Lecture # 1

(Automata Theory)

Material / Resources

- Text Books

1. *Introduction to Computer Theory*, by Daniel I. Cohen, John Wiley and Sons, Inc., 2005, Second Edition
2. *Introduction to Languages and Theory of Computation*, by J. C. Martin, McGraw Hill Book Co., 2003, Third Edition.

- www
- Any other good book on Automata Theory

Grading

- Assignments 10%
- Quizzes 10%
- Mid Term.. 25%
- Final Exam... 50%
- Class Contribution 5%

Theory of Automata

What does Automata mean?

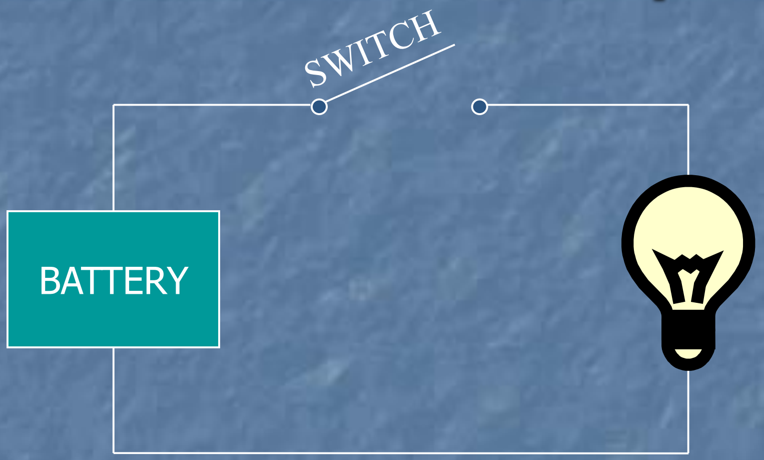
- It is the plural of automaton, and it means "something that works automatically".
- Automata theory is the study of abstract computational devices and the computational problems that can be solved using them.
- Abstract devices are (simplified) models of real computations.

- 1- Automata and formal language.
- Which answers - What are computers (Or what are models of computers)
- 2- Compatibility.
- Which answers - What can be computed by computers?
- 3- Complexity.
- Which answers - What can be efficiently computed?

- In automata we will simulates parts of computers.

- Helps in design and construction of different software's and what we can expect from our software's.
- Automata play a major role in theory of computation, compiler design, artificial intelligence.

A simple computer



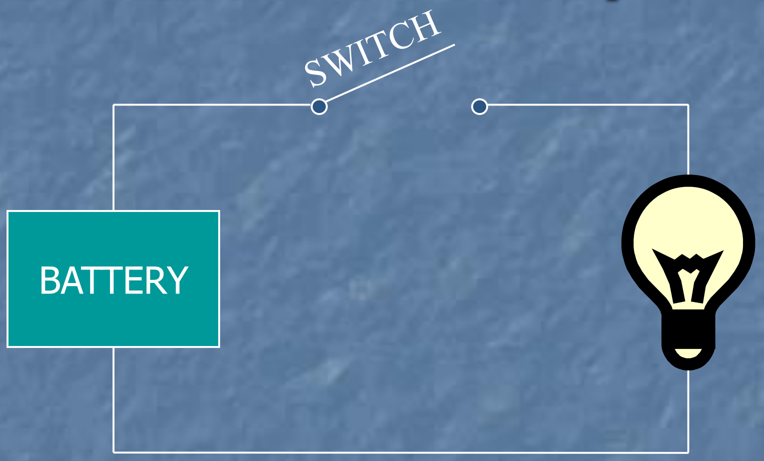
input: switch

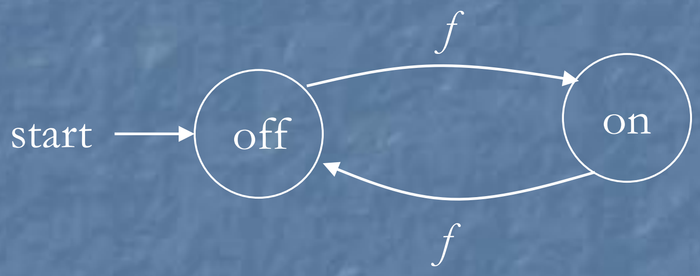
output: light bulb

actions: flip switch

states: on, off

A simple “computer”





