# Regular expressions

Tuesday, January 19, 2021 11:09 AM

A *language* for describing patterns in strings.

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

(1) $\phi$ is a regular expression
(2) $\varepsilon$ is a regular expression
(3) $a \in \Sigma$ is a regular expression
(4) If $R, S$ are regular expressions so is $R \bullet S$
(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) $\phi$

SEMANTICS of regular expressions:

Each regular expression defines a subset of $\Sigma^*$ i.e. a language.

(1) $\phi$ stands for the empty set
(2) $\varepsilon$ defines $\{\varepsilon\} \neq \phi$
(3) $a$ defines $\{a\}$

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

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

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

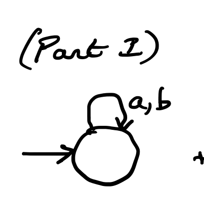
$\widehat{(a^* b)^*} = \{\varepsilon, b, bb, abab, aabab, \dots \}$

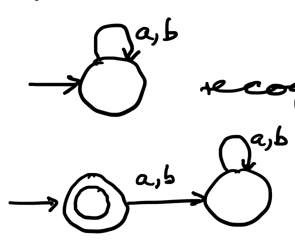
---

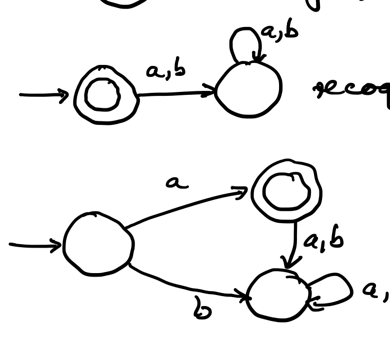
Thm (KLEENE) The language defined by any regular expression is a regular language i.e. can be recognized by an NFA+ε (NFA, DFA).

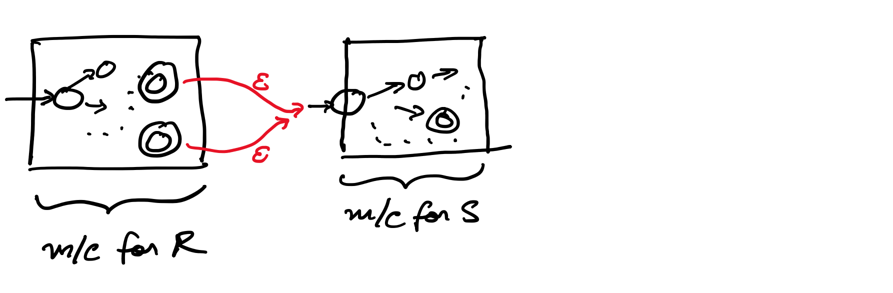
Furthermore every regular language can be described by a regular exp.

