Theory of Automata

Welcome

Class: BS(VI)

Sukkur IBA University, Kandhkot Campus

Week – 01

Lecture – 01

Your Facilitator, Muhammad Haris

Marks Distribution

Theory	Marks
Mid Term	30 Marks
Sessional	20 Marks
Final	50 Marks
Total	**100 Marks**

Recommended Books

- Introduction to Computer Theory (Automata Theory)

  By Daniel I. A. Cohen
- Introduction to the Theory of Computation

  By Michael Sipser (3rd Edition)
- Introduction to Automata Theory, Languages, and Computations

  By John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman-

About Course

What is Automata?

- It is the plural of **automaton**, and it means "something that works automatically"
- Automatic Machine/ Self Controlled
  - Computer, ATM, any machine that has automatic mechanism
  - Computer is invented on this basic rule.

Why we study this subject?

- It allows us to think systematically about what machine do without going into hardware details.
- Learning of languages.
  - Computer languages are based on this principle.
  - Computer is machine needs language to communicate to human.
  - So we need letters, alphabets, strings, words in language.
- Designing of theoretical models for machines.
- Core subject of Computer Science

Introduction to Languages

Introduction to Languages

- There are two types of languages

1. Formal Languages (Syntactic languages)
   - Is normally defined by an alphabet and formation rules.
   - The alphabet of a **formal language** is a set of symbols on which this **language** is built.
   - Some of the symbols in an alphabet may have a special meaning.
2. Informal Languages (Semantic languages)
   - We can not communicate with computer by informal languages like English so we need computer languages so we use some symbols like alphabets.

Alphabets

- Definition:

  A finite non-empty set of symbols (letters), is called an alphabet.

  It is denoted by Σ (Greek letter sigma).

- Example:

  Σ={a,b} //a, b are alphabets

  Σ={0,1} //important as this is the language

  //which the computer understands.

  Σ={i,j,k}

Note

- A certain version of language ALGOL has 113 letters

Σ (alphabet) includes letters, digits and a variety of operators including sequential operators such as GOTO and IF

String

- Definition:

  Concatenation of finite symbols from the alphabet is called a string.

- Example:

  If Σ= {a,b} then

  a, abab, aaabb, abababababababababab

- **Strings** not necessary have meaning so called string while **word** have some meaning in languages.

Note

EMPTY STRING or NULL STRING

- Sometimes a string with no symbol at all is used, denoted by
  - (Small Greek letter Lambda) λ or
  - (Capital Greek letter Lambda) Λ, is called an empty string or null string.
- The capital lambda Λ will mostly be used to denote the empty string, in further discussion.

Words

- Definition:

  Words are strings belonging to some language.

Example:

If Σ= {x} then a language L can be defined as

$L=\{x^n : n=1,2,3,.....\}$ or L={x,xx,xxx,....}

Here x,xx,… are the **words** of L

Note: All strings are not words but all words are strings.

Valid/ Invalid Alphabets

- Suppose we can have the language:
- While defining an alphabet, an alphabet may contain letters consisting of group of symbols for example Σ_1= {B, aB, bab, d}.

Valid/ Invalid Alphabets

- Now consider an alphabet Σ_2= {B, Ba, bab, d} and a string BababB.

This string can be **tokenized** in two different ways

- (Ba), (bab), (B)
- (B), (abab), (B)

Which shows that the second group cannot be identified as a string, defined over Σ = {a, b}.

C.W: Write two valid and two invalid strings

- When this string is scanned by the compiler (Lexical Analyzer), first symbol B is identified as a letter belonging to Σ, while for the second letter the lexical analyzer would not be able to identify, so while defining an alphabet it should be kept in mind that ambiguity should not be created.

Remarks

- While defining an alphabet of letters consisting of more than one symbols, no letter should be started with the letter of the same alphabet *i.e.* one letter should not be the prefix of another. However, a letter may be ended in the letter of same alphabet *i.e.* one letter may be the suffix of another.

Conclusion

- Σ_1 = {B, aB, bab, d}
- Σ_2 = {B, Ba, bab, d}

Σ_1 is a valid alphabet while

Σ_2 is an in-valid alphabet.

Length of Strings

- Definition:

  The length of string s, denoted by |s|, is the number of letters in the string.

- Example:

  Σ={a,b}

  s=ababa

  |s|=5

Length of Strings

- Example:

  Σ= {B, aB, bab, d}

  s=BaBbabd

Strings can be **tokenized** like

Tokenizing=(B), (aB), (bab), (d)

|s|=4

Reverse of a String

- Definition:

  The reverse of a string s denoted by Rev(s) or s^r, is obtained by writing the letters of s in reverse order.

