ch-9. Regular Languages.

Regular Language

A Language that can be defined by a regular expression is called a regular Language.

Th. If L_1 and L_2 are regular languages, then L_1+L_2, L_1L_2, L_1^* are also regular languages.

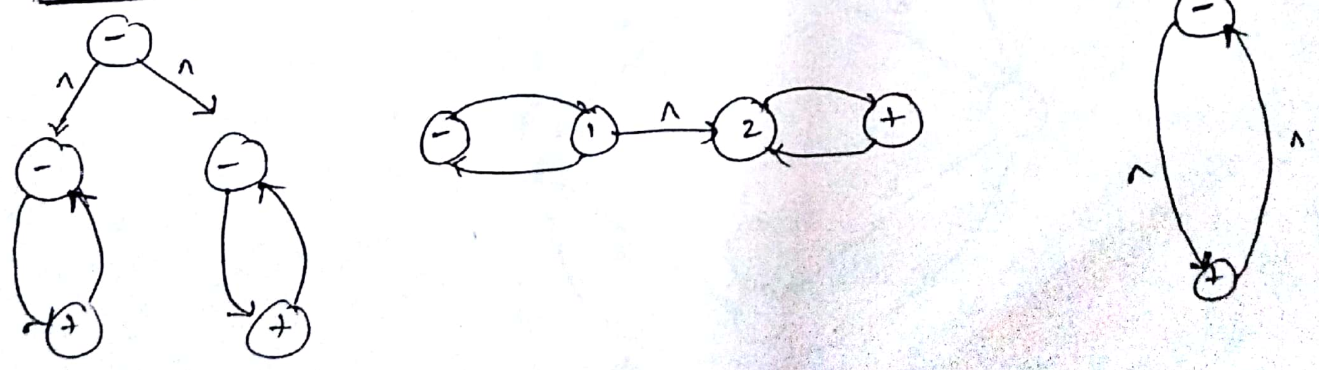
Proof. ⓐ. By Regular Expressions.
ⓑ. By FA machines.

ⓐ. **By Regular Expressions.**

If L_1 & L_2 are regular languages, there are regular expressions r_1 and r_2 that define these languages. Then (r_1+r_2) is a regular ~~Lan~~ Expr. that defines the language L_1+L_2. The language L_1L_2 can be defined by regular expression r_1r_2. The language L_1^* can be defined by the regular Expression $(r_1)^*$. ∴ All three of these sets of words are definable by regular Expressions & so are themselves regular languages.

