**Chapter 2: REGULAR EXPRESSIONS**

**Topic – 1: Basic Information**

**Introduction**

* Algebraic representation of **languages**.

**Applications**

* Text editors to **search** for a word.
* Compiler designing.

**Topic – 2: Intuitive Examples**

**Examples**

|  |  |
| --- | --- |
| **Regular Expression** | **Language** |
| **0** | **{0}** |
| **Є** | **{Є}** |
| **0 U 01** | **{0, 01}** |
| **1\*** | **{Є, 1, 11, 111, …}** |
| **(0 U 01).1\*** | **{0, 01, 011, 0111, …}** |

**Explanation**

**(0 U 01).1\***

**= {0.Є, 01.Є, 0.1, 01.1, 0.11, 01.11, …}**

**= {0, 01, 01, 011, 011, 0111, …}**

**= {0, 01, 011, 0111, …}**

**Topic – 3: Formal Definition**

**Definition**

* **R** is said to be a ***regular expression*** **(RE)** if R has one of the following forms.

|  |  |  |
| --- | --- | --- |
| **Regular Expression** | **Language L(R)** | **Comment** |
| **Φ** | **{ }** |  |
| **Є** | **{Є}** |  |
| **a** | **{a}** | **a Є Z** |
| **R1 U R2** | **L(R1) U L(R2)** |  |
| **R1.R2** | **L(R1).L(R2)** |  |
| **R1\*** | **(L(R1))\*** |  |
| **(R1)** | **L(R1)** |  |

**Note!**

**🡪 For every R, there is a unique language L(R) corresponding to it.**

**🡪 Converse is not true.**

**🡪 We replace (U) with the (+) symbol & drop the (.) symbol.**

**Symbol Precedence**

* **()** [Highest], **\***, **(.)**, **U** [Lowest]
* **R\*** means set of strings made from taking **0** **or more** strings from **R**.

**Φ\* = {Є}**

**Topic – 4: More Examples**

**Regular Expressions To Language**

|  |  |
| --- | --- |
| **Regular Expression** | **Language** |
| **01** | **{01}** |
| **01 + 1** | **{01, 1}** |
| **(01 + Є) 1** | **{011, 1}** |
| **(0 + 10)\*(Є + 1)** | **{Є, 1, 0, 01, 10, 101, 00, 001, 010, 0101, 100, 1001, 1010, 10101, …}** |

**Language To Regular Expressions**

|  |  |
| --- | --- |
| **Language** | **Regular Expression** |
| **{w | w has a single 1}** | **0\*10\*** |
| **{w | w has at most a single 1}** | **0\* + 0\*10\*** |
| **{w | |w| is divisible by 3}** | **((0+1)(0+1)(0+1))\*** |
| **{w | w has a 1 at every odd position & |w| is odd}** | **1((0+1)1)\*** |

**Topic – 5: Properties**

**Associative Properties**

**R1 + (R2 + R3) = (R1 + R2) + R3**

**R1(R2R3) = (R1R2)R3**

**Commutative Properties**

**(R1 + R2)R3 = R1R3 + R2R3**

**R1(R2 + R3) = R1R2 + R1R3**

**Commutative Property**

**R1 + R2 = R2 + R1**

**Identity Properties**

**R + φ = R**

**RЄ = R**

**(R\*)\* = R\***

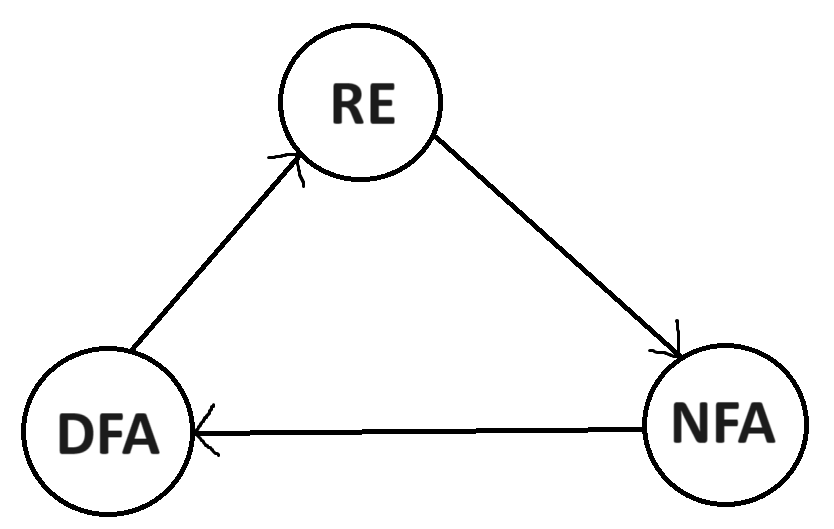
**Theorem**

A language **L** is **regular** if and only if there is a **regular expression** **R** such that **L = L(R)**.

