = x\,y\,z$ such that the following conditions must be true:

(1) $x y^i z \in A$ for every $i \geq 0$
(2) $|y| > 0$
(3) $|xy| \leq P$

To prove that a language is not Regular using PUMPING LEMMA, follow the below steps:

(We prove using Contradiction)

-> Assume that A is Regular
-> It has to have a Pumping Length (say P)

(1) $x y^i z \in A$ for every $i \geq 0$
(2) $|y| > 0$
(3) $|xy| \leq P$

To prove that a language is not Regular using PUMPING LEMMA, follow the below steps:

(We prove using Contradiction)

-> Assume that A is Regular
-> It has to have a Pumping Length (say P)

-> All strings longer than P can be pumped $|S| \geq P$
-> Now find a string 'S' in A such that $|S| \geq P$
-> Divide S into x y z
-> Show that $x y^i z \notin A$ for some i
-> Then consider all ways that S can be divided into x y z
-> Show that none of these can satisfy all the 3 pumping conditions at the same time
-> S cannot be Pumped == CONTRADICTION

PUMPING LEMMA

Prove that $L = \{0^i 1^i \mid i \geq 1\}$ is not regular.

$n \rightarrow$ states

1) $w = 0^n 1^n$ $\quad |w| = |0^n 1^n| = 2n. \quad \{2n > n\}$

$w = xyz \Rightarrow \underbrace{\overset{x}{0}\overset{y}{0}0\ldots0}_{n}\ \underbrace{111\ldots1}_{n}$

$y = 0^k$ $\quad \leftarrow xy \rightarrow$

$w = 0^n 1^n \Rightarrow \underset{x}{0^{n-k}}\ \underset{y}{(0^k)^2}\ \underset{z}{1^n} \rightarrow \notin L.$

$0^{n-k}\, 0^{2k}\, 1^n \Rightarrow 0^{n+k} 1^n$

1) Assume L is regular. Let n be the number of states in the corresponding finite automata.
2) Choose string w such that $|w| \geq n$. Use pumping Lemma to write $w = xyz$ with $|xy| \leq n$ & $|y| > 0$
3) find suitable i such $xy^iz \notin L$.

Pumping Lemma (For Regular Languages) - EXAMPLE (Part-1)

Using Pumping Lemma prove that the language $A = \{a^n b^n \mid n \geq 0\}$ is Not Regular

Proof:

Assume that A is Regular

Pumping length = P

$S = a^P b^P \Rightarrow S = aaaaaaabbbbbbb$

x y z

P = 7

Case 1: The Y is in the 'a' part

P = 7

Case 1: The Y is in the 'a' part

$\underbrace{aa}_{x}\underbrace{aaaa}_{y}\underbrace{abbbbbbb}_{z}$

Case 2: The Y is in the 'b' part

$\underbrace{aaaaaaab}_{x}\underbrace{bbbb}_{y}\underbrace{b}_{z}$

Case 3: The Y is in the 'a' and 'b' part

$\underbrace{aaaaa}_{x}\underbrace{aabb}_{y}\underbrace{bbbbb}_{z}$

$xy^1z \Rightarrow xy^2z$ ✗

aa aaaa aaaa a bbbbbbb

$11 \neq 7$

$xy^1z \Rightarrow xy^2z$ ✗

aaaaaaa bb bbbb bbbb b

$7 \neq 11$

$xy^1z \Rightarrow xy^2z$ ✗

aaaaa aabbaabb bbbbb

$a^n b^n$

$|xy| \leq P$ $P = 7$

Pumping Lemma (For Regular Languages) EXAMPLE (Part-2)

Using Pumping Lemma prove that the language $A = \{yy \mid y \in \{0,1\}^*\}$ is Not Regular

Proof:

$\underline{01}\,\underline{01}$

Assume that A is Regular

then it must have a Pumping Length = P

$S = 0^P 1 0^P 1$

x y z

P = 7

$\underbrace{00}_{x}\underbrace{00000}_{y}\underbrace{0100000001}_{z}$

$xy^iz \Rightarrow xy^2z$

00 0000000000 100000001

$\notin A$

$|y| > 0$

$|xy| \leq P = 7$

A is not Regular

Regular Grammar

Noam Chomsky gave a Mathematical model of Grammar which is effective for writing computer languages

