Teoretická informatika, FIT, VUT Brno

1. domácí úloha

David Mihola
xmihol00

29. října 2023

1 Náležitost jazyků do $\mathcal{L}_3$

Následující sekce obsahují analýzu průniku a sjednocení jazyků L_1 a L_2 a důkaz jejich náležitosti, respektive nenáležitosti, do $\mathcal{L}_3$.

1.1 Počty symbolů v jazyce L_1

Počty symbolů v libovolném řetězci jsou vždy stejné, jako počty symbolů v jeho reverzi. Je tedy zřejmé, že počty jednotlivých symbolů v řetězcích náležících do L_1 budou sudé, protože $\forall u \in L_1\ \exists w \in \{a, b, c\}^*\ \forall x \in \{a, b, c\} : u = ww^R \Rightarrow \#_x(u) = 2 \cdot \#_x(w)$.

1.2 Počty symbolů v jazyce L_2

Z definice jazyka L_2 naopak plyne, že $\forall w \in \mathrm{L}_2\ \exists i, j, k \in \mathbb{N} : \#_a(w) = 2i + 1 \wedge \#_b(w) = 2j + 1 \wedge \#_c(w) = k$. Tzn., počty symbolů a a b v řetězcích náležících do tohoto jazyka jsou liché a počty symbolů c jsou větší nebo rovny nule.

1.3 Průnik jazyků L_1 a L_2

Z předchozí analýzy počtů symbolů v jazycích L_1 a L_2 je zřejmé, že průnik těchto jazyků je prázdná množina. Tento jazyk je zřejmě regulární, lze pro něj sestrojit např. následující regulární gramatiku $G_1 = (\{S\}, \{a, b, c\}, \emptyset, S)$.

1.4 Sjednocení jazyků L_1 a L_2

U sjednocení jazyků L_1 a L_2 budu dokazovat, že není regulární. Není tedy nutné přesně znát všechny řetězce, které do něj patří. Při důkazu je pouze nutné s jistotou říci, jestli daný řetězec patří do sjednocení L_1 a L_2, nebo nepatří.

1.4.1 Důkaz neregularity sjednocení jazyků L_1 a L_2

Před samotným důkazem je dobré zmínit, že nestačí dokázat neregularitu jazyků L_1 a L_2 samostatně[1]. Je nutné dokázat nenáležitost celého sjednocení L_1 a L_2 do $\mathcal{L}_3$, protože i sjednocením dvou neregulárních jazyků lze získat regulární jazyk.
Pro důkaz neregularity použiji obměněnou implikaci s *pumping lemma* pro $\mathcal{L}_3$ a na základě ní provedu přímý důkaz[2]. Obměněná implikace pro jazyk $L_1 \cup L_2$ vypadá následovně:
$(\forall k \in \mathbb{N}^+ : \exists w \in \Sigma^* : w \in L_1 \cup L_2 \wedge |w| \geq k \wedge \forall x, y, z \in \Sigma^* : xyz = w \wedge y \neq \varepsilon \wedge |xy| \leq k \Rightarrow \exists i \in \mathbb{N} : xy^iz \notin L_1 \cup L_2) \Rightarrow L_1 \cup L_2 \notin \mathcal{L}_3$.
Samotný důkaz je poté veden v následující posloupnosti kroků:

[1] I když zde bych neuspěl, protože L_2 je zjevně regulární jazyk.

[2] Obměnou dojde k znegování obou stran implikace. Důkaz má tedy podobný průběh jako důkaz sporem vycházející z původní implikace.

1. I když nelze dokazovat neregularitu jazyků samostatně, lze při důkazu vhodným výběrem slova w jazyk L_2 úplně ignorovat. Takové slovo zřejmě nebude obsahovat symboly a nebo b. Proto pro důkaz zvolím $w = a^k cca^k$. Pro vybrané w platí: $(w \in L_1 \Rightarrow w \in L_1 \cup L_2) \wedge (|w| = 2k + 2 \geq k)$.