- Example:

  If s=abc is a string defined over Σ={a,b,c}

  then Rev(s) or s^r = cba

Reverse of a String

- Example:

  Σ= {B, aB, bab, d}

  s=BaBbabBd

(First tokenize then make strings reverse)

--(B)(aB)(bab)(B)(d) no need to reverse words like aB will remain aB not Ba. Its Better to tokenize a string.

--So if we have Baa then reverse is same Baa not aaB

Rev(s)=dBbabaBB

How to create languages?

- Spoken languages like English have all words with some meaning so they are defined.
- But languages we are talking about are computer languages. Used to communicate computer.
- We have created ourselves not naturally evaluated
- So we define rules ourselves.
- We can define language by applying some conditions.

Defining Languages

- The languages can be defined in **different** ways , such as

1. Descriptive definition,
2. Recursive definition, using
   I. Regular Expressions(RE) and
   II. using Finite Automaton(FA) etc.

Descriptive definition of language:

The language is defined, describing the conditions imposed on its words.

Examples: Defining Languages

- Example:

  The language L of strings of odd length, defined over Σ={a}, can be written as

  L={a, aaa, aaaaa,.....}

  - For Language L we used Curly brackets
  - Strings are separated by comma
  - Its basically a set
  - Dots in the end showing it has infinite number of words/strings but finite in word length.

- Example:

  The language L of strings that does not start with a, defined over Σ={a,b,c}, can be written as

  L={b, c, ba, bb, bc, ca, cb, cc, ...}

  - When we write any language in tabular form
  - Then we used in a systemic way like single, double and triple letter strings

Examples: Defining Languages

- Example:

  The language L of strings of length 2, defined over Σ={0,1,2}, can be written as

  L={00, 01, 02,10, 11,12,20,21,22}

  - First we have written words in systematic order starting with length o then 1 and then 2

- Example:

  The language L of strings ending in 0, defined over Σ ={0,1}, can be written as

  L={0,00,10,000,010,100,110,…}

  - Half strings starts/ends with 0 and half with 1
  - So condition on string length matters in similar way on starting or ending string.

Examples: Defining Languages

- Example: The language **EQUAL**, of strings with number of a's equal to number of b's, defined over Σ={a,b}, can be written as

  {Λ ,ab,aabb,abab,baba,abba,…}

  - Λ lambda have no b's and no a's both are zero so called EQUAL
  - No three letters words can satisfy this language

- Example: The language **EVEN-EVEN**, of strings with even number of a's and even number of b's, defined over Σ={a,b}, can be written as

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

  - In tabular form write enough words so than one can understand the condition applied on language
  - Like here we have given four words of length four then use dots.
  - Number of a's and no b's are 0, means **zero** is even.

Examples: Defining Languages

- Example: The language **INTEGER**, of strings defined over Σ={-,0,1,2,3,4,5,6,7,8,9}, can be written as

  INTEGER = {…,-2,-1,0,1,2,…}

-2, -1 are strings of two letters

- Example: The language **EVEN**, of stings defined over Σ={-,0,1,2,3,4,5,6,7,8,9}, can be written as

  EVEN = { …,-4,-2,0,2,4,…}

Examples: Defining Languages

- Example: The language $\{a^n b^n\}$, of strings defined over Σ={a,b}, as $\{a^n b^n : n=1,2,3,\ldots\}$, can be written as
  {ab, aabb, aaabbb,aaaabbbb,…}
  - If n=0 then $(a^n b^n)$ is Λ.

- Example: The language $\{a^n b^n a^n\}$, of strings defined over Σ={a,b}, as $\{a^n b^n a^n : n=1,2,3,\ldots\}$, can be written as
  {aba, aabbaa, aaabbbaaa,aaaabbbbaaaa,…}

If n=1,2,3 then first string will be a^0 equals to Null/lemda/Λ

Examples: Defining Languages

- Example: The language **FACTORIAL**, of strings defined over Σ={1,2,3,4,5,6,7,8,9} *i.e.*

  {1,2,6,24,120,…}

- Example: The language **FACTORIAL**, of strings defined over Σ={a}, as

  $\{a^{n!} : n=1,2,3,\ldots\}$, can be written as

  {a,aa,aaaaaa,…}. It is to be noted that the language FACTORIAL can be defined over any single letter alphabet.

Factorial is always of positive integers.

If n=1,2,3 then factorial of 1 is 1 and factorial of 0 is also 1.

Examples: Defining Languages

- Example: The language **DOUBLEFACTORIAL**, of strings defined over Σ={a, b}, as

  $\{a^{n!}b^{n!} : n=1,2,3,\ldots\}$, can be written as

  {ab, aabb, aaaaaabbbbbb,…} we can not write a^2 or b^2 because it is a string and strings are concatenation of letters and letters does not used square/cube or such notations in words. we cant write word (committee) like ($com^2it^2e^2$). So put all a's first then place b's.