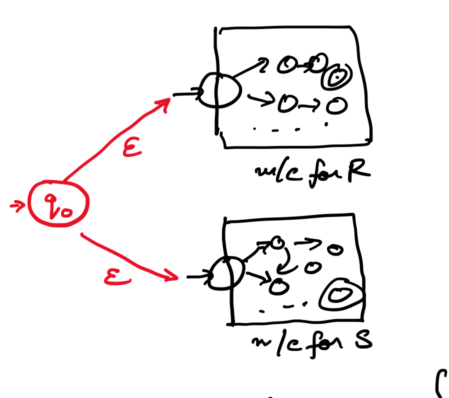
Proof (Part I) From regexp → NFA


recognizes $\phi$


recognizes $\varepsilon$


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

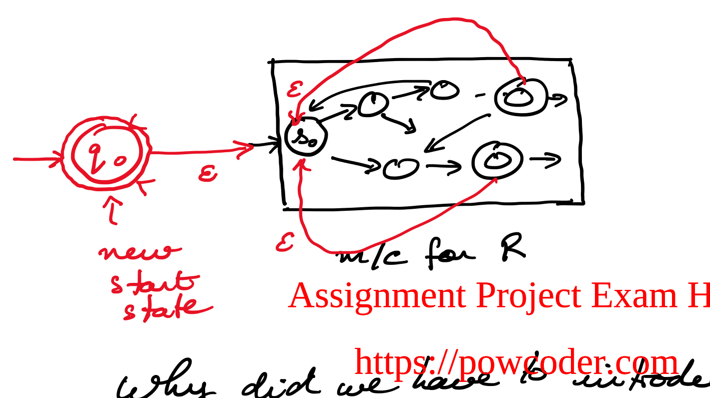
states $= S_1 \cup S_2 \cup \{q_0\}$

start states $: \{q_0\}$

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

Final states $F_1 \cup F_2$

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



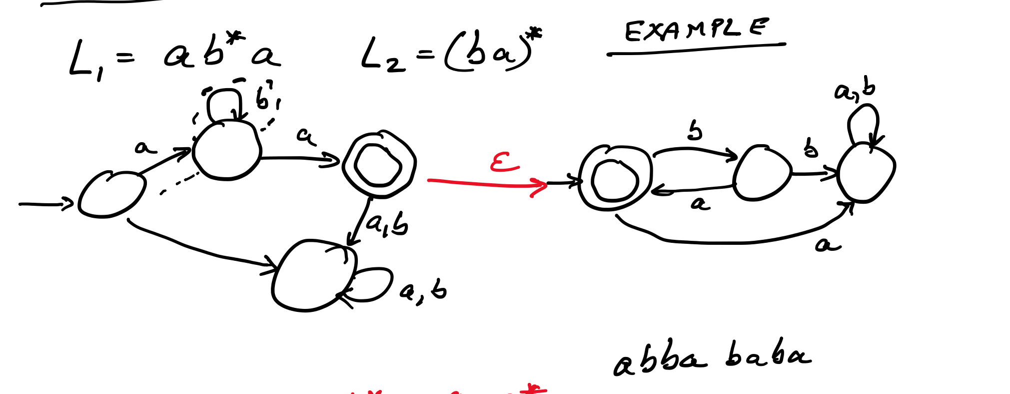



Why did we have to introduce a new start state?

EXERCISE: Explain why we have to do this.

---

$L_1 = ab^*a$ $L_2 = (ba)^*$ EXAMPLE




abba baba

$ab^*a(ba)^*$

---

PART II Given a DFA $M = (S, s_0, \delta, F)$

We are going to enumerate the states of $M$ : $1, \dots n$

We are going to define a family of regular expressions $R_{ij}^{(k)}$ $i,j,k \in \{1, \dots n\}$

$R_{ij}^{(k)}$ : this is a regular expression describing all words that take the m/c from state $i$ to state $j$ and along the way they only stop at states numbered $k$ or less.

We will construct this family for each $k$ from $0$ to $n$

$k = 0$ : there are no states numbered 0.

$R_{ij}^{(0)}$ direct jumps from $i$ to $j$.

If $\exists!\, a \in \Sigma$ s.t. $\delta(s_i, a) = s_j$

$R_{ij}^{(0)} = a$

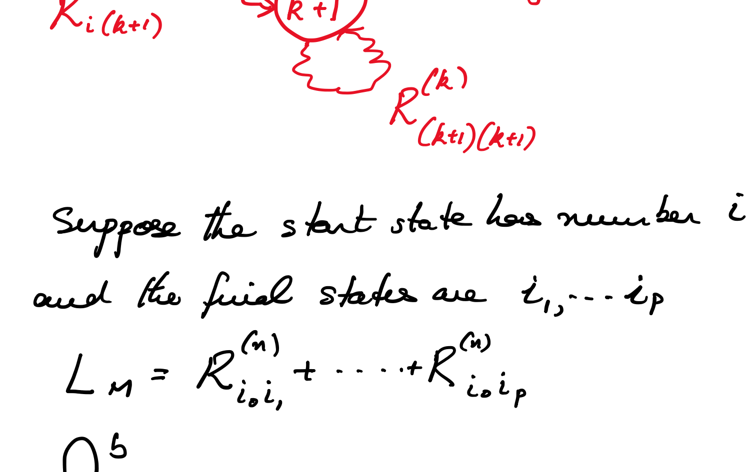
If there are several such letters in $\Sigma$ say $a_1, \dots a_\ell$ then

$R_{ij}^{(0)} = a_1 + a_2 + \dots + a_\ell$

If $i = j$ we do the same except we add $\dots\dots + \varepsilon$

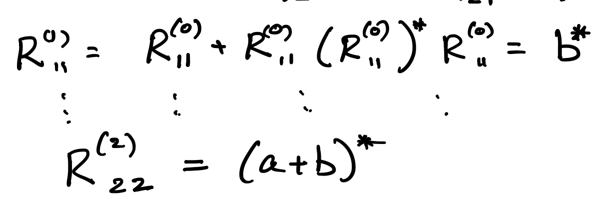
Suppose we have constructed the regexp for every $i$ and $j$ up to $k$ for some value of $k$. Now consider $R_{ij}^{(k+1)}$

$$R_{ij}^{(k+1)} = R_{ij}^{(k)} + R_{i(k+1)}^{(k)} \left( R_{(k+1)(k+1)}^{(k)} \right)^* R_{(k+1)j}^{(k)}$$




Suppose the start state has number $i_0$ and the final states are $i_1, \dots i_p$

$L_M = R_{i_0 i_1}^{(n)} + \dots + R_{i_0 i_p}^{(n)}$




$R_{11}^{(0)} = \varepsilon + b$ $R_{12}^{(0)} = a$ $R_{21}^{(0)} = \phi$ $R_{22}^{(0)} = \varepsilon + a + b$

$R_{11}^{(1)} = R_{11}^{(0)} + R_{11}^{(0)} (R_{11}^{(0)})^* R_{11}^{(0)} = b^*$

$\vdots \quad \vdots \quad \vdots \quad \vdots$

$R_{22}^{(2)} = (a+b)^*$

$R_{12}^{(2)}$ is the language of this m/c

$b^*a + b^*a(\varepsilon + a + b)^*(\varepsilon + a + b)$

$= b^*a(a+b)^*$

---

There is an *algebra* of regular expressions

$R_{12}^{(1)} = b^*a$