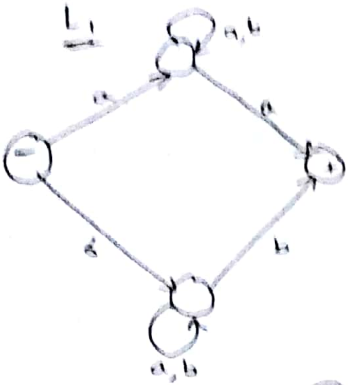
ⓑ. **By Machines.**

L_1+L_2 $\quad$ L_1L_2 $\quad$ L_1^*

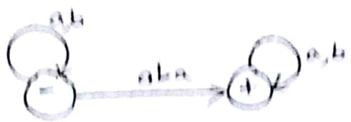


eg 1

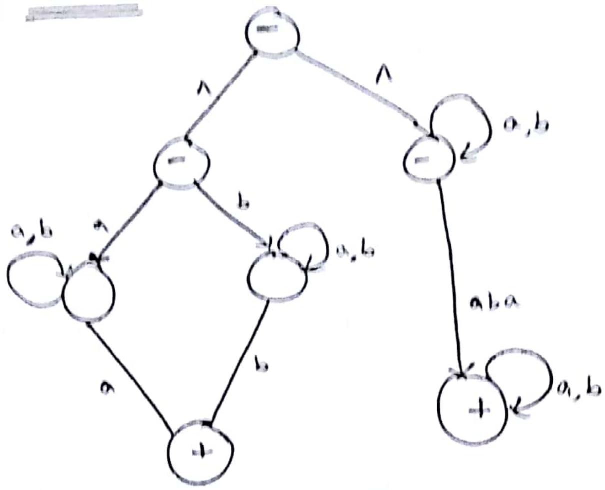
L_1



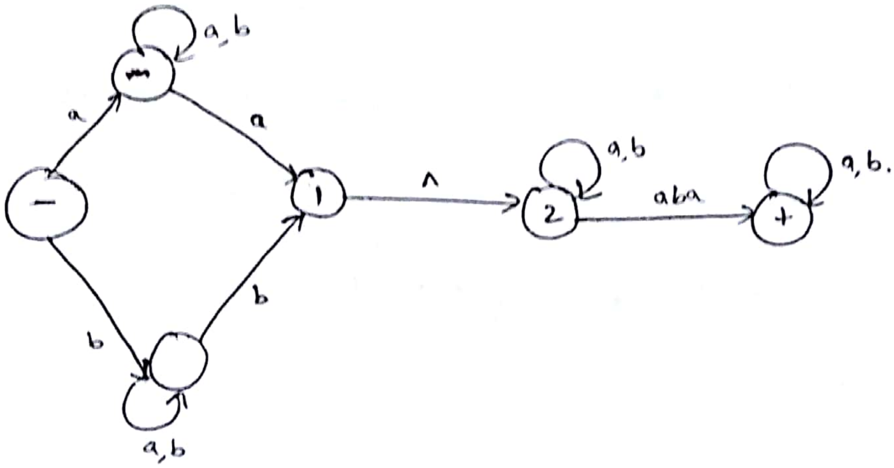
L_2



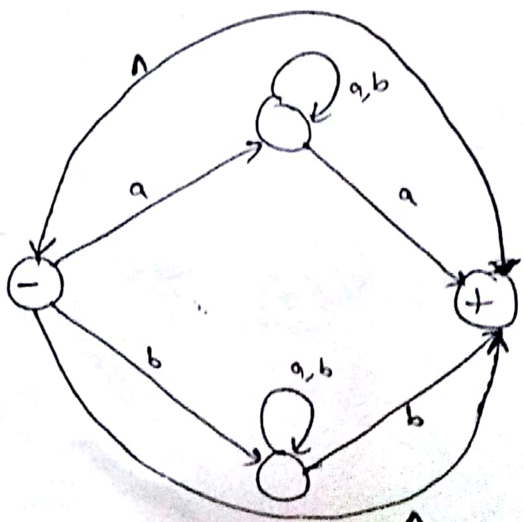
(a) $L_1 + L_2$



eg (b) $L_1 L_2$



(c) L_1^*



ch-10. Non-regular Languages.

Def^n: A lang that cannot be defined by a regular Expressi is called a non-regular language.

eg. $L = \{ \Lambda \quad ab \quad aabb \quad aaabbb \quad aaaabbbb \ldots \}$.

ie $L = \{ a^n b^n \text{ for } n = 0 \; 1 \; 2 \; 3 \; 4 \ldots \}$

for short $L = \{ a^n b^n \}$.

(Theorem 13)

Pumping Lemma.

Pumping Lemma states that there must be strings x, y and z such that all words of the form $xy^n z$ are in L.

$$w = xy^n z \text{ for } n = 1, 2, 3, \ldots$$

1). x is part of w starting at the beginning that lead up to the first state that is revisited. [It can be null string]

2). y denote the substring of w that travels around the circuit coming back to the same state the circuit began with. [It cannot be null].

3). z is rest of w starting with the letter after the substring y & going to the end of string w. (It can be null)

A regular language should satisfy pumping Lemma.

eg. $L = \{ a^n b^n$ for $n = 0, 1, 2, \ldots \}$.

This is not regular & hence Pumping Lema should not be satisfied.

sol^n. A typical word of L Looks Like ..

aaa .... aaaabbbb... bbb.

Break them into three peices comfortable to the roles of

x, y & z.

1). If y is made entirely of 'a's & when we pump it to xyyz, the word will have more a's than b's, which is not allowed.

2). If y is made of 'b's then more b's than a.

3). If y is made of 'ab', then 'xyyz' yeilds more than one pair of ab, which is not allowed.

∴ xyyz cannot be a word in L. This Proves Pumping Lemma cannot apply to L & therefore L is not regular.

eg.

Prove $a^n b a^n = \{b \quad aba \quad aabaa \ldots\}$ is not regular using Pumping Lemma.

Sol^n: observation 1: If the 'y' string contained the b, then $xyyz$ would contain two b's, which no words in this Language can have.

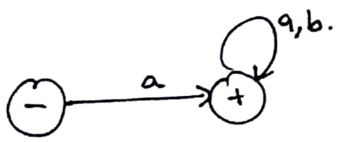
observation 2: If the y string is all a's, then the 'b' in the middle of the word xyz is in the 'x' side or 'z' side.

If either case, $xyyz$ has increased the no. of a's either in front of the b or after the b, but not both.

Conclusion 1: Therefore, $xyyz$ does not have it's 'b' in the middle & is not in the form $a^n b a^n$.

Conclusion 2: This Language cannot be Pumped & is therefore not regular.

eg. $a(a+b)^*$



$w = \{\cancel{\epsilon}\, a, aa, ab, aaa, aab, abb, aba, \ldots\}$

$xy^n z = \epsilon$

1). $x = a$

2). $y = (a+b)^*$

3). $z = \Lambda$.

