Welcome!!!!!!!!!

LECTURE # 3

Muhammad Hafeez Javed
Foundation University Rawalpindi
rmhjaved@gmail.com
www.rmhjaved.com

Recap Lecture-2

- Kleene Star Closure, Plus operation, recursive definition of languages, INTEGER, EVEN, factorial, PALINDROME, **{a^nb^n}**, languages of strings (i) ending in a, (ii) beginning and ending in same letters, (iii) containing aa or bb (iv)containing exactly aa,

Task

- Q)

1) Let S={ab, bb} and T={ab, bb, bbbb} Show that $S^* = T^*$ [Hint $S^* \subseteq T^*$ and $T^* \subseteq S^*$]

2) Let S={ab, bb} and T={ab, bb, bbb} Show that $S^* \neq T^*$ But $S^* \subset T^*$

   Solution: Since $S \subset T$, so every string belonging to S^* , also belongs to T^* but bbb is a string belongs to T^* but does not belong to S^*.

- 3) Let $S=\{a, bb, bab, abaab\}$ be a set of strings. Are abbabaabab and baabbbabbaabb in S^*? Does any word in S^* have odd number of b's?

  Solution: since abbabaabab can be grouped as (a)(bb)(abaab)ab , which shows that the last member of the group does not belong to S, so abbabaabab is not in S^*, while baabbbabbaabb can not be grouped as members of S, hence baabbbabbaabb is not in S^*. Since each string in S has even number of b's so there is no possiblity of any string with odd number of b's to be in S^*.

Task

Q1)Is there any case when S^+ contains Λ? If yes then justify your answer.

Solution: consider $S=\{\Lambda,a\}$ then

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

Here Λ is in S^+ as member of S.

Thus Λ will be in S^+, in this case.

Q2) Prove that for any set of strings S

i. $(S^+)^* = (S^*)^*$

Solution: In general Λ is not in S^+, while Λ does belong to S^*. Obviously Λ will now be in $(S^+)^*$, while $(S^*)^*$ and S^* generate the same set of strings. Hence $(S^+)^* = (S^*)^*$.

Q2) continued…

ii) $(S^+)^+ = S^+$

Solution: since S^+ generates all possible strings that can be obtained by concatenating the strings of S, so $(S^+)^+$ generates all possible strings that can be obtained by concatenating the strings of S^+, will not generate any new string.

Hence $(S^+)^+ = S^+$

Q2) continued…

iii) Is $(S^*)^+=(S^+)^*$

Solution: since Λ belongs to S^* ,so Λ will belong to $(S^*)^+$ as member of S^* .Moreover Λ may not belong to S^+, in general, while Λ will automatically belong to $(S^+)^*$.

Hence $(S^*)^+=(S^+)^*$

Regular Expression

- As discussed earlier that a^* generates

  Λ, a, aa, aaa, …

  and a^+ generates a, aa, aaa, aaaa, …, so the language L_1 = {Λ, a, aa, aaa, …} and L_2 = {a, aa, aaa, aaaa, …} can simply be expressed by a^* and a^+, respectively.

  a^* and a^+ are called the regular expressions (RE) for L_1 and L_2 respectively.

  Note: a^+, aa^* and a^*a generate L_2.

Recursive definition of Regular Expression(RE)

Step 1: Every letter of Σ including Λ is a regular expression.

Step 2: If r_1 and r2 are regular expressions then

1. (r_1)
2. $r_1 r_2$
3. $r_1 + r_2$ and
4. r_1^*

are also regular expressions.

Step 3: Nothing else is a regular expression.

Defining Languages (continued)…

- **Method 3 (Regular Expressions)**
  - Consider the language L={Λ, x, xx, xxx,…} of strings, defined over Σ = {x}.

    We can write this language as the Kleene star closure of alphabet Σ or $L=\Sigma^*=\{x\}^*$

    this language can also be expressed by the regular expression x^*.
  - Similarly the language L={x, xx, xxx,…}, defined over Σ = {x}, can be expressed by the regular expression x^+.

- Now consider another language L, consisting of all possible strings, defined over $\Sigma = \{a, b\}$. This language can also be expressed by the regular expression

  $(a + b)^*$.

- Now consider another language L, of strings having exactly double a, defined over $\Sigma = \{a, b\}$, then it’s regular expression may be

  b^*aab^*

- Now consider another language L, of even length, defined over Σ = {a, b}, then it's regular expression may be

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

- Now consider another language L, of odd length, defined over Σ = {a, b}, then it's regular expression may be

  $(a+b)((a+b)(a+b))^*$ or

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

Remark

- It may be noted that a language may be expressed by more than one regular expressions, while given a regular expression there exist a unique language generated by that regular expression.

- Example:
  - Consider the language, defined over $\Sigma=\{a , b\}$ of words having at least one a, may be expressed by a regular expression $(a+b)^*a(a+b)^*$.
  - Consider the language, defined over $\Sigma = \{a, b\}$ of words having at least one a and one b, may be expressed by a regular expression

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

- Consider the language, defined over $\Sigma=\{a, b\}$, of words starting with double a and ending in double b then its regular expression may be $aa(a+b)^*bb$
- Consider the language, defined over $\Sigma=\{a, b\}$ of words starting with a and ending in b OR starting with b and ending in a, then its regular expression may be $a(a+b)^*b+b(a+b)^*a$

TASK

- Consider the language, defined over $\Sigma=\{a, b\}$ of **words beginning with a**, then its regular expression may be $a(a+b)^*$

- Consider the language, defined over $\Sigma=\{a, b\}$ of **words beginning and ending in same letter**, then its regular expression may be $(a+b)+a(a+b)^*a+b(a+b)^*b$

TASK

- Consider the language, defined over

  Σ={a, b} of **words ending in b**, then its regular expression may be $(a+b)^*b$.
- Consider the language, defined over

  Σ={a, b} of **words not ending in a**, then its regular expression may be $(a+b)^*b + \Lambda$. It is to be noted that this language may also be expressed by $((a+b)^*b)^*$.

SummingUP Lecture 3

RE, Recursive definition of RE, defining languages by RE, $\{ x\}^*$, $\{ x\}^+$, $\{a+b\}^*$, Language of strings having **exactly one aa**, Language of strings of **even length**, Language of strings of **odd length**, RE defines unique language (as Remark), Language of strings having **at least one a**, Language of strings havgin **at least one a and one b**, Language of strings **starting with aa and ending in bb,** Language of strings **starting with and ending in different letters**.