L(A) = {w | w is in {0,1}* and the length of w is divisible by 3}

Important Observations:

1. States possible is {0, 1, 2}

L(A) = {w | w is in {0,1}* and w is divisible by 3}

Important Observations:

1. States possible is {0, 1, 2}
2. If w represents an integer i then δ(0, w) = i % 3
3. If w0 represents an integer 2i then δ(i % 3, 0) = (2i) % 3
4. If w1 represents an integer (2i + 1) then δ(i % 3, 1) = (2i + 1) % 3

w	a	input	i	Decimal (i)	Decimal (2i)	Transition Function	State
0	-	0	0	0	0	δ(0, 0)	0
1	-	1	1	1	1	δ(0, 1)	1
1	0	10	1	1	(2i) = 2	δ(0, 10) δ(δ(0, 1), 0) δ(1, 0)	(2i) % 3 = 2
1	1	11	1	1	(2i + 1) = 3	δ(0, 11) δ(δ(0, 1), 1) δ(1, 1)	(2i + 1) % 3 = 0
10	0	100	10	2	(2i) = 4	δ(0, 100) δ(δ(δ(0, 1), 0), 0) δ(δ(1, 0), 0) δ(2, 0)	(2i) % 3 = 1
⋮							

Regular Expression (RE):

- Representing the Finite Automaton by expression
- Widely used in Computing Science
  - If a is any symbol then a is a RE, and L(a) = {a}
  - ϵ is a RE, and L(ϵ) = {ϵ}
  - $\varnothing$ is RE, and L($\varnothing$) = $\varnothing$

Regular Expression Operator:

- ( ) parenthesis, used to describe the grouping of operators
- Union ($\cup$) also called 'OR' (+) : Binary Opeartor

  If E_1 and E_2 are RE then $E_1 + E_2$ is a RE and $L(E_1 + E_2) = L(E_1) \cup L(E_2)$
  - $L(0) \Rightarrow (0) = \{0\}$
  - $L(1) \Rightarrow (1) = \{1\}$
  - $L(0 + 1) \Rightarrow (0 + 1) = L(0) \cup L(1) = \{0, 1\}$
  - $L(01) \Rightarrow (01) = \{01\}$
  - $L(01 + 1) \Rightarrow (01 + 1) = L(01) \cup L(1) = \{01, 1\}$
  - $L(11) \Rightarrow (11) = \{11\}$
  - $L(01 + 11) \Rightarrow (01 + 11) = L(01) \cup L(11) = \{01, 11\}$
  - L = {01, 111, 10} and M = {00, 01}

    L + M = {01, 111, 10, 00}

- Concatenation also called 'AND' (.) : Binary Opeartor

  If E_1 and E_2 are RE then E_1E_2 is a RE and $L(E_1E_2) = L(E_1)L(E_2)$
  - L(0) = {0}
  - L(1) = {1}
  - $L(0.1) \Rightarrow L(0) \,.\, L(1) \Rightarrow L(01) \Rightarrow (01) = \{01\}$
  - $L(01 \,.\, 11) \Rightarrow L(01) \,.\, L(11) \Rightarrow L(0111) = (0111) = \{0111\}$
  - L = {01, 111, 10} and M = {00, 01}

    LM = {0100, 0101, 11100, 11101, 1000, 1001}

- Kleene Star also called ‘Star’/ ‘Closure’ (*) : Unary Operator

  If E RE then E* is a RE and L(E*) = (L(E))*

  $L^* = \{\epsilon\} \cup L \cup LL \cup LLL \cup \ldots$

  $L(0)^* \Rightarrow (0)^* \Rightarrow 0^* = \{\epsilon, 0, 00, 000, 0000, \ldots\}$

  $L(ab)^* \Rightarrow (ab)^* = \{\epsilon, ab, abab, ababab, \ldots\}$

  $L(01)^* \Rightarrow (01)^* = \{\epsilon, 01, 0101, 010101, \ldots\}$

  $L = \{0, 10\}$ then $\{0, 10\}^* = \{\epsilon, 0, 10, 00, 010, 100, 1010, \ldots\}$

  $\{a, b, c\}^* = ??$

- Kleene Plus also called ‘Plus Closure’ ($^+$)

  $L^+ = L \cup LL \cup LLL \cup \ldots$

  $L(0)^+ \Rightarrow (0)^+ \Rightarrow 0^+ = \{0, 00, 000, 0000, \ldots\}$

  $\{a, b, c\}^+ = ??$

Precedence of Operator:

- ( ) Highest
- *
- .
- + Lowest

Algebraic Laws for RE’s:

- (+) is commutative and associative
- (.) distributed over (+)
- (.) is not commutative
- ($\emptyset$) is identity for (+) $\Rightarrow R + \emptyset = \emptyset + R = R$
- (ϵ) is the identity for (.) $\Rightarrow \epsilon R = R\epsilon = R$
- ($\emptyset$) is annihilator for (.) $\Rightarrow \emptyset R = R\emptyset = \emptyset$

Example:

1. L((a + b) a) $\Rightarrow$ (a + b) a $\Rightarrow$ {aa, ba}
2. L(0 (1 + 0)) $\Rightarrow$ 0 (1 + 0) $\Rightarrow$ {01, 00}
3. a*. b* = ??
4. (a + b)* = ??
5. (0 + 1)* = ??
6. L = {w| w is in {0, 1}* and w is odd}
7. L = {w| w is in {0, 1}* and w is even}
8. L = {w| w is in {0, 1}* and all strings starts with 0}
9. L = {w| w is in {0, 1}* and all strings starts with 1}
10. L = {w| w is in {0, 1}* and all strings ends with 0}
11. L = {w| w is in {0, 1}* and all strings ends with 1}
12. L = {w| w is in {0, 1}* and all strings starts with 01}
13. L = {w| w is in {0, 1}* and all strings ends with 01}
14. L = {w| w is in {0, 1}* and all strings contains 0}
15. L = {w| w is in {0, 1}* and all strings holds substring 01}
16. L = {w| w is in {0, 1}* and all strings starts and ends with same symbol}
17. L = {w| w is in {0, 1}* and all strings starts and ends with different symbol}
18. L = {w| w is in {0, 1}* and all strings contains exactly 2 0's}
19. L = {w| w is in {0, 1}* and all strings contains at least 2 0's}
20. L = {w| w is in {0, 1}* and all strings contains at most 2 0's}
21. L = {w| w is in {0, 1}* and all strings contains even number of 0's}
22. L = {w| w is in {0, 1}* and all strings with even length}
23. L = {w| w is in {0, 1}* and all strings with odd length}
24. L = {w| w is in {0, 1}* and all strings with length divisible by 2}
25. L = {w| w is in {0, 1}* and all strings with length divisible by 3}
26. L = {w| w is in {0, 1}* and all strings with no occurrence of 00}
27. L = {w| w is in {0, 1}* and all strings with exact occurrence of 00 once}
28. L = {w| w is in {0, 1}* and all strings has no 00}