input: switch

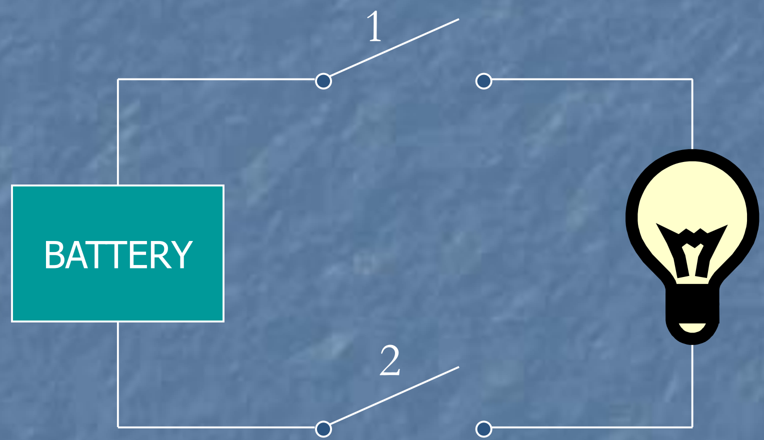
output: light bulb

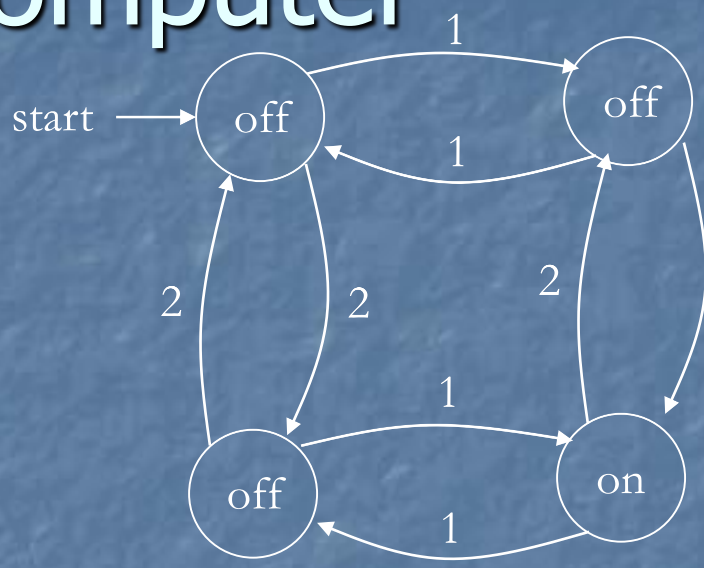
actions: f for “flip switch”

states: on, off

bulb is on if and only if there was an odd number of flips

Another "computer"





inputs: switches 1 and 2

actions: 1 for "flip switch 1"
2 for "flip switch 2"

states: on, off

bulb is on if and only if both switches were flipped an odd number of times

Introduction to languages

Kinds of languages:

- Talking language
- Programming language
- Formal Languages (Syntactic languages)

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}

  Σ={0,1} // important as this is the language //which the computer understands.

  Σ={i,j,k}

- A certain version of language ALGOL has 113 letters.

Strings

- Definition:

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

- Example:

  If $\Sigma = \{a,b\}$ then a, abab, aaabb, abababababababab

Language

- Language is the set of all strings of terminal symbols derivable from alphabet.

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 will mostly be used to denote the empty string, in further discussion.

Words

- Definition:

  Words are strings belonging to some language.

- Example:

  If $\Sigma = \{a\}$ then a language L can be defined as $L=\{a^n : n=1,2,3,\ldots..\}$ or $L=\{a,aa,aaa,\ldots.\}$

  Here a,aa,… are the words of L

NOTE:

All words are strings, but not all strings are words

Valid/In-valid alphabets

- While defining an alphabet, an alphabet may contain letters consisting of group of symbols for example Σ_1= {B, aB, bab, d}.

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

Valid/In-valid alphabets

- This BababB (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

  $\Sigma_2 = \{B, Ba, bab, d\}$

Valid/In-valid alphabets

- As 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 <u>*not be the prefix*</u> of another. However, a letter may be ended in the letter of same alphabet *i.e.* one letter may be the <u>*suffix of another*</u>.

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:

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

  s=ababa

  |s|=5

- Example:

  Σ= {B, aB, bab, d}

  s=BaBbabBd

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

  |s|=5

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

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

  Rev(s)=dBbabaBB

Defining Languages

- The languages can be defined in different ways , such as
  1. Descriptive definition,
  2. Recursive definition,
  3. using Regular Expressions(RE) and
  4. using Finite Automaton(FA) etc.

- Descriptive definition of language:

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

- Example:

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

  L={a, aaa, aaaaa,…..}

- 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, …}

- 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}

- 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,…}

- 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,…}

- 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,…}

- 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,…}

- 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,…}

- Example:

  The language $\{a^n b^n\}$, of strings defined over $\Sigma=\{a,b\}$, as $\{a^n b^n : n=1,2,3,\ldots\}$, can be written as

  {ab, aabb, aaabbb,aaaabbbb,…}

- Example:

  The language $\{a^n b^n a^n\}$, of strings defined over $\Sigma=\{a,b\}$, as $\{a^n b^n a^n : n=1,2,3,\ldots\}$, can be written as

  {aba, aabbaa, aaabbbaaa,aaaabbbbaaaa,…

- Example:

  The language **factorial**, of strings defined over $\Sigma=\{1,2,3,4,5,6,7,8,9\}$ *i.e.* $\{1,2,6,24,120,\ldots\}$

- Example:

  The language **FACTORIAL**, of strings defined over $\Sigma=\{a\}$, as $\{a^{n!} : n=1,2,3,\ldots\}$, can be written as $\{a,aa,aaaaaa,\ldots\}$.

  It is to be noted that the language FACTORIAL can be defined over any single letter alphabet.

- 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,…}

- 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,…}

- 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,…}

- Example:

  The language **PRIME**, of strings defined over $\Sigma=\{a\}$, as

  $\{a^p : p \text{ is prime}\}$, can be written as {aa,aaa,aaaaa,aaaaaaa,aaaaaaaaaaa…}

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, ...}

Thank You…