Regular expressions

Tuesday, January 19, 2021 11:09 AM

A language for describing patterns in strings.

Fix an alphabet $\Sigma = \{a, b\}$

(1) $\emptyset$ is a regular expression

(2) ε is a regular expression

(3) $a \in \Sigma$ is a regular expression

(4) If R, S are regular expressions
so is $R \cdot S$

(5) If R, S are regular expressions
so is $R + S$

(6) If R is a regular expression
R^* is also a regular expression.

EXAMPLES $\Sigma = \{a, b\}$

(i) $ab + \varepsilon$ (ii) $(a^* b)^*$ (iii) $a^* + b^*$ (iv) $aa^* b$ (v) $\emptyset$

SEMANTICS of regular expressions:

Each regular expression defines a subset of Σ^* i.e. a language.

(1) ϕ stands for the empty set

(2) ε defines $\{\varepsilon\} \neq \phi$

(3) a defines $\{a\}$

Suppose R defines the set $\hat{R}$, S defines $\hat{S}$

(4) $\widehat{R \cdot S} = \{\omega_1 \omega_2 \mid \omega_1 \in \hat{R}, \omega_2 \in \hat{S}\}$

(5) $\widehat{R + S} = \hat{R} \cup \hat{S}$

(6) $\widehat{R^*} = \{\omega_1 \omega_2 \cdots \omega_n \mid \text{each } \omega_i \in \hat{R}\} \cup \{\varepsilon\}$

$\widehat{ab + \varepsilon} = \{\varepsilon, ab\}$

$\widehat{(a^* b)^*} = \{\varepsilon, b, bb, abab, aabab, \cdots\}$

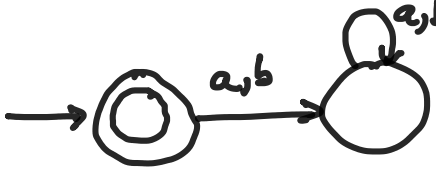
Thm (KLEENE) The language defined by any regular expression is a regular language i.e. can be recognized by an NFA+ε (NFA, DFA).

Furthermore every regular language can be described by a regular exp.

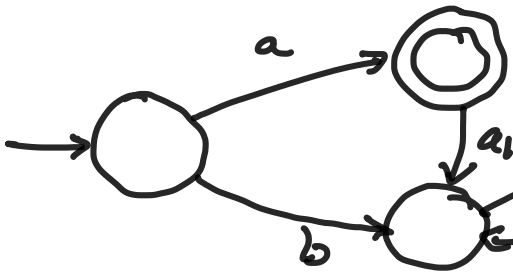
can be written ... if:

Proof (Part I) From regexp → NFA

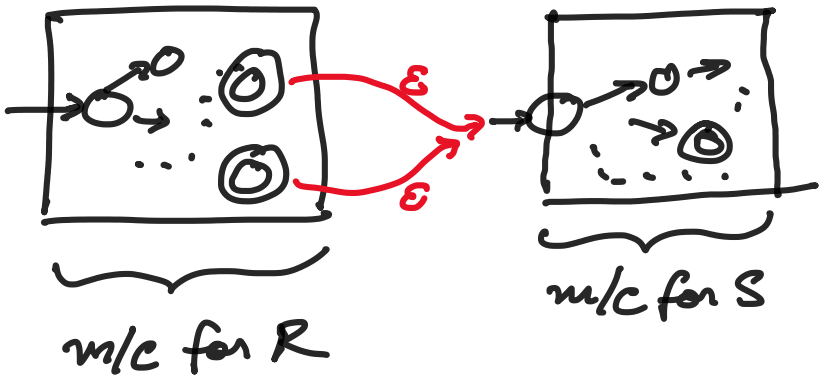
recognizes ϕ

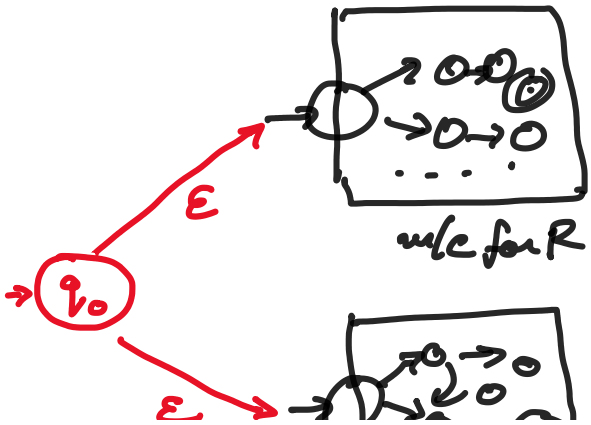


recognizes ε



recognizes a





$M_1 = (S_1, s_1, \delta_1, F_1)$

$M_2 = (S_2, s_2, \delta_2, F_2)$

New m/c NFA+ε

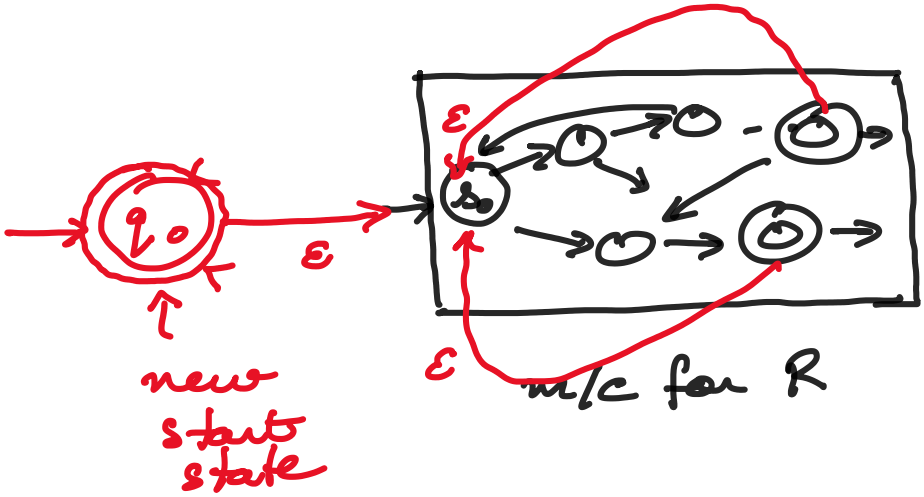
States = $S_1 \cup S_2 \cup \{q_0\}$

Start states : $\{I_0\}$

$$\Delta(q,a) = \begin{cases} \{\delta_1(q,a)\} \text{ if } q \in S_1 \\ \{\delta_2(q,a)\} \text{ if } q \in S_2 \\ \{s_1, s_2\} \text{ if } q = q_0,\ a = \varepsilon \end{cases}$$

Final states $F_1 \cup F_2$

Given a DFA to recognize $\hat{R}$ we construct a new machine to recognize $\hat{R}^*$.



Why did