6/1/2021 Yash Verma 19075083 CSE

Notations:

- If Σ is an alphabet, $a \in \Sigma$, $x \in \Sigma^*$ and $L \subseteq \Sigma^*$

$a^k = \underbrace{a\,a\,a \ldots a}_{k\text{ times}}$

$x^k = x\,x\,x \ldots x$.

$\Sigma^k = \Sigma\,\Sigma\,\Sigma \ldots \Sigma = \{x \in \Sigma^* : |x| = k\}$.

$L^k = L.L.L \ldots L$.

- When $k = 0$, $a^0 = \epsilon$, $x^0 = \epsilon$, $\Sigma^0 = \epsilon$, $L^0 = \epsilon$

- Unit of concatenation is $\epsilon / \{\epsilon\}$.

- Set of all strings that can be obtained by concatenating any no. of elements of L:

$$L^* = \bigcup_{i=0}^{\infty} L^i$$

The operation $*$ is called Kleene star.

- $L^0 = \{\epsilon\}$ for any language L.

ϵ is always element of L^*

- Let $L^+ = \bigcup_{i=1}^{\infty} L^i$. $\boxed{L^+ = L.L^* = L^*.L}$

- String are finite, language may be infinite.

$L^+ = L^* \mid \{\epsilon\}$

Describing Languages:

Example:

$L_1 = \{ab, bab\}^* \cup \{b\}\{bb\}^*$

Language L_1 is constructed either by ~~con~~ (+) an arbitary no. of strings each of which is either ab or bab.
OR
by ~~cos~~ (+) string b with arbitary no. of string bb.

$L_2 = \{x \in \{a, b\}^* \mid n_a(x) \geq n_b(x)\}$

L_3 = set of strings.

Recursive definition of L^*

(1) $\epsilon \in L^*$

(2) For any $x \in L^*$, $y \in L$, $xy \in L^*$

(3) No string is in L^*, unless it cant be obtaind by (1) and (2)

Example:

$L = \{a, ab\}$

$\epsilon \in L^*$ — (1)

$L^* \Rightarrow \{\epsilon a, \epsilon ab\} = \{a, ab\}$ — (2)

Repeating step (2), we get ϵa, ϵab, aa, aab, aba...

For any $k \geq 0$, a string obtained by (+) k elements of L can be obtained by using k-times rule 2.

Proof will mainly use mathematical induction techniques:

1. Basis step
2. Induction hypothesis
3. Show induction step.
4. Proof of induction step.

Problem:

If Σ is an alphabet, L is a language over Σ, then the problem is:

"Given a string $w \in \Sigma^*$, decide whether or not $w \in L$".

Deterministic Finite Automata

DFA consists of:

1. finite set of states, (Q)
2. finite set of input symbols (Σ)
3. start state, one of states in Q (q_0)
4. Set of final states $F \subseteq Q$
5. Transition fn that takes an argument a state and a an input symbol and returns a state, denoted by δ, defined as

$$\delta : Q \times \Sigma \rightarrow Q$$

Representation of DFA:

$$A = (Q, \Sigma, \delta, q_0, F)$$

$\downarrow$

name

DFA processing a string:

Suppose $w = a_1 a_2 \ldots a_n$,

- DFA begins in its initial state q_0.
- Consult the transition fn, δ.

  $\delta(q_0, a_1) = q_1$ (say)

  to find the state that the DFA A enters after processing the first input symbol a_1.
- Process next input symbol a_2,

  $\delta(q_1, a_2) = q_2$ (say)
- Finding states $q_3, q_4 \ldots$

  $\delta(q_{i-1}, a_i) = q_i \quad \forall i$
- At the end, $\delta(q_{n-1}, a_n) = q_n$.

$q_n \in F$

$\Rightarrow w \in L(A)$ [input string w is accepted, otherwise the input is rejected]

Transition table:

→ Conventional tabular representation of fn δ, takes two arguments and returns a value.

$Q \backslash \Sigma$	a_1	a_2	a_3	$\ldots$
q_0	$\delta(q_0, a_1)$			
q_1		$\delta(q_1, a_2)$		
q_2				
$\vdots$				

Example:

δ	0	1
q_0	q_2	q_0
$*q_1$	q_1	q_1
q_2	q_2	q_1

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{0, 1\}$

$01 \in L(A)$

$000 \notin L(A)$

Extended transition function:

δ^*.

It is defined using induction on the length of input string.

- (Basic) $\delta^*(q, \epsilon) = q$
- (Induction) w is a string of form $w = xa$.

$a \in \Sigma$, is last symbol of w and x is the string consisting of last symbol

$$\delta^*(q, w) = \delta(\delta^*(q, x), a)$$

To compute $\delta^*(q, w)$, first $\delta^*(q, x)$.

If $p = \delta^*(q, x)$

$\delta^*(q, w) = \delta(p, a)$

$A = (Q, \Sigma, \delta, q_0, F)$ is defined by

$L(A) = \{ w : \delta^*(q_0, w) \in F \}$

If L is a language for DFA, then we say L is a regular language.

Example:

$L = w \in \{0,1\}^* \mid w$ is of even length and starts with $0,1 \}$

$M = \{ \{q_0, q_1, q_2, q_3, q_4\}, \{0,1\}, \delta, q_0, \{q_2\} \}$

δ is given by