- Example: The language **SQUARE**, of strings defined over Σ={a}, as

  $\{a^{n_2} : n=1,2,3,\ldots\}$, can be written as

  {a, aaaa, aaaaaaaaa,…}

Examples: Defining Languages

- Example: The language **DOUBLESQUARE**, of strings defined over Σ={a,b}, as

  $\{a^{n^2} b^{n^2} : n=1,2,3,\ldots\}$, can be written as

  {ab, aaaabbbb, aaaaaaaaabbbbbbbbb,…}

Examples: Defining Languages

- Example: The language **PRIME**, of strings defined over Σ={a}, as

  $\{a^p : p \text{ is prime}\}$, can be written as

  {aa,aaa,aaaaa,aaaaaaa,aaaaaaaaaaa…}

-Prime means numbers divisible by one and itself

An Important language

- **PALINDROME**:

  The language consisting of Λ and the strings s defined over Σ such that Rev(s)=s.

  It is to be denoted that the words of PALINDROME are called palindromes.

- Example:For Σ={a,b},

  PALINDROME={Λ , a, b, aa, bb, aaa, aba, bab, bbb, ...}

  Note: Total 8 double letter words only 4 satisfy above condition

Remark

- There are as many palindromes of length 2n as there are of length 2n-1.

  To prove the above remark, the following is to be noted:

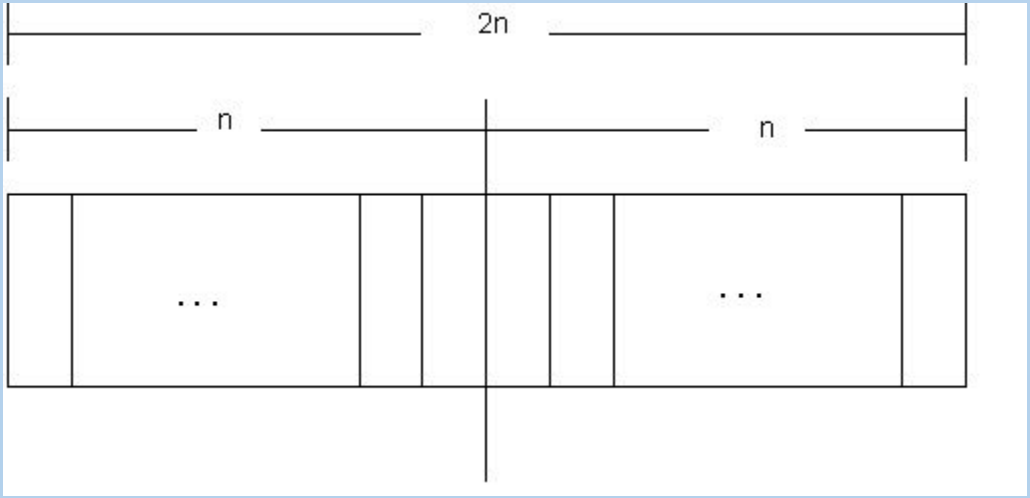
Note

- If string/word length is 3 represented by m.
  - Example aaa, aba, abb
- If alphabets length is 2 denoted by n.
  - Example Σ={a,b}
- $(n)^m = 2^3 = 8$
- It means there will be total 8 words/strings of length 3 in a language have 2 alphabets i.e a,b
- **Find words/strings of length 2 in a language have 3 alphabets i.e** Σ={a,b,c}

Determine String Length

- Number of strings of length ‘m’ defined over alphabet of ‘n’ letters is n^m.
- Examples:
  - The language of strings of length 2, defined over Σ={a,b} is L={aa, ab, ba, bb} *i.e.* number of strings = 2^2=4
  - The language of strings of length 3, defined over Σ={a,b} is L={aaa, aab, aba, baa, abb, bab, bba, bbb} *i.e.* number of strings = 2^3=8
  - The language of strings of length 2, defined over Σ={a,b,c} is L={aa,ab,ac,bb,ba,bc,ca, cb,cc} i.e. number of strings = 3^2=9

- To calculate the number of palindromes of length(2n), consider the following diagram,



