**APPLICATIONS OF FINITE AUTOMATA(FA):**

The applications of Finite Automata are as follows-

* Design of the lexical analysis of a compiler.
* Recognize the pattern by using regular expressions.
* Use of the melay and moore machines for designing the combination and sequential circuits.
* Helpful in the text editors.
* Used for spell checkers.
* Used for protocol specification.
* Special purpose hardware design etc…

**APPLICATIONS OF REGULAR EXPRESSION:**

1.Regular expressions are useful in a wide variety of text processing tasks,and more generally string processing ,where the data need not be textual.

2.common applications include:

1.Data validation

2.Data Scraping

3.Data wrangling

4.Simple parsing

5.The production of syntax highlighting systems

**IDENTITY RULES FOR REGULAR EXPRESSION:**

Two Regular Expressions P&Q are equivalent(p=q) if P&Q represents the same set of strings

Identify rules are useful for simplifying Regular Expressions

* I1: Ø+R=R
* I2: ØR=RØ= Ø
* I3: ɅR=RɅ=R
* I4: n\*=ɅᶓØ\*=Ʌ
* I5: R+R=R
* I6: (R\*R\*)=R\*
* I7: RR\*=R\*R
* I8: (R\*)\*=R\*
* I9: Ʌ+RR\*=R\*=Ʌ+R\*R
* I10: (PQ)\*P=P(QP)\*
* I11: (P+Q)\*=(P\*Q\*)\*=(P\*+Q\*)\*
* I12: (P+Q)R=PR+QRᶓR(P+Q)=RP+RQ

**CLOSURE PROPERTIES OF REGULAR SET:**

We can generate new regular language (or)regular sets by applying closure properties on existing regular language(or)regular sets. The closure properties of regular set under

1.union

2.concatenation

3.closure(or)iteration

4.complementation

5.intersection

6.substitution

7.Homomorphism

8.inverse homomorphism

9.transpose

1. union:if L1&L2 are two regular sets then the union of L1&L2 i.e L1+L2 is also a regular set

2. concatenation: if L1&L2 are two regular sets then L1.L2 is also a regular set

3. if L is a regular set where L ≤ ∑\*,then the complementation of L i.e L' is alsoregular set L'=∑\*-L

4. if L is a a Regular Language then the L\* is also Regular Expression

5. if L1&L2 are regular then the intersection of L1&L2 is L1∩L2 is also regular

6.if L is a regular language such that L≤∑\* then its substitution f(L) is also regular

7.if L is a regular language then the homomorphic image of L is also regular H(L)

H(L)

H

L

domain co-domain

8. if L is a Regular Language then inverse homomorphic image of L i.e. H**-1**(L) is also regular

H

domain Co-domain

9. if L is a regular language then transpose of L i.e., LT is also regular

**DIFFERENCE BETWEEN DFA AND NFA:**

|  |  |
| --- | --- |
| **DFA** | NFA |
| 1.it is a deterministic | 1.it is not deterministic |
| 2.the mapping function (or) transition function is defined as  δ=QX∑→Q | 2. the transition function is defined as  δ=∑XQ→2Q |
| 3.every DFA is NFA | 3.every NFA need not to be DFA |
| 4.on one input symbol there will be only one transiyion to the next state | 4.there may be more than one transition over single input to multiple no of next states |
| 5.ε-transition are not allowed | 5.ε-transitions are allowed |
| 6.it is similar to serial(or)sequential processing | 6.it is similar to parallel(or)concurrent processing |
| 7.in DFA over an input symbol we know that what will be the next state | 7.here we don’t know the correct state on input symbols |
| 8. number of final states can be more than one | 8.in NFA the number of final states can be more than one |

**DIFFERENCE BETWEEN MOORE AND MELAY MACHINE:**

|  |  |
| --- | --- |
| **MOORE** | **MELAY** |
| 1.the output of moore depends on present state only | 1.the output of mealy machine is depends on present state and present input |
| 2 .the length of input string is n,output string is n+1 | 2.the length of input string is n,output string is n |
| 3.the output string is always associated with states | 3.the output string is always associated with transition |
| 4.the output function is defined as λ=Q→∆ | 4.the output function in defined as λ=QX∑→∆ |
| 5.the input string is ε,then the output string is λ(q0) | 5.the input string is ε,then the output string is ε |