**Topic – 6: Conversions**

**Possible Conversions**

* Possible conversions are as below:



**Proof**

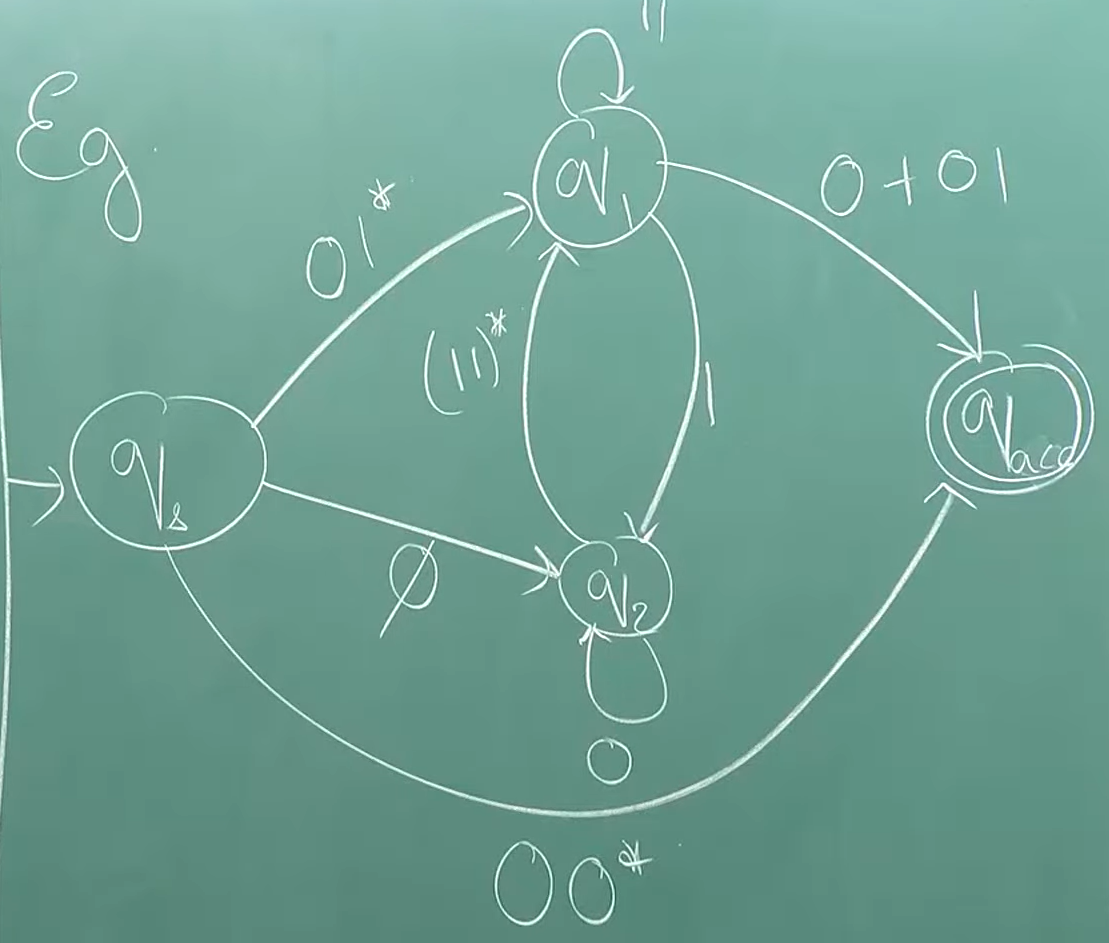
* Based on **case analysis**.

|  |  |
| --- | --- |
| **RE** | **NFA** |
| **Φ** | **Single state.** |
| **Є** | **Single state, initial state is the accept state.** |
| **a** | **Second state transitioning with symbol 'a' is the accept state.** |
| **R1 + R2** | **Both RE have same initial & accept state, with concurrent Є transition to their individual system.** |
| **R1.R2** | **Both RE have same initial & accept state, with sequential Є transition to their individual system.** |
| **R\*** | **Self-recursion using Є.** |

**Generalized Nondeterministic Automata (GNFA)**

* **NFA** that has **regular expressions** labelling its transition.

**GNFA Example**



* For an example, **01101** when cut into some **pieces** of **irregular length** will satisfy the diagram above.