Left Half Right Half

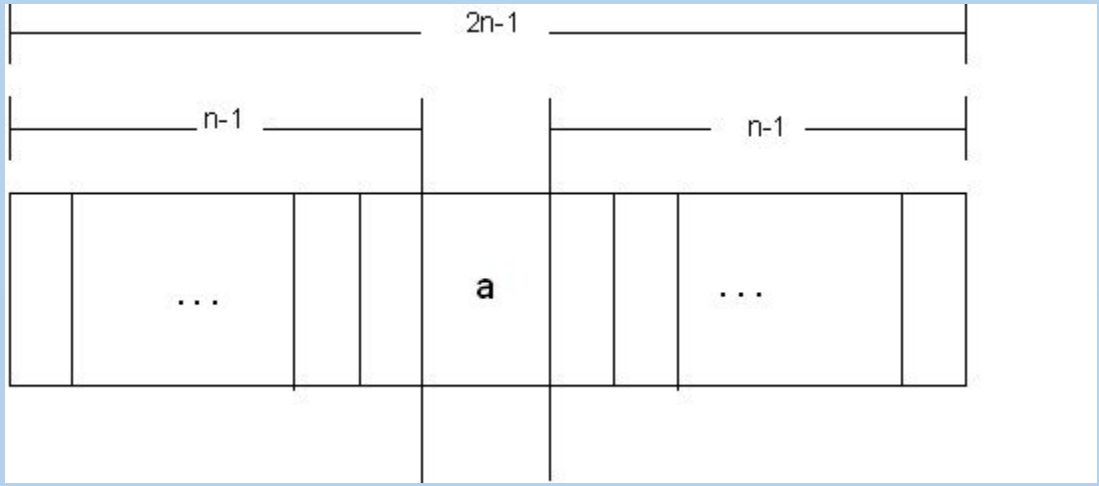
Palindromes of Even Length (i.e 2n)

- If string have length 2n(even length string) can be divided into two equal parts
- It will be symmetric to a line
- Example:(How to make palindromes?)
  - String of length n like *ab*
  - Write *ab* in left half
  - Write reverse of ab in right half
  - Like
    - *ab ba* i.e *abba*
- We know:
  - n length strings defined over alphabets of 2 letters is 2n
  - So that total number of palindromes of length 2n will be 2^n

which shows that there are as many palindromes of length 2n as there are the strings of length n *i.e.* the required number of palindromes are 2^n.

Where 2n Show even numbers and n=0,1,2,3

- To calculate the number of palindromes of length (2n-1) with ‘a’ as the middle letter, consider the following diagram,



Palindromes of Odd Length (i.e 2n-1)

- For two letters language like ∑ = {a,b} we have two possibilities
  - Either “a” or “b” should be the middle letter
- It will be **symmetric to any letter** but not line.
- If string is of length 2n-1
- Then after reducing 1 letter (i.e middle letter a or b)
- On both sides (left half and right half) the spaces to be filled are n-1

- (n-1) | a | (n-1) 2^{n-1}
- (n-1) | b | (n-1) 2^{n-1}
- So add both values
  - $2^{n-1} + 2^{n-1} = 2(2^{n-1}) = 2^n$
- Example: The number of palindromes of length 2 and length (2-1) are equal.
- Hence proved the number of palindromes of length 2n and length (2n-1) are equal.

which shows that there are as many palindromes of length 2n-1 as there are the strings of length n-1 *i.e.* the required number of palindromes are 2^{n-1}.

Similarly the number of palindromes of length 2n-1, with ‘ b ’ as middle letter, will be 2^{n-1} as well. Hence the total number of palindromes of length 2n-1 will be $2^{n-1} + 2^{n-1} = 2\,(2^{n-1}) = 2^n$.

Proof…

- 2n means **even** numbers
- 2n-1 means **odd** numbers
- n=1,2,3 but not 0 in odd numbers b/c length will not be negative.
- Even and one less odd number have same number of palindromes
  - 2,1 (2n=2n-1; if n=1)
  - 4,3 (2n=2n-1; if n=2)
  - 6,5 (2n=2n-1; if n=3) have same number of palindromes
- Example: ∑ = {a,b} have words of length two and one less than two(i.e 1) have same palindromes
  - Palindromes of length 02 are 2 i.e aa,bb (Even Palindrome)
  - Palindromes of length 01 are 2 i.e a,b (Odd Palindrome)
  - C.W find 4,3 and 6,5

Exercise

- Q) Prove that there are as many palindromes of length 2n, defined over $\Sigma = \{a,b,c\}$, as there are of length 2n-1. Determine the number of palindromes of length 2n defined over the same alphabet as well.

Summing-up Lecture-1

- Introduction to the course title,
  - Formal and In-formal languages,
  - Alphabets,
  - Strings, Null string,
  - Words, Valid and In-valid alphabets,
  - length of a string,
  - Reverse of a string,
  - Defining languages,
  - Descriptive definition of languages,
  - EQUAL, EVEN-EVEN, INTEGER, EVEN, $\{ a^n b^n \}$, $\{ a^n b^n a^n \}$,
  - FACTORIAL, DOUBLEFACTORIAL, SQUARE, DOUBLESQUARE, PRIME, PALINDROME.

Thank you!

Any Questions???