FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Teoretická informatika (TIN) – 2019/2020
Úkol 1

Jakub Svoboda (xsvobo0z)

27. 10. 2019

1. Příklad číslo 1

$L_1 \circ L_2 = L_1 \cup \overline{L_2}$

a) $L_1, L_2 \in \mathcal{L}_3 \Rightarrow L_1 \circ L_2 \in \mathcal{L}_3$

Třída regulárních jazyků je uzavřená vzhledem k operaci sjednocení (věta 3.22 ze skript). Třída regulárních jazyků je také uzavřena vzhledem k operaci doplněk (důkaz věty 3.23 krok číslo 2 ve skriptech). Jelikož se operace $\circ$ skláda pouze z těchto dvou operací, je pak zřejmé, že vztah $L_1, L_2 \in \mathcal{L}_3 \Rightarrow L_1 \circ L_2 \in \mathcal{L}_3$ neporušuje uzávěrové vlastnosti pro $\mathcal{L}_3$ a je tedy platný.

b) $L_1 \in \mathcal{L}_{3,}\ L_2 \in \mathcal{L}^D_{2,} \Rightarrow L_1 \circ L_2 \in \mathcal{L}^D_2$

Výraz upravíme na ekvivalentní tvar:

1. $L_1 \circ L_2$
2. $L_1 \cup \overline{L_2}$
3. $\overline{\overline{L_1 \cup \overline{L_2}}}$
4. $\overline{\overline{L_1} \cap \overline{\overline{L_2}}}$
5. $\overline{\overline{L_1} \cap L_2}$

Třída regulárních jazyků je uzavřena vzhledem k operaci doplněk (důkaz věty 3.23 krok číslo 2 ve skriptech). Věta 4.27 značí, že deterministické bezkontextové jazyky jsou uzavřené vůči průniku s regulárními jazyky a vůči doplňku. Výše zmíněný tvar tedy obsahuje operace, které jsou uzavřené vůči dané třídě jazyků a vztah $L_1 \in \mathcal{L}_{3,}\ L_2 \in \mathcal{L}^D_{2,} \Rightarrow L_1 \circ L_2 \in \mathcal{L}^D_2$ je tedy platný.

c) $L_1 \in \mathcal{L}_{3,}\ L_2 \in \mathcal{L}_{2,} \Rightarrow L_1 \circ L_2 \in \mathcal{L}_2$

Věta 4.24 říká, že bezkontextové jazyky nejsou uzavřené vůči doplňku. Jelikož zadaný výraz obsahuje doplněk bezkontextového jazyka, je tato uzávěrová vlastnost porušena.

Důkaz sporem:

Předpokládáme, že zadaný vztah je platný. Nechť L_1, L_2 jsou jazyky nad konečnou abecedou Σ. Za L_1 si zvolíme jazyk $\emptyset$, který je regulární a je tedy možné vytvořit konečný automat, který tento jazyk přijímá (věta 3.8 ze skript). S tímto dosazením lze výraz $\emptyset \cup \overline{L_2} \in \mathcal{L}_2$ zjednodušit na $\overline{L_2} \in \mathcal{L}_2$, což je v rozporu s větou 4.24 a výraz $L_1 \in \mathcal{L}_{3,}\ L_2 \in \mathcal{L}_{2,} \Rightarrow L_1 \circ L_2 \in \mathcal{L}_2$ je tedy neplatný.

