Teorie jazyků a automatů— 5. týden

Marie Demlová
http://math.feld.cvut.cz/en/people/demlova

21. 10. 2024

Operace s jazyky

$L_1 \cup L_2$, $\overline{L_1}$, $L_1 \cap L_2$, $L_1 L_2$, L_1^*, $L_1^R = \{w^R \mid w \in L_1\}$

↓

Věta.

Třída regulárních jazyků je uzavřena na následující operace: sjednocení, doplněk, průnik, zřetězení, Kleeneho operaci $\star$ a reverzi.

Přesněji, jestliže L, L_1 a L_2 jsou regulární jazyky, pak také $L_1 \cup L_2$, $\overline{L}$, $L_1 \cap L_2$, $L_1 L_2$, $L^\star$ a L^R jsou regulární jazyky.

$L_1 \in Reg$ ⇒ L_1 existuje $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ DFA

$\rightsquigarrow$ $\overline{M_1} = (Q_1, \Sigma, \delta_1, q_1, Q_1 - F_1)$ $L(\overline{M_1}) = \overline{L_1}$

$L(M_1) = L_1$ $w \in L_1$ iff $\delta_1^*(q_0, w) \in F_1$

Operace s jazyky

Součinová konstrukce.

Dány DFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ a $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ takové, že $L_1 = L(M_1)$ a $L_2 = L(M_2)$. Definujeme DFA $M = (Q, \Sigma, \delta, q_0, F)$ takto:

- $Q = Q_1 \times Q_2$,
- $q_0 = (q_1, q_2)$,
- $\delta((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$,
- $F = F_1 \times F_2$.

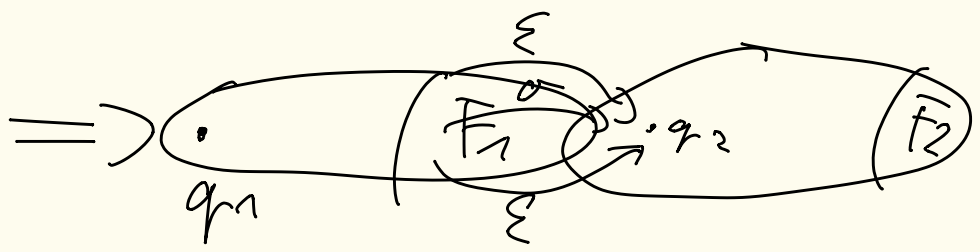
Pak $L(M) = L_1 \cap L_2$.

Pz. $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$

$L_1 L_2$ $M_i = (Q_i, \Sigma, \delta_i, q_{i}, F_i)$ $i = 1,2$ $L(M_i) = L_i$

$Q_1 \cap Q_2 = \emptyset$

$M \subseteq L(M) = L_1 L_2$



$w \in L_1 L_2$ iff $w = xy$

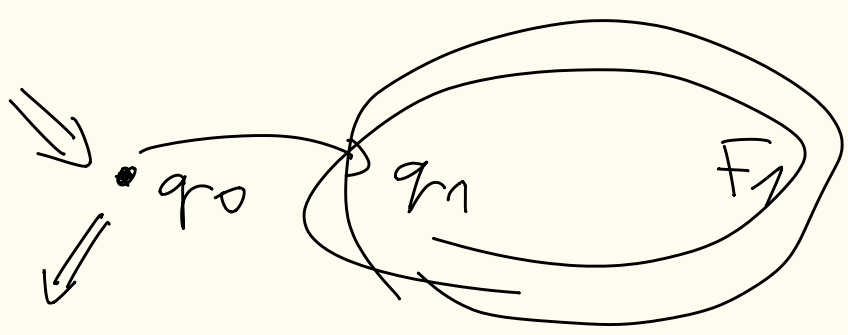
$x \in L_1$ $y \in L_2$

$\delta(q,a) = \delta_1(q,a)$ per $q \in Q_1$

$\delta(p,a) = \delta_2(p,a)$ per $p \in Q_2$

L_1^* $\overline{L_1^*}$

$L_1^* = \{\varepsilon\} \cup L_1 \cup L_1^2 \cup$



$Q = Q_1 \cup \{q_0\}$

$\delta(q_0, \varepsilon) = q_1$

$\delta(q,a) = \delta_1(q,a)$ $a \in Q_1$

$\delta(p, \varepsilon) = q_1$ $\forall p \in F_1$

$F = F_1 \cup \{q_0\}$

L_1^R M_1 per L_1 $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1) \rightsquigarrow$