2. Všechna možná rozdělení slova w na x, y, z popíši pomocí parametrů m, n, pro které platí: $m, n \in \mathbb{N} \wedge m \geq 0 \wedge n > 0 \wedge m + n \leq k$. Z vybraného slova je zřejmé, že podřetězce x a y budou obsahovat pouze symboly a. Všechna možná rozdělení lze tedy popsat následovně:
   - $x = a^m$,
   - $y = a^n$,
   - $z = a^{k-m-n} cca^k$.

3. Nyní zvolím $i = 0$. Pak slovo $xy^0z = a^{k-n}cca^k$ zajisté nepatří do L_2, protože neobsahuje žádný symbol b. Nepatří ale ani do L_1, protože nyní byl z jeho první poloviny odebrán alespoň jeden symbol a a není možné získané slovo rozdělit tak, aby první symbol c byl na konci nějakého slova α, druhý symbol c na začátku α^R a aby $|\alpha| = |\alpha^R|$. Tudíž $xy^0z \notin L_1 \cup L_2$.

4. V předchozím bodě jsem dokázal, že platí levá strana obměněné implikace. Aby byl výrok daný implikací pravdivý, musí platit i její pravá strana.

5. Z předchozích bodů plyne, že $L_1 \cup L_2 \notin \mathcal{L}_3$. $\square$

2 Bezkontextová gramatika G_3 a zásobníkový automat Z_3

Následující sekce obsahují postup tvorby bezkontextové gramatiky G_3 a zásobníkového automatu přijímajícího vyprázdněním zásobníku Z_3 pro jazyk $L_3 = \{puvw \mid p, v \in \{a, b\}^* \wedge u, w \in \{c, d\}^* \wedge (p = v^R \vee u = w^R)\}$.

2.1 Bezkontextová gramatika G_3

$G_3 = (N, T, P, S)$, kde:

- $T = \{a, b\} \cup \{c, d\} = \{a, b, c, d\}$,
- pro množinu pravidel P bude platit následující:
  - obsahuje pravidla, podle kterých se rozhodne, jestli se generuje řetězec $v^R uvw$ ($\{S \rightarrow OU\}$) nebo $pw^R vw$ ($\{S \rightarrow QR\}$)[3],
  - následně obsahuje pravidla, která umožňují vygenerovat větné formy $v^R UvU$ ($\{O \rightarrow aOa, O \rightarrow bOb, O \rightarrow U\}$) a $Qw^R Qw$ ($\{R \rightarrow cRc, R \rightarrow dRd, R \rightarrow Q\}$),

[3]Pokud platí oba výroky, pak lze vybrat libovolné z těchto dvou pravidel.

- nakonec jsou v P obsažena pravidla pro generování libovolného řetězce[4] z $\{a, b\}^*$ ($\{Q \rightarrow aQ, Q \rightarrow bQ, Q \rightarrow \varepsilon\}$) a z $\{c, d\}^*$ ($\{U \rightarrow cU, U \rightarrow dU, U \rightarrow \varepsilon\}$),

celkově tedy $P = \{S \rightarrow OU, S \rightarrow QR, O \rightarrow aOa, O \rightarrow bOb, O \rightarrow U, R \rightarrow cRc, R \rightarrow dRd, R \rightarrow Q, Q \rightarrow aQ, Q \rightarrow bQ, Q \rightarrow \varepsilon, U \rightarrow cU, U \rightarrow dU, U \rightarrow \varepsilon\}$,

- na základě výchozího nonterminálního symbolu a nonterminálních symbolů použitých v pravidlech musí platit, že $N = \{O, Q, R, S, U\}$.

