Prémiové příklady

Příklad 1: Říkáme, že slovo x je *prefixem* slova y, jestliže existuje slovo z takové, že $xz = y$. Slovo x je *vlastním prefixem* slova y, jestliže navíc platí, že $x \neq y$. V každém ze dvou následujících bodů definujeme operaci na jazyce L. Pro obě tyto operace ukažte, že třída regulárních jazyků je vůči nim uzavřená:

a) $\textsc{NoPrefix}(L) = \{w \in L \mid$ žádný vlastní prefix w nepatří do $L\}$

b) $\textsc{NoExtend}(L) = \{w \in L \mid w$ není vlastním prefixem žádného slova z $L\}$

Příklad 2: Mějme nějaký jazyk A. Zápisem $A_{\frac{1}{2}-}$ označme jazyk tvořený prvními polovinami slov z A, tj.

$$A_{\frac{1}{2}-} = \{x \mid \text{existuje } y \text{ takové, že } |x| = |y| \text{ a } xy \in A\}\,.$$

Ukažte, že, jestliže A je regulární, pak i $A_{\frac{1}{2}-}$ je regulární.

Příklad 3: Uveďte příklad posloupnosti jazyků $E_1, E_2, E_3, \ldots$, kde existuje konstanta $c > 1$ taková, že každý jazyk E_n je rozpoznáván nedeterministickým konečným automatem s n stavy, ale jakýkoliv deterministický konečný automat rozpoznávající E_n musí mít alespoň c^n stavů. Dokažte, že vámi uvedená posloupnost jazyků má tuto vlastnost.

Příklad 4: Mějme nějaký jazyk A. Zápisem $A_{\frac{1}{3}-\frac{1}{3}}$ označme jazyk tvořený slovy, která vzniknou ze slov z A odstraněním jejich prostřední třetiny, tj.

$$A_{\frac{1}{3}-\frac{1}{3}} = \{xz \mid \text{existuje } y \text{ takové, že } |x| = |y| = |z| \text{ a } xyz \in A\}\,.$$

Ukažte, že, jestliže A je regulární, pak $A_{\frac{1}{3}-\frac{1}{3}}$ nemusí být nutně regulární.

Příklad 5: Dokažte, že následující jazyky nejsou regulární:

a) $\{0^n 1^m 0^n \mid m, n \geq 0\}$

b) Doplněk jazyka $\{0^n 1^n \mid n \geq 0\}$

c) $\{0^m 1^n \mid m \neq n\}$

d) $\{w \mid w \in \{0,1\}^* \text{ není palindrom}\}$ (Palindrom je slovo, které se čte stejně zleva doprava i zprava doleva, tj. slovo w splňující $w = w^R$.)

Příklad 6: Uveďte bezkontextové gramatiky, které generují následující jazyky:

a) Množina slov nad abecedou $\{\mathtt{a}, \mathtt{b}\}$, kde je dvakrát tolik symbolů `a` než `b`.

b) Doplněk jazyka $\{\mathtt{a}^n \mathtt{b}^n \mid n \geq 0\}$.

c) $\{w\#x \mid w^R \text{ je podslovem } x, \text{ přičemž } w, x \in \{\mathtt{0}, \mathtt{1}\}^*\}$.

d) $\{x_1\texttt{\#}x_2\texttt{\#}\cdots\texttt{\#}x_k \mid k \geq 1$, všechna $x_i \in \{\texttt{a},\texttt{b}\}^*$, a pro nějaká i a j platí $x_i = x_j^R\}$.

Příklad 7: Předpokládejme, že máme dánu nějakou bezkontextovou gramatiku G, jejíž každé pravidlo je jednoho ze dvou následujících typů:

- Na pravé straně jsou dva neterminály (a žádné terminály), například $X \longrightarrow YZ$.
- Na pravé straně je pouze jeden terminál, například $X \longrightarrow \texttt{a}$.

(*Poznámka:* O takové gramatice říkáme, že je v Chomského normální formě.)

Předpokládejme, že gramatika G obsahuje b neterminálů. Ukažte, že pokud v ní existuje derivace nějakého slova, která má více než 2^b kroků, pak je jazyk $L(G)$ nekonečný.