$M_1^R = (Q_1, \Sigma, \delta_1^R, F_1, \{q_1\})$ $\delta_1^R(q,a) = p$ iff $\delta_1(p,a) = q$

NFA

$M_i = (Q_i, \Sigma, \delta_i, q_i, F_i)$ $i = 1,2$ $L(M_i) = L_i$

$\downarrow$

$M = (Q, \Sigma, \delta, q_0, F)$ $Q = Q_1 \times Q_2$ $\delta((p,q), a) = (\delta_1(p,a), \delta_2(q,a))$

$\delta^*((p,q), w) = (\delta_1(p,w), \delta_2^*(q,w))$ $q_0 = (q_1, q_2)$ $F = F_1 \times F_2$

$w \in L(M)$ iff $(\delta_1^*(q_1,w), \delta_2^*(q_0,w))$ iff $\delta_1^*(q_1,w) \in F_1$, $\delta_2^*(q_2,w) \in F_2$ iff $w \in L_1$ e $w \in L_2$

Regulární výrazy

Použití

- V počítačových jazycích lze rozlišit části, tzv. *lexikální elementy*, *tokeny*, které jsou slova z regulárního jazyka. Jsou to např. zápisy konstant, zápisy identifikátorů, klíčová slova, relační znaménka apod. Syntaktický analyzátor (parser) obsahuje lexikální analyzátor, který funguje jako konečný automat. Lexikální analyzátor při každém svém vyvolání přečte ze vstupního souboru nejdelší část, která tvoří jeden lexikální element a volajícímu programu vrátí jeho typ a hodnotu. Zbytek syntaktického analyzátoru může díky tomu být jednodušší, neboť pracuje už jen s lexikálními elementy.

Regulární výrazy

- Vyhledávání výskytů slov daných regulárním výrazem.

  Program grep opisuje na výstup ty řádky vstupního souboru, které obsahují podřetězec popsaný regulárním výrazem.

- Mnohé editory dovedou při operacích „vyhledej“, popř. „vyhledej a nahraď“ hledat podle daného regulárního výrazu.

  V Perl, AWK, Ruby je vyhledávání (a popř. náhrada) vzoru daného regulárním výrazem součástí jazyka. V mnoha dalších jazycích jsou k dispozici knihovny.

- Vyhledávání vzorů se také využívá v antivirech a v antispamových filtrech.

Regulární výrazy

Definice

Je dána abeceda Σ. Množina všech **regulárních výrazů** nad Σ je definována induktivně takto:

- $\emptyset$ je regulární výraz,
- ε je regulární výraz,
- a je regulární výraz pro každé písmeno $a \in \Sigma$,
- jsou-li r_1 a r_2 regulární výrazy, pak $(r_1 + r_2)$, $r_1 r_2$ a $r_1^\star$ jsou regulární výrazy.

$\Sigma = \{a, b, c\}$
$\Sigma^* \leftarrow (a+b+c)^*$ $\Sigma = \{a \ldots z\} \rightarrow$ podobných jako $[a-z]^*$

Regulární výrazy

Jazyk reprezentující regulární výraz.

Každý regulární výraz **r** nad abecedou Σ reprezentuje jazyk L_r nad abecedou Σ:

- $\emptyset$ reprezentuje jazyk $\emptyset$.
- ε reprezentuje jazyk $\{\varepsilon\}$.
- Je-li $a \in \Sigma$, pak a reprezentuje jazyk $\{a\}$.
- Jestliže r_1 reprezentuje jazyk L_1 a r_2 reprezentuje jazyk L_2, pak $r_1 + r_2$ reprezentuje $L_1 \cup L_2$
  a $r_1 r_2$ reprezentuje $L_1 L_2$.
- Jestliže r reprezentuje jazyk L, pak $r^\star$ reprezentuje $L^\star$.

Regulární výrazy Regular expression

Věta.

Každý jazyk reprezentovaný regulárním výrazem je regulární.

Kleeneho věta.

Každý jazyk přijímaný konečným automatem je možné reprezentovat regulárním výrazem.

Regulární výrazy

Definice.

Řekneme, že dva regulární výrazy r a q jsou ekvivalentní, jestliže jimi reprezentované jazyky jsou stejné. Fakt, že regulární výrazy r a q jsou ekvivalentní zapisujeme $r \eqsim q$.

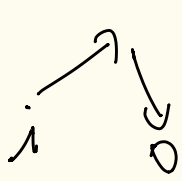
Některé ekvivalentní regulární výrazy.

