

Lecture Mar 18

Monday, March 18, 2019 11:11 AM

Language Theory

Definition Σ finite set of letters

Σ^* set of all finite strings with letters from Σ

$\lambda \in \Sigma$ empty string

A language over Σ is a subset of Σ^* .

- If $L_1, L_2 \subseteq \Sigma^*$, then $L_1 L_2 = \{xy \mid x \in L_1 \text{ and } y \in L_2\}$

If $L_1 = \{a, bb\}$, and $L_2 = \{x, c\}$. then $L_1 L_2 = \{a, bb, ac, bba\}$

- $L_i^k = \underbrace{L_1 \dots L_1}_k$ $L_i^1 = L_i$, $L_i^0 = \{\lambda\}$.

L_i^* all strings that are the concatenation of 0 or more strings in L_i

$$L_i^* = \bigcup_{k \geq 0} L_i^k$$

$$L_i^+ = \bigcup_{k \geq 1} L_i^k \quad L_i^* = L_i^+ \text{ if and only if } \lambda \in L_i$$

Union, intersection, complementation

$$\overline{L_i} = \Sigma^* - L_i$$

Regular Expressions

Let Σ be a finite alphabet $\emptyset, \lambda \in \Sigma$, $\Sigma \subseteq R$

If $r, r' \in R$ then:

$$(r+r'), (r \cdot r') \text{ and } r^* = R$$

R: set of regular expressions over Σ

$$R \subseteq (\Sigma \cup \{\lambda, \emptyset, \cdot, +, *\}, (.,)^*)^*$$

The language $L(r)$ denoted by a regular expression $r \in R$.

.. concatenation

$L: R \rightarrow P(\Sigma^*)$ is defined inductively as:

$$L(\emptyset) = \emptyset \leftarrow \text{empty set}$$

$$L(a) = \{a\} \text{ for } a \in \Sigma$$

$$L(r \cdot r') = L(r) \cdot L(r') \quad (\text{first dot is just dot, second is concatenation})$$

$$L(r+r') = L(r) \cup L(r')$$

$$L(r^*) = (L(r))^*$$

$L \subseteq \Sigma^*$ is regular if and only if $L = f(r)$ for some r over Σ .

Example: - Language of all strings in $\{a,b,c\}$ that start with ab

$$L(a \cdot b \cdot (a+b+c)^*)$$

- Regular expression for strings over $\{0,1\}$ with even parity.

$$\{x \in \{0,1\}^* \mid \#\text{1s in } x \text{ is congruent to } \#0 \pmod{2}\}$$

$$(0^* 1 0^* 1 0^*)^* 0^*$$

- First letter different from last letter $\{0,1\}$

$$L((0 \cdot (0+1)^* \cdot 1) + (1 \cdot (0+1)^* \cdot 0))$$

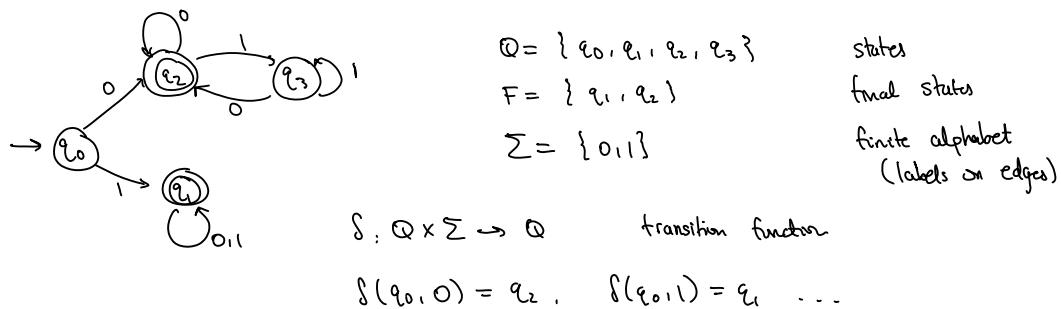
if set is instead $\{0,1,2\}$

$$0 \cdot (0+1+2)^* \cdot (1+2) + 1 \cdot (0+1+2)^* \cdot (0+2) + 2 \cdot (0+1+2)^* \cdot (0+1)$$

- x contains at most 2 a's

$$b^* + b^* \cdot a \cdot b^* + b^* \cdot a \cdot b^* \cdot a \cdot b^*$$

Deterministic Finite Automaton (DFA)



Formally: $M = (Q, \Sigma, \delta, q_0, F)$

Given a string $a_1, \dots, a_n \in \Sigma^*$.

M starts in q_0 . Reads 1 letter at a time left to right.

Changes state according to δ . When all letters are read, it accepts if it is in final state. Reject if its in $Q-F$.

$$L(M) = \{x \mid M \text{ accepts } x\} = \{x \in \Sigma^* \mid \delta^*(q_0 x) \in F\}$$

$$\delta^*: Q \times \Sigma^* \rightarrow Q : \quad \delta^*(q, \lambda) = q \quad \forall q \in Q$$

$$\delta^* : Q \times \Sigma^* \rightarrow Q$$
$$\delta^*(q, \lambda) = q \quad \forall q \in Q$$
$$\delta^*(q \alpha) = \delta(\delta^*(q, \alpha)) \quad \forall q \in Q, \alpha \in \Sigma, \alpha \in \Sigma^*$$
$$\stackrel{\alpha}{=} \delta^*(q, \alpha x) = \delta^*(\delta(q, \alpha), x)$$