2. Příklad číslo 2

$M_L = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$

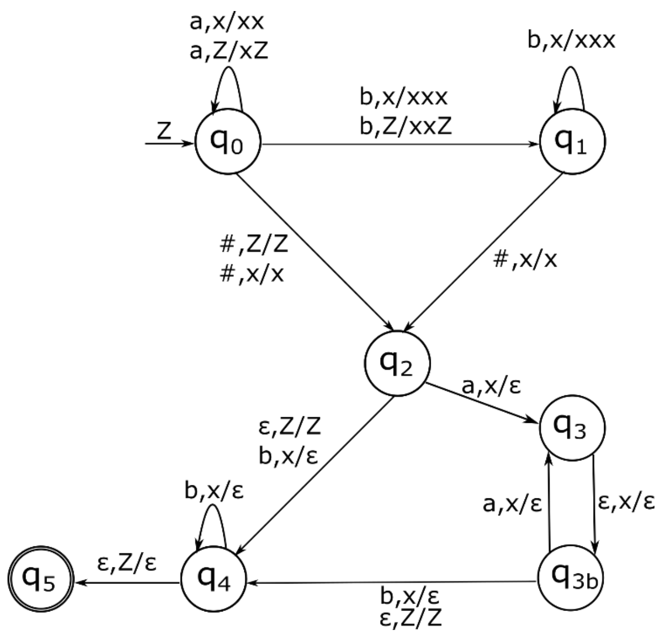
$Q = \{q_0, q_1, q_2, q_3, q_{3b}, q_4, q_5\}$

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

$\Gamma = \{x, Z\}$

$F = \{q_5\}$

δ:
$\delta(q_0, a, Z) = (q_0, xZ)$
$\delta(q_0, a, x) = (q_0, xx)$
$\delta(q_0, b, Z) = (q_1, xxZ)$
$\delta(q_0, b, x) = (q_1, xxx)$
$\delta(q_0, \#, Z) = (q_2, Z)$
$\delta(q_0, \#, x) = (q_2, x)$
$\delta(q_1, b, x) = (q_1, xxx)$
$\delta(q_1, \#, x) = (q_2, x)$
$\delta(q_2, a, x) = (q_3, \varepsilon)$
$\delta(q_2, b, x) = (q_4, \varepsilon)$
$\delta(q_2, \varepsilon, Z) = (q_4, Z)$
$\delta(q_3, \varepsilon, x) = (q_{3b}, \varepsilon)$
$\delta(q_{3b}, a, x) = (q_3, \varepsilon)$
$\delta(q_{3b}, \varepsilon, Z) = (q_4, Z)$
$\delta(q_{3b}, b, x) = (q_4, \varepsilon)$
$\delta(q_4, b, x) = (q_4, \varepsilon)$
$\delta(q_4, \varepsilon, Z) = (q_5, \varepsilon)$



3. Příklad číslo 3

$L = \{a^i b^j \# a^k b^l \mid i + 2j = 2k + l\}$

Věta 3.18 (Ze skript TIN): Nechť L je nekonečný regulární jazyk. Pak existuje celočíselná konstanta $p > 0$ taková, že platí: $w \in L \quad \wedge \quad |w| \geq p \quad \Rightarrow \quad w = xyz \ \wedge$
$y \neq \varepsilon \ \wedge \ |xy| \leq p \ \wedge$
$xy^iz \in L$ pro $i \geq 0$.

Předpokládáme, že zadaný jazyk L je regulární a splňuje tedy větu 3.18.

Zvolme řetězec $w = a^0 b^p \# a^p b^0 \in L$, pro který je splněna podmínka $|w| \geq p$, protože platí $2p + 1 \geq p$.

Při rozdělení řetězce dle pravidel věty 3.18 může nastat pouze jeden případ:

$x = b^k \wedge y = b^l \wedge z = b^{p-k-l}\#a^p \wedge k \geq 0 \wedge l > 0 \wedge k+l \leq p$ pro $k, l \in \mathbb{N}_0$

$xy^iz = b^k(b^l)^i b^{p-k-l}\#a^p = b^{(l*i)+p-l}\#a^p \notin L$ pro všechna $i \neq 1$, protože nesplňuje podmínku jazyka L.

Z tohoto stavu vyplývá, že jazyk L nesplňuje podmínky věty 3.18 a není tedy regulární.

4. Příklad číslo 4

a) Podmínka: $\forall w \in L(A) : |w| \geq 5$.

