Recap 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**.

Task

- Determine the RE of the language, defined over $\Sigma=\{a, b\}$ of **words beginning with a**.

Solution:

The required RE may be $a(a+b)^*$

- Determine the RE of the language, defined over $\Sigma=\{a, b\}$ of **words beginning with and ending in same letter.**

Solution:

The required RE may be $(a+b)+a(a+b)^*a+b(a+b)^*b$

Task Continued …

- Determine the RE of the language, defined over $\Sigma=\{a, b\}$ of **words ending in b.**

Solution:

The required RE may be

$(a+b)^*b$.

- Determine the RE of the language, defined over $\Sigma=\{a, b\}$ of **words not ending in a.**

Solution: The required RE may be

$(a+b)^*b + \Lambda$ Or $((a+b)^*b)^*$

An important example

The Language EVEN-EVEN :

Language of strings, defined over Σ={a, b} having **even number of a's and even number of b's**. *i.e.*

EVEN-EVEN = {Λ, aa, bb, aaaa,aabb,abab, abba, baab, baba, bbaa, bbbb,…} ,

its regular expression can be written as

$(aa+bb+(ab+ba)(aa+bb)^*(ab+ba))^*$

Note

- It is important to be clear about the difference of the following regular expressions

$r_1=a^*+b^*$

$r_2=(a+b)^*$

Here r_1 does not generate any string of concatenation of a and b, while r_2 generates such strings.

Equivalent Regular Expressions

- **Definition:**

Two regular expressions are said to be equivalent if they generate the same language.

Example:

Consider the following regular expressions

$r_1 = (a + b)^* (aa + bb)$

$r_2 = (a + b)^* aa + (a + b)^* bb$ then

both regular expressions define the language of strings **ending in aa or bb**.

Note

- If r_1 =(aa + bb) and r_2=(a + b) then
  1. r_1+r_2 =(aa + bb) + (a + b)
  2. r_1r_2 =(aa + bb) (a + b)
     =(aaa + aab + bba + bbb)
  3. $(r_1)^*$ =$(aa + bb)^*$

Regular Languages

- **Definition**:

  The language generated by any regular expression is called a **regular language**.

  It is to be noted that if r_1, r_2 are regular expressions, corresponding to the languages L_1 and L_2 then the languages generated by $r_1 + r_2$, $r_1 r_2$(or $r_2 r_1$) and r_1^*(or r_2^*) are also regular languages.

Note

- It is to be noted that if L_1 and L_2 are expressed by $r_1$and r_2, respectively then the language expressed by

  1) $r_1+ r_2$, is the language $L_1 + L_2$ or $L_1 \cup L_2$

  2) r_1r_2, , is the language L_1L_2, of strings obtained by prefixing every string of L_1 with every string of L_2

  3) r_1^*, is the language L_1^*, of strings obtained by concatenating the strings of L, including the null string.

Example

- If r_1=(aa+bb) and r_2=(a+b) then the language of strings generated by r_1+r_2, is also a regular language, expressed by (aa+bb)+(a+b)
- If r_1=(aa+bb) and r_2=(a+b) then the language of strings generated by r_1r_2, is also a regular language, expressed by (aa+bb)(a+b)
- If r=(aa+bb) then the language of strings generated by r^*, is also a regular language, expressed by $(aa+bb)^*$

All finite languages are regular.

Example:

Consider the language L, defined over Σ={a,b}, of strings of length 2, **starting with a**, then

L={aa, ab}, may be expressed by the regular expression aa+ab. Hence L, by definition, is a regular language.

Note

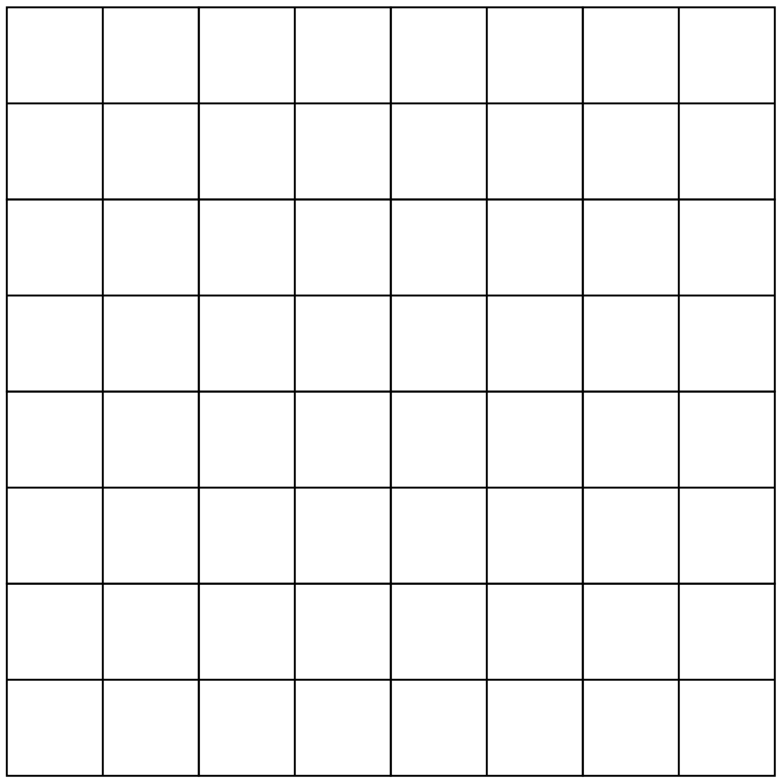
It may be noted that if a language contains even thousand words, its RE may be expressed, placing ‘ + ’ between all the words.