The four types of Grammar according to Noam Chomsky are:

Grammar Type	Grammar Accepted	Language Accepted	Automaton
TYPE-0	Unrestricted Grammar	Recursively Enumerable Language	Turing Machine
TYPE-1	Context Sensitive Grammar	Context Sensitive Language	Linear Bounded Automaton
TYPE-2	Context Free Grammar	Context Free Language	Pushdown Automata
TYPE-3	Regular Grammar	Regular Language	Finite State Automaton

Grammar:

A Grammar 'G' can be formally described using 4 tuples as $G = (V,T,S,P)$ where,

- V = Set of Variables or Non-Terminal Symbols
- T = Set of Terminal Symbols
- S = Start Symbol
- P = Production rules for Terminals and Non-Terminals

  A production rules has the form $\alpha \rightarrow \beta$ where α and β are strings on $V \cup T$ and atleast one symbol of α belongs to V.

Example: $G = (\{S,A,B\}, \{a,b\}, S, \{S \rightarrow AB,\ A \rightarrow a,\ B \rightarrow b\})$

$V = \{S, A, B\}$

$T = \{a, b\}$

$S = S$

$P = S \rightarrow AB,\ A \rightarrow a,\ B \rightarrow b$

Eg. $S \rightarrow AB$

$\rightarrow aB$

$\rightarrow ab$

Regular Grammar:

Regular Grammar can be divided into two types:

Right Linear Grammar	Left Linear Grammar
A grammar is said to be Right Linear if all productions are of the form $A \rightarrow xB$ $A \rightarrow x$ where $A,B \in V$ and $x \in T$	A grammar is said to be Left Linear if all productions are of the form $A \rightarrow Bx$ $A \rightarrow x$ where $A,B \in V$ and $x \in T$

Eg. $S \rightarrow abS \mid b$ – Right linear

$S \rightarrow Sbb \mid b$ – Left linear

Derivations from a Grammar

The set of all strings that can be derived from a Grammar is said to be the LANGUAGE generated from that Grammar

Example 1: Consider the Grammar $G1 = (\{S, A\}, \{a, b\}, S, \{S \to aAb,\ aA \to aaAb,\ A \to \epsilon\})$

$$
\begin{aligned}
S &\to \underline{a A} b && [\text{by } S \to aAb] \\
&\to a\underline{aA}bb && [\text{by } aA \to aaAb] \\
&\to a\,a\underline{aA}bbb && [\text{by } aA \to aaAb] \\
&\to aaabbb && [\text{by } A \to \epsilon]
\end{aligned}
$$

Example 2: $G2 = (\{S, A, B\}, \{a, b\}, S, \{S \to AB,\ A \to a,\ B \to b\})$

$$
\begin{aligned}
S &\to AB \\
&\to ab
\end{aligned}
$$

$L(G2) = \{ab\}$

$\to aaabbb \quad [\text{by } A \to \epsilon]$

Example 2: $G2 = (\{S, A, B\}, \{a, b\}, S, \{S \to AB,\ A \to a,\ B \to b\})$

$$
\begin{aligned}
S &\to AB \\
&\to ab
\end{aligned}
$$

$L(G2) = \{ab\}$

Example 3: $G3 = (\{S, A, B\}, \{a, b\}, S, \{S \to AB,\ A \to aA \mid a,\ B \to bB \mid b\})$

$$
\begin{aligned}
S &\to AB \\
&\to ab
\end{aligned}
\qquad
\begin{aligned}
S &\to AB \\
&\to aAbB \\
&\to aabb
\end{aligned}
\qquad
\begin{aligned}
S &\to AB \\
&\to aAb \\
&\to aab
\end{aligned}
\qquad
\begin{aligned}
S &\to AB \\
&\to abB \\
&\to abb
\end{aligned}
$$

$$
\begin{aligned}
L(G3) &= \{ab,\ a^2b^2,\ a^2b,\ ab^2,\ \ldots\} \\
&= \{a^m b^n \mid m \geq 0 \text{ and } n \geq 0\}
\end{aligned}
$$