Příklad 8: Mějme gramatiku $G = (\Pi, \Sigma, \langle\textsc{stmt}\rangle, P)$, kde P obsahuje pravidla

$$\begin{array}{rcl}
\langle\textsc{stmt}\rangle & \longrightarrow & \langle\textsc{assign}\rangle \mid \langle\textsc{if-then}\rangle \mid \langle\textsc{if-then-else}\rangle \mid \langle\textsc{begin-end}\rangle \\
\langle\textsc{if-then}\rangle & \longrightarrow & \texttt{if condition then } \langle\textsc{stmt}\rangle \\
\langle\textsc{if-then-else}\rangle & \longrightarrow & \texttt{if condition then } \langle\textsc{stmt}\rangle \texttt{ else } \langle\textsc{stmt}\rangle \\
\langle\textsc{begin-end}\rangle & \longrightarrow & \texttt{begin } \langle\textsc{stmt-list}\rangle \texttt{ end} \\
\langle\textsc{stmt-list}\rangle & \longrightarrow & \langle\textsc{stmt-list}\rangle\langle\textsc{stmt}\rangle \mid \langle\textsc{stmt}\rangle \\
\langle\textsc{assign}\rangle & \longrightarrow & \texttt{a:=1}
\end{array}$$

$\Sigma = \{\texttt{if}, \texttt{condition}, \texttt{then}, \texttt{else}, \texttt{begin}, \texttt{end}, \texttt{a:=1}\}$.
$\Pi = \{\langle\textsc{stmt}\rangle, \langle\textsc{if-then}\rangle, \langle\textsc{if-then-else}\rangle, \langle\textsc{begin-end}\rangle, \langle\textsc{stmt-list}\rangle, \langle\textsc{assign}\rangle\}$.

Gramatika G popisuje určitý fragment nějakého programovacího jazyka. Tato gramatika je ovšem nejednoznačná.

a) Ukažte, že gramatika G je nejednoznačná.

b) Navrhněte, jak tuto gramatiku upravit, aby nejednoznačná nebyla.

Příklad 9: Navrhněte bezkontextové gramatiky pro následující jazyky:

a) $C = \{x\texttt{\#}y \mid x, y \in \{\texttt{0},\texttt{1}\}^*,\ x \neq y\}$

b) $D = \{xy \mid x, y \in \{\texttt{0},\texttt{1}\}^*,\ |x| = |y|,\ x \neq y\}$

Příklad 10: Uveďte neformální popis zásobníkových automatů, které rozpoznávají jazyky z příkladu 6.

Příklad 11: Napište program ve vašem oblíbeném programovacím jazyce (C, C++, Java, Python, ...), který na standardní výstup vypíše svůj vlastní kód, přičemž ovšem nesmí načítat údaje ze žádného vstupu (např. číst ze souboru apod.).

Příklad 12: Na libovolný rozhodovací problém můžeme pohlížet jako na jazyk, do kterého patří ta slova, která jsou zápisem instancí, pro něž je odpověď Ano. Díky tomu se na třídu PTIME můžeme dívat jako na třídu jazyků.

Ukažte, že tato třída je uzavřená vzhledem k iteraci.

Poznámka: Předpokládejte, že vstupy všech problémů jsou kódovány jako slova v abecedě $\{\mathtt{0}, \mathtt{1}\}$.

Příklad 13: Permutace P na množině $M = \{1, 2, \ldots, k\}$ je libovolná bijekce na této množině. Permutace na množině M můžeme skládat: Složením permutací P a Q je permutace R taková, že $R(x) = Q(P(x))$ pro všechna $x \in M$.

Pro libovolné $t \geq 0$ můžeme definovat permutaci P^t:

$$P^t(x) = \begin{cases} x & \text{pro } t = 0 \\ P(P^{t-1}(x)) & \text{pro } t > 0 \end{cases}$$

Ukažte, že následující problém patří do třídy PTIME:

Vstup: Kladné celé číslo k, permutace P a Q na množině $\{1, 2, \ldots, k\}$ a kladné celé číslo t reprezentované binárně.

Otázka: Je $P = Q^t$?

Poznámka: Pozor, nejpřímočařejší algoritmus není polynomiální.

Příklad 14: Ukažte, že následující problém patří do třídy PTIME:

Vstup: Booleovská formule φ v konjunktivní normální formě taková, že každá klauzule obsahuje nejvýše dva literály.

Otázka: Je formule φ splnitelná?