Here the special structure of RE is not important.

Consider the language L={aaa, aab, aba, abb, baa, bab, bba, bbb}, that may be expressed by a RE aaa+aab+aba+abb+baa+bab+bba+bbb, which is equivalent to (a+b)(a+b)(a+b).

Introduction to Finite Automaton

- Consider the following game board that contains 64 boxes



Finite Automaton Continued …

There are some pieces of paper. Some are of white colour while others are of black color. The number of pieces of paper are 64 or less. The possible arrangements under which these pieces of paper can be placed in the boxes, are finite. To start the game, one of the arrangements is supposed to be initial arrangement. There is a pair of dice that can generate the numbers 2,3,4,…12 . For each number generated, a unique arrangement is associated among the possible arrangements.

Finite Automaton Continued …

It shows that the total number of transition rules of arrangement are finite. One and more arrangements can be supposed to be the winning arrangement. It can be observed that the winning of the game depends on the sequence in which the numbers are generated. This structure of game can be considered to be a finite automaton.

Defining Languages (continued)…

- Method 4 (Finite Automaton)
  Definition:
  A Finite automaton (FA), is a collection of the followings
  1) Finite number of states, having one initial and some (maybe none) final states.
  2) Finite set of input letters (Σ) from which input strings are formed.
  3) Finite set of transitions *i.e.* for each state and for each input letter there is a transition showing how to move from one state to another.

Example

- $\Sigma = \{a,b\}$
- **States:** x, y, z where x is an initial state and z is final state.
- **Transitions:**

1. At state **x** reading **a** go to state **z**,
2. At state **x** reading **b** go to state **y**,
3. At state **y** reading **a**, **b** go to state **y**
4. At state **z** reading **a**, **b** go to state **z**

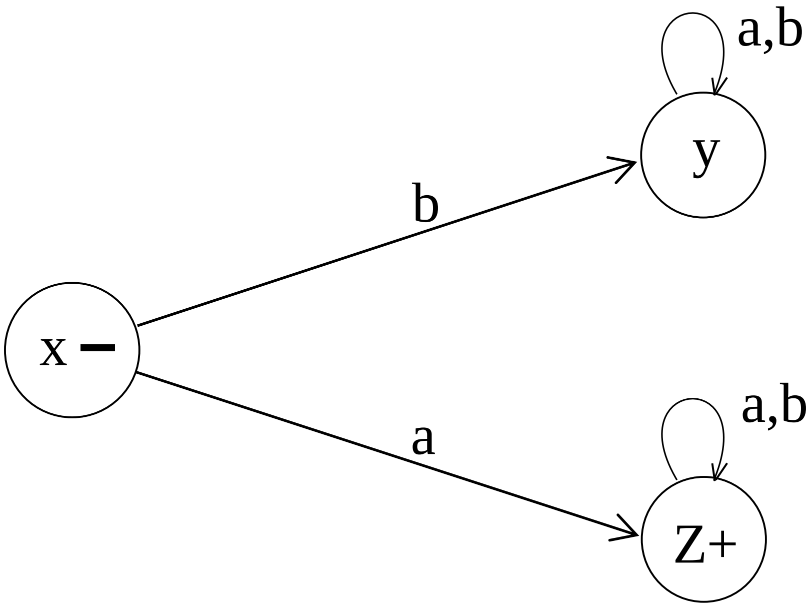
Example Continued …

⌘ These transitions can be expressed by the following table called transition table

Old States	New States	
	Reading a	Reading b
x -	z	y
y	y	y
z +	z	z

Note

- It may be noted that the information of an FA, given in the previous table, can also be depicted by the following diagram, called the **transition diagram**, of the given FA



Remark

- The previous transition diagram is an FA accepting the language of strings, defined over Σ={a, b}, **starting with a**. It may be noted that this language may be expressed by the regular expression

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

Summing Up

- ⌘ Regular expression of **EVEN-EVEN** language, Difference between a* + b* and (a+b)*, Equivalent regular expressions; sum, product and closure of regular expressions; regular languages, finite languages are regular, introduction to finite automaton, definition of FA, transition table, transition diagram