MATH 2301

25/09/2023

* Last time : A word w matches a regex r if $w \in L(r)$.

Example

Given $L = \{ w \in \Sigma^* \mid w \text{ contains an even number of 1s} \}$

find a regex r such that $L(r) = L$.

Attempts from Friday :

$r = (11)^*$ ← not good ; misses e.g. 101

$r = (1 0^* 1)^*$ ← not good ; misses 0101

$r = (\underbrace{0^* 1 0^* 1 0^*}_{})^*$ ← not good ; misses 000

↳ matches anything with exactly 2 1's

$r = (0^* 1 0^* 1 0^*)^* \mid 0^*$ [convince yourself that this one works!]

---

* Notation : If $a \in \Sigma$, and $k \in \mathbb{N}$, then a^k denotes a k-fold concatenation of a.

$\underbrace{a\, a \cdots a}_{k \text{ times}}$

Technically not a regex, but often used as a shorthand.

! ** Try not to use this if asked to produce a regex!!

** Questions :

From Friday, we have a procedure that takes in regexes and tells you the language of the regex.

E.g. $L(0^*) = (L(0))^* = (\{0\})^*$

(*) Can you go backwards? Given a description of a language L, can we always produce a regex r such that $L = L(r)$?

We'll investigate ...

(*) If $L = L(r)$ for some r, then is r the only regex with this property?

Previous example : $r = (0^* 1 0^* 1 0^*)^* \mid 0^*$

Other options: $0^* \mid (0^* 1 0^* 1 0^*)^*$,

$0^* \mid (0^* 1 0^* 1 0^*)^* \mid 00$

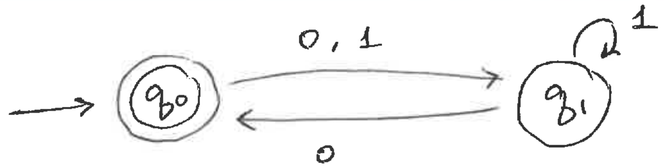
$\boxed{\text{NO}}$. There can be many regexes that produce the same language.

---

* Automata : If a human/machine tries to check whether some string satisfies some pattern, it is more natural to read the string letter by letter, to check the pattern.

* Deterministic Finite Automata (DFAs)

Example (state diagram)



E.g. $w = 1011$

$$q_0 \xrightarrow{1} q_1$$
$$q_1 \xrightarrow{0} q_0$$
$$q_0 \xrightarrow{1} q_1$$
$$q_1 \xrightarrow{1} q_1$$

** Def: A DFA consists of the following:

(1) An alphabet Σ

(2) A finite set of "states" or vertices Q.

(3) A start state $q_0 \in Q$ (denoted by dangling incoming arrow $\rightarrow q_0$)

(4) A set $A \subseteq Q$ of "accepting/accept" states (denoted by a double circle q)

(5) A transition function

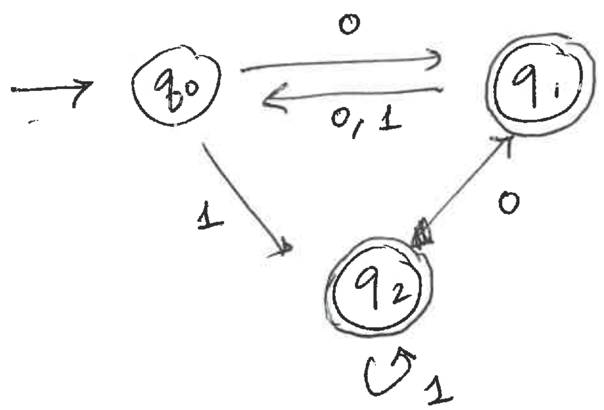
$$\delta : Q \times \Sigma \longrightarrow Q.$$

$$(q, a) \longmapsto q'$$

(Move from q to q' if you see the letter a)

* <u>Note</u>: Given a DFA, you can draw its state diagram as in the first example.

<u>E.g.</u>



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

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

$A = \{q_1, q_2\}$

start state is q_0

δ ?

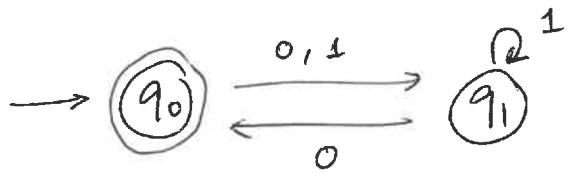
Q	Σ	output of δ
q_0	0	q_1
q_0	1	q_2
q_1	0	q_0
q_1	1	q_0
q_2	0	q_1
q_2	1	q_2

[Follow the arrows in the state diagram]

<u>Note</u>: A label of 0, 1 is equivalent to writing Σ as a shorthand, if $\Sigma = \{0, 1\}$

More generally, if $\Sigma = \{a_1, a_2 \ldots, a_n\}$ then you may use Σ to denote the label $a_1, a_2, \ldots, a_n$.

E.g.



An example calculation

$w = \underline{1}011$

Read w from left to right, letter by letter, starting at q_0, and applying δ as we go.

$q_0 \xrightarrow{1} q_1$ is equivalent to saying $\delta(q_0, 1) = q_1$

$q_1 \xrightarrow{0} q_0$; i.e $\delta(q_1, 0) = q_0$

$q_0 \xrightarrow{1} q_1$

$q_1 \xrightarrow{1} \boxed{q_1}$ ← last state we reach at the end of the string

If this state is in A, return $\boxed{\text{ACCEPT}}$

If not, return. $\boxed{\text{REJECT}}$

Note: If M is a DFA, we say that L(M) is the set of all strings accepted by M.