2.2 Zásobníkový automat Z_3

Zásobníkový automat Z_3 přijímající jazyk L_3 vyprázdněním zásobníku jsem konstruoval za použití přednášeného algoritmu konstrukce zásobníkového algoritmu přijímajícího vyprázdněním zásobníku převodem z libovolné bezkontextové gramatik. V tomto případě šlo o bezkontextovou gramatiku G_3. $Z_3 = (Q, \Sigma, \Gamma, \delta, q, S, \emptyset)$, kde:

- $Q = \{q\}$,
- $\Sigma = T = \{a, b, c, d\}$,
- $\Gamma = T \cup N = \{a, b, c, d, O, Q, R, S, U\}$,
- $\delta = \{((q, \varepsilon, S), (q, OU)), ((q, \varepsilon, S), (q, QR)),$
  $((q, \varepsilon, O), (q, aOa)), ((q, \varepsilon, O), (q, bOb)), ((q, \varepsilon, O), (q, U))$
  $((q, \varepsilon, R), (q, cRc)), ((q, \varepsilon, R), (q, dRd)), ((q, \varepsilon, R), (q, Q))$
  $((q, \varepsilon, Q), (q, aQ)), ((q, \varepsilon, Q), (q, bQ)), ((q, \varepsilon, Q), (q, \varepsilon))$
  $((q, \varepsilon, U), (q, cU)), ((q, \varepsilon, U), (q, dU)), ((q, \varepsilon, U), (q, \varepsilon))$
  $((q, a, a), (q, \varepsilon)), ((q, b, b), (q, \varepsilon)), ((q, c, c), (q, \varepsilon)), ((q, d, d), (q, \varepsilon))\}$.

3 Platnost výroků – uzávěrové vlastnosti

Následující sekce dokazují platnost, nebo neplatnost zadaných výroků.

3.1 $\exists L_1 \in \mathcal{L}_2 \setminus \mathcal{L}_3 \Rightarrow \overline{L_1} \in \mathcal{L}_{Fin}$

Tento výrok neplatí. Neplatnost dokáži sporem. Pro důkaz uvažujme $L_C = \overline{L_1} \in \mathcal{L}_{Fin}$.
Důkaz je obsažen v následující posloupnosti kroků:

- Předpokládejme, že $\exists L_1 \in \mathcal{L}_2 \setminus \mathcal{L}_3 \Rightarrow \overline{L_1} \in \mathcal{L}_{Fin}$.
- Každý konečný jazyk je regulární, tzn., $L_C \in \mathcal{L}_3$.
- Uzávěrové vlastnosti $\mathcal{L}_3$ říkají $\forall L \in \mathcal{L}_3 \Rightarrow \overline{L} \in \mathcal{L}_3$, tudíž $L_C \in \mathcal{L}_3 \Rightarrow \overline{L_C} \in \mathcal{L}_3$.

[4] Tzn., i řetězce splňující podmínku $p = v^R$, nebo $u = w^R$.

- $\overline{L_C} = \overline{\overline{L_1}} = L_1 \in \mathcal{L}_3$, to je ale spor s předpokladem.
- Výrok neplatí. $\square$

3.2 $\exists L_1 \in \mathcal{L}_2 \setminus \mathcal{L}_3 \Rightarrow \forall L_2 \in \mathcal{L}_3 : L_1 \cap L_2 \in \mathcal{L}_2 \setminus \mathcal{L}_3$

Výrok neplatí. Zřejmě pro $L_2 = \emptyset$ bude pro každé L_1 platit, že $L_1 \cap \emptyset = \emptyset \in \mathcal{L}_3$.

3.3 $\exists L_1 \in \mathcal{L}_3 \Rightarrow \forall L_2 \in \mathcal{L}_2 \setminus \mathcal{L}_3 : L_1 \cap L_2 \in \mathcal{L}_2 \setminus \mathcal{L}_3$

Výrok platí. Zřejmě pro $L_1 = \Sigma^*$ bude pro každé L_2 platit, že $\Sigma^* \cap L_2 = L_2 \in \mathcal{L}_2 \setminus \mathcal{L}_3$.