Jsou-li r, p a q regulární výrazy, pak

- $p + q \eqsim q + p$,
- $(p + q)r \eqsim p\,r + q\,r$, $r(p + q) \eqsim r\,p + r\,q$,
- $(r^\star)^\star = r^\star$, $(\varepsilon + r)\,(\varepsilon + r)^\star \eqsim r^\star$,
- $(p + q)^\star \eqsim (p^\star q^\star)^\star$,

D. Kleeneho věty

$M=(Q,\Sigma,\delta,q_0,F)$ $\quad Q=\{1,2,3,\dots,n\}$ $\quad q_0=1$

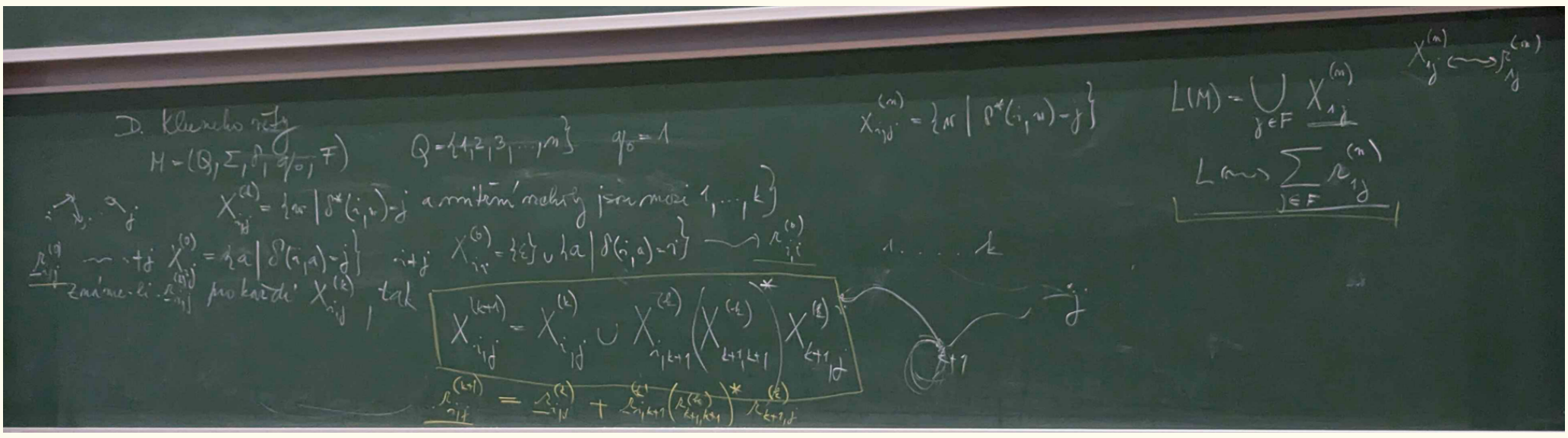


$X_{ij}^{(n)}=\{w \mid \delta^*(i,w)=j$ a mezitémi uzly jsou mezi $1,\dots,k\}$

tj. $X_{ij}^{(0)}=\{a \mid \delta(i,a)=j\}$ $i\neq j$ $\quad X_{ij}^{(k)}=\{\varepsilon\}\cup\{a \mid \delta(i,a)=i\}$

$X_{ij}^{(n)}=\{w \mid \delta^*(i,w)=j\}$ $\quad L(M)=\bigcup_{j\in F} X_{1j}^{(n)}$ $\quad X_{ij}^{(n)} \hookrightarrow R_{ij}^{(n)}$

$L \hookrightarrow \sum_{j\in F} R_{1j}^{(n)}$



Regulární výrazy

Některé ekvivalentní regulární výrazy – II

- $(p + q)^\star \between (p^\star + q)^\star$,
- $(p + q)^\star \between (p^\star q)^\star p^\star$,
- $r^\star \between \varepsilon + rr^\star$,
- $rr^\star \between r^\star r$,
- $(p\,q)^\star \between \varepsilon + p\,(q\,p)^\star\,q$,
- $(p\,q)^\star\,p \between p\,(q\,p)^\star$,
- $r\,\emptyset \between \emptyset \between \emptyset\,r$.

Regulární výrazy

Konstrukce konečného automatu k regulárnímu výrazu.

