Operations on Languages

- **Concatenation** :- Let L_1 & L_2 be language over an alphabet Σ.
Then the concatenation of L_1 & L_2 is denoted by $L_1 \cdot L_2$.

$w_1 = abc \quad w_2 = bca$

$w_1 \cdot w_2 = abcbca$

$w_1 \cdot w_2 \neq w_2 \cdot w_1$

$|w_1| + |w_2| = |w_1 \cdot w_2|$

- **Reverse (w^R)** : Let L be a language over an alphabet Σ the Reversed of L_1 is denoted by L^R.

$w = abba$

$w^R = abba$

$|w| = |w^R|$

- **Prefix and suffix** :- Given two strings a and b, form a new string from these strings by combining the prefix of string a and suffix of string b.

$w = aabca$

$p(w) = \{\varepsilon, a, aa, aab, aabc, aabca\}$ (Prefix)

- Example

String a : remuneration

String b : acquiesce

Length - 5

Output : remuniese

$w = \{aabca\}$

$S(w) = \{\epsilon, a, ca, bca, abca, aabca\}$ (Suffix)

JPNotes

- **Substring** :- If 'w' is a string then 'v' is substring of 'w' if there exists string x and y such that $w = xvy$

Example

$|w| = abbcd$

substring ?

w_1 = abbc ✓ Yes
w_2 = bcd ✓ Yes
w_3 = cba ✗ NO
w_4 = abc ✗ NO
w_5 = bc ✓ yes

JPwebdevelopers

* All prefix and suffix are substring but not vice versa.

- **Power of Σ** :- In TOC the Power of Sigma refers to the Computational strength of the alphabet Σ and the lang. that can be construct over the alphabet.

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

minimum Power → $\Sigma^0 = \{\epsilon\}$ → set of all string with length '0'

$\Sigma^1 = \{a, b\}$ ← set of all string with length 1

$\Sigma^2 = \{aa, ab, bb, ba\}$

⋮

$\Sigma^k = \{w \mid |w| = k\}$

⋮

$\Sigma^\infty = \{w \mid |w| = \infty\}$

Imp

Σ^* (kleene closure) :- It is a set of all strings of all length possible on a and b.

$(a+b)^*$ Infinite language

Σ^+ = Positive closure (all string possible except ϵ)

$\boxed{\Sigma^* - \epsilon = \Sigma^+}$

- **Union**

If L1 and L2 are two regular languages, then union L1 ∪ L2 will also be regular

$L1 = (1+0) \cdot (1+0)$

$= \{11, 10, 01, 00\}$

$L2 = \{\epsilon, 100\}$

then $L1 \cup L2 = \{\epsilon, 00, 10, 11, 01, 100\}$

- **Intersection**

If L1 and L2 are two regular languages, their intersection L1 ∩ L2 will also be regular.

$L1 = \{0, 1\}$

and

$L2 = \{0, 01, 11, 10\}$

$L1 \cap L2 = \{0\}$

* **Symmetric Difference**

$L1 \oplus L2$

$= (L1 \cup L2) - (L1 \cap L2)$

- **Complement**

$L^c = \Sigma^* - L$

- **De Morgan's law**

let L1, L2, L3 be any three languages then $L1 \cdot (L2 \cap L3) = (L1 \cdot L2) \cup (L1 \cdot L3)$