(Extension of Pumping Lemma Th.) (Reg. Lang)

Th:- Let L be an infinite language accepted by a finite automaton with N states. Then for all words w in L that have more than N letters, there are strings x, y & z, where y is not null & length (x) + length (y) does not exceed N such that $w = xyz$. & all strings of the form

$$xy^nz \text{ (for } n = 1, 2, 3, \ldots) \text{ are in } L.$$

Eg. Show that language 'PALINDROME' is non-Regular

Sol^n. We cannot use Th-13 because $x = a$, $y = b$, $z = a$ satisfy the lemma as $xy^nz = ab^na$ are in PALINDROME. Consider an FA that accept the lang. Suppose it has 77 states. Now the word $a^{80}ba^{80}$ must be accepted by this machine. Break w in three parts. But because length of x & y must be in total 77 or less, they must not be made of solid a's, becoz first 77 letters of w are all a's. That means we form the word $xyyz$, we are adding more a's to the front of w. But we are not adding more a's to the back of w because all the rear a's are in z part, which stays fixed at 80 a's. This means string $xyyz$ is not palindrome because it will be of the form $a^{\text{more than } 80}ba^{80}$.

$\underbrace{aaa\ldots}_{x}\underbrace{aa}_{y}\underbrace{baaa}_{z}$

(Theorem - 35) Extension of Pumping Lemma Th (CFL's) 68

Th. Let L be a CFL in CNF with p live productions. Then any word w in L with length $> 2^p$ can be broken into 5 parts : $w = uvxyz$ such that

length $(vxy) \le 2^p$, length $(x) > 0$, length (v) + length$(y) > 0$

and such that all the words $uv^n xy^n z$ are in the lang. L.

Eg. $L = \{a^n b^m a^n b^m\}$ where $n, m \geq 1$ & independent values

ie $L = \{abab, aabaab, abbabb, aabbaabb, \ldots\}$

If we try to prove that this lang was non-context free using (Th 34), we can write $u = \Lambda$, v = first a's $= a^s$, x = middle b's $= b^t$, y = second a's $= a^s$, z = last b's $= b^t$.

$$uv^n xy^n z = \underbrace{\Lambda}_{u} \underbrace{(a^s)^n}_{v} \underbrace{b^t}_{x} \underbrace{(a^s)^n}_{y} \underbrace{b^t}_{z}$$ all of

which are in L. ∴ we have no contradiction & P. Lema does not apply to L. How, Try P.L with length approach.

If L have a CFG that generates it & it has p live prod.'s, then consider the word $w = a^{2^p} b^{2^p} a^{2^p} b^{2^p}$. This '$w$' has length greater than 2^p (long enough) (acc. to Th 35). Next, we know that, length $(vxy) < 2^p$. So v & y cannot be solid blocks of one letter separated by a clump of the other letter, because the separator letter clump is longer than the length of the whole substring ie length$(x) > 2^p$ (counting substrings of "ab" & "ba"). we see that v & y must be one solid letter, But because of the length condition, all the letters must come from the same clump. Any of the four clump will do.

This means $uvvxyyz$ is not of the form $a^m b^m a^n b^m$ but must also be in L.

∴ L is non-context free.

Eg:- If we have $p = 3$ live productions, then length condition is, any word w_d in L ~~with~~ having length $\text{length}(w) > 2^p$ can be broken in 5 parts

$$w = u\,v\,x\,y\,z \text{ such that}$$

(a) $\text{length}(vxy) \leq 2^p$

(b) $\text{length}(x) > 0$

(c) $\text{length}(v) + \text{length}(y) > 0$.

& all w must be in the form

$$w = u\,v^n\,x\,y^n\,z.$$

Suppose $p = 3$. & w for lang.

$$L = \{ \overset{u}{a^n}\, \overset{v}{b^n}\, \overset{x}{a^m}\, \overset{y}{b^n}\, \overset{z}{a^m},\ n, m \geq 1$$

suppose w is $\Rightarrow$ $w = a^{2^p} b^{2^p} a^{2^p} b^{2^p}$.

$\therefore$ $\text{length}(w) > 2^p$ [long enough]

ie $32 > 8$

Next, condition is, to satisfy & $\text{length}(vxy) \leq 2^p$.

$\text{length}(x) = 8$

$\text{length}(v) = 8$

$\text{length}(y) = 8$.

Here length of separator, ie 'x' is ~~greater than~~ not satisfying & 2^p the condition itself.

So 'v' & 'y' cannot be solid blocks of 'a' [or one letter] separated by solid clumps of another letter.

Scanned by CamScanner

because the separator letter clumps, itself is ~~not equal to or~~ longer than length of the whole substring vxy i.e. length$(x) > 2p$.

Thus v & y must consist of 'ab' or 'ba' which is not permissible when the word is pumped.