I. Pomocí důkazu Kleeneho věty.

II. Přímou metodou.

Regulární výrazy

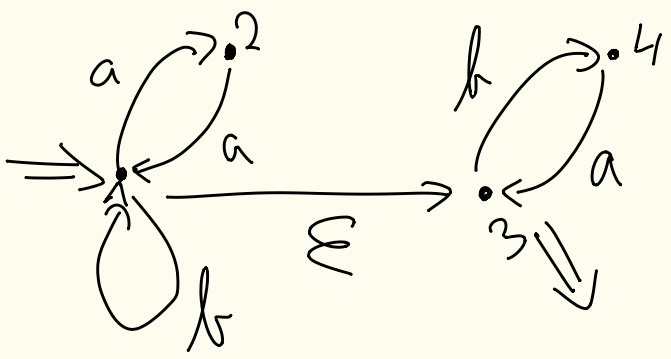
Příklad.

Pro regulární výraz

$$r = (aa + b)^\star(ba)^\star.$$

sestrojte DFA, který přijímá jazyk reprezentovaný tímto regulárním výrazem.

$r = \underbrace{(aa + b)^*}\,\underbrace{(ba)^*}$



		ε	a	b
$\Rightarrow$	1	3	2	1
	2	$\emptyset$	1	$\emptyset$
$\Leftarrow$	3	$\emptyset$	$\emptyset$	4
	4	$\emptyset$	3	$\emptyset$

$\varepsilon\text{-}uz(1) = \{1,3\}$

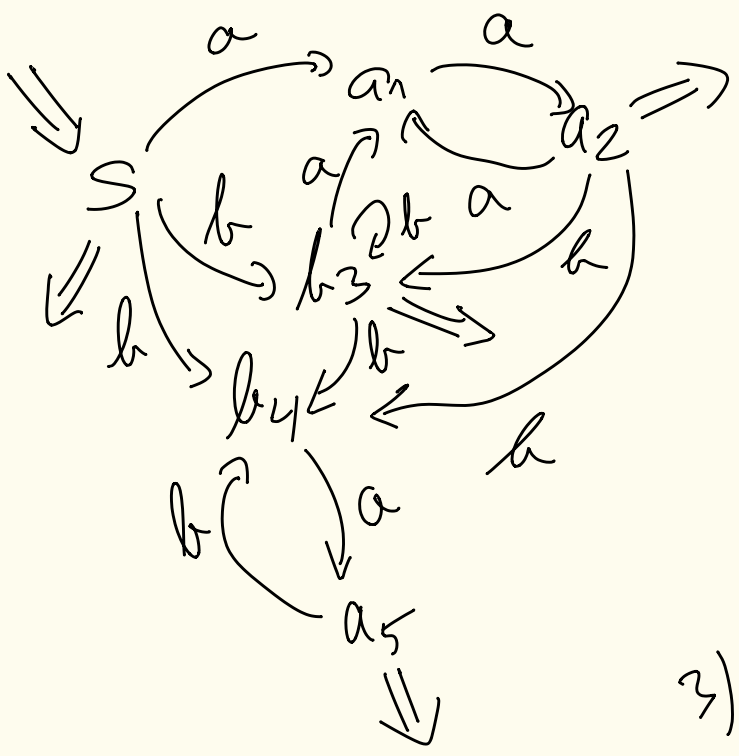
$\Rightarrow 1,3$	

2. metoda

$(a_1a_2 + b_3)^*(b_4a_5)^*$

1) začnat může a_1, b_3, b_4

2) následovat po sobě může:



$a_1 : a_2$

$a_2 : a_1, b_3, b_4$

$b_3 : a_1, b_3, b_4$

$b_4 : a_5$

$a_5 : b_4$

3) může končit:

a_2, b_3, a_5

4) je ε v jazyce? ano

$\varepsilon \in L_R$

Regulární výrazy

Regulární výraz reprezentující jazyk $L(M)$.

Ke stavovému diagramu přidáme

- nový počáteční stav *start* a nový koncový stav *fin*,
- hranu ze *start* do počátečního stavu automatu a hrany z koncových stavů do *fin*, všechny označené ε.

S takto vzniklým orientovaným grafem provádíme úpravy:

a) Odstranění paralelních hran.

b) Odstranění smyčky.

c) Odstranění vrcholu q.

Končíme, když zbyde *start* a *fin* a jediná hrana. Ta je ohodnocena hledaným regulárním výrazem.

Regulární výrazy

Regulární výraz reprezentující jazyk $L(M)$ – příklad.

Příklad. Vytvořte regulární výraz, který reprezentuje jazyk přijímaný

	a	b
$\rightarrow 1$	1	2
$\leftarrow 2$	3	2
$\leftarrow 3$	1	3

Regulární výrazy

Praktické využití regulárních výrazů.