4 Relace pravé kongruence ~

Relaci pravé kongruence ~ sestrojím v následujících krocích:

1. Sestrojím úplný DKA přijímající jazyk $L = \{w \in \{a, b\}^* \mid \#_a(w) \geq 2 \vee \#_b(w) = 0\}$. Takový DKA zadán tabulkou je např.:

		a	b
$\leftrightarrow$	0	1A	B+
$\leftarrow$	1A	2A+	AB+
	B+	AB+	B+
$\leftarrow$	2A+	2A+	2A+
	AB+	2A+	AB+

2. Minimalizuji tento automat následovně:

   (a) 0 nerozlišitelnost:

			a	b
I	$\leftrightarrow$	0	I	II
		1A	I	II
		2A+	I	I
II		B+	II	II
		AB+	I	II

   (b) 1 nerozlišitelnost:

			a	b
I	$\leftrightarrow$	0	I	II
		1A	III	IV
III	$\leftarrow$	2A+	III	III
II		B+	IV	II
IV		AB+	III	IV

(c) 2 = 3 nerozlišitelnost:

			a	b
I	$\leftrightarrow$	0	V	II
V	$\leftarrow$	1A	III	IV
III	$\leftarrow$	2A+	III	III
II		B+	IV	II
IV		AB+	III	IV

3. Z bodu 2) je zřejmé, že DKA definovaný v bodě 1) je již minimální. Z vlastností minimálních úplných DKA, prefixové ekvivalence a Myhill-Nerodovy věty plyne, že řetězce dočtené v různých stavech tohoto DKA jsou v různých třídách $\Sigma^*/{\sim_L}$. Těchto tříd je 5, tedy index $\sim_L$ je 5, a platí, že:

$$
\begin{aligned}
u \sim_L v \Leftrightarrow\ & (\#_a(u) = \#_a(v) = \#_b(u) = \#_b(v) = 0) \vee \\
& (\#_a(u) = \#_a(v) = 1 \wedge \#_b(u) = \#_b(v) = 0) \vee \\
& (\#_a(u) \geq 2 \wedge \#_a(v) \geq 2) \vee \\
& (\#_a(u) = \#_a(v) = 0 \wedge \#_b(u) \geq 1 \wedge \#_b(v) \geq 1) \vee \\
& (\#_a(u) = \#_a(v) = 1 \wedge \#_b(u) \geq 1 \wedge \#_b(v) \geq 1)
\end{aligned}
$$

4. Zřejmě přidáním jednoho stavu do dříve definovaného DKA bez změněny přijímaného jazyka budou řetězce dočtené v různých stavech tohoto DKA v různých třídách $\Sigma^*/\sim$ a tím pádem index $\sim$ bude o 1 větší než index $\sim_L$. DKA popsaný tabulkou obsahující stav navíc může vypadat následovně[5]:

		a	b
$\leftrightarrow$	0	1A	B+
$\leftarrow$	1A	2A	AB+
	B+	AB+	B+
$\leftarrow$	2A	3A+	3A+
	AB+	2A	AB+
$\leftarrow$	3A+	3A+	3A+

[5] Od původního DKA se liší přejmenováním stavu 2A+ na 2A, přidáním stavu 3A+ a patřičnou úpravou přechodové funkce.

Tím pádem pro $\sim$ platí, že:

$$
\begin{aligned}
u \sim v \Leftrightarrow\ & (\#_a(u) = \#_a(v) = \#_b(u) = \#_b(v) = 0) \lor \\
& (\#_a(u) = \#_a(v) = 1 \land \#_b(u) = \#_b(v) = 0) \lor \\
& (\#_a(u) = 2 \land \#_a(v) = 2) \lor \\
& (\#_a(u) = \#_a(v) = 0 \land \#_b(u) \geq 1 \land \#_b(v) \geq 1) \lor \\
& (\#_a(u) = \#_a(v) = 1 \land \#_b(u) \geq 1 \land \#_b(v) \geq 1) \lor \\
& (\#_a(u) \geq 3 \land \#_a(v) \geq 3)
\end{aligned}
$$

5. Relace $\sim$ je jistě reflexivní, symetrická, tranzitivní i pravá kongruence, protože byla konstruována na základě úplného DKA.
6. $L = [\varepsilon] \cup [a] \cup [aa] \cup [aaa]$, kde výraz $[x]$ značí ekvivalenčí třídu určenou řetězcem x, $x \in L$.