Algoritmus implementuje vyhledávání do šířky s omezením hloubky (maximální hloubka pro tento případ je rovna pěti). Algoritmus vytvoří frontu a zařadí do ní stav q_0. U každého prvku v této frontě je také uložena informace o hloubce. Algoritmus poté vybírá prvky z fronty a ověřuje, zda stav uložený v tomto prvku je koncový stav. Pokud je tato podmínka splněna a hloubka je menší než 5, pak existuje řetězec porušující výše zmíněnou podmínku. Pokud takovýto řetězec není nalezen do dané hloubky nebo ve frontě již nejsou žádné prvky, je algoritmus ukončen a podmínka je tak splněna.

Vstup: Nedeterministický konečný automat $A = (Q, \Sigma, \delta, q_0, F)$.

Výstup: Pravdivostní hodnota značící splnění nebo rozpor výše zmíněné podmínky.

Algoritmus:

1. Vytvoř frontu Queue.
2. Označ q_0 jako objevené.
3. Zařaď q_0 do fronty s označením hloubky 0.
4. Dokud není Queue prázdná:
5. v = vyřaď z Queue
6. h = hloubka v
7. Pokud je $h \geq 5$:
8. Vrať True.
9. Pokud $v \in F$:
10. Vrať False.
11. Pro každý přímý přechod z v do stavu w: //$w \in Q : \exists\, a \in \Sigma \wedge w \in \delta(v,a)$
12. Pokud w není objevené:
13. Zařaď w do Queue s označením hloubky h+1.
14. Vrať True.

b)

1.	v = null	h = null	Queue: $(q_0, 0)$	Objevené: null
2.	$v = q_0$	h = 0	Queue: $(q_1,1)$	Objevené: q_0
3.	$v = q_1$	h = 1	Queue: $(q_2,2)$	Objevené: q_0, q_1
4.	$v = q_2$	h = 2	Queue: $(q_3,3)$	Objevené: q_0, q_1, q_2
5.	$v = q_3$	h = 3	Queue: $(q_4,4)$	Objevené: q_0, q_1, q_2, q_3
6.	$v = q_4$	h = 4	Queue: null	Objevené: q_0, q_1, q_2, q_3

v $(q_4) \in F$ a zároveň hloubka $h < 5$. Algoritmus navrátí False, což značí porušení výše zmíněné podmínky.

5. Příklad číslo 5

a) $\forall$ u, v $\in \Sigma^*$: u $\sim_L$ v $\Leftrightarrow$ ($\#_a(u) \bmod 2 = \#_a(v) \bmod 2) \wedge [(\#_b(u) > 2 \wedge \#_b(v) > 2) \vee (\#_b(u) \leq 2 \wedge \#_b(v) \leq 2) \wedge (\#_b(u) = \#_b(v))]$

b) $\Sigma^* / \sim_L$:

Rozklad je tvořen následujícími třídami:

$L_1 = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = 0 \wedge \#_b(w) = 0\}$
$L_2 = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = 0 \wedge \#_b(w) = 1\}$
$L_3 = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = 0 \wedge \#_b(w) = 2\}$
$L_4 = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = 0 \wedge \#_b(w) > 2\}$
$L_5 = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = 1 \wedge \#_b(w) = 0\}$
$L_6 = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = 1 \wedge \#_b(w) = 1\}$
$L_7 = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = 1 \wedge \#_b(w) = 2\}$
$L_8 = \{w \in \{a, b\}^* \mid \#_a(w) \bmod 2 = 1 \wedge \#_b(w) > 2\}$

c) Jazyk L je tvořený sjednocením výše zmíněných tříd:

$L = L_5 \cup L_6 \cup L_7$

a to z důvodu, že řetězce těchto tříd splňují logickou formuli z definice jazyka L.

Protože má rozklad konečný počet tříd (index = 8), musí pro jazyk L nutně existovat nějaký DKA, kde L(M) = L (věta 3.20 ze skript) a jazyk L je tedy regulární.