Týden 1

Přednáška

(Tento text se určitým způsobem doplňuje se slidy k přednášce, které jsou také na webu předmětu k dispozici.)

Na začátku jsme si připomněli základní informace o kursu a podmínkách jeho absolvování, které jsou všechny v Edisonu a na web-stránce předmětu.

Zdůraznil jsem ještě tuto věc: Přednáška se v zásadě bude soustřeďovat na ilustraci základních myšlenek z probírané problematiky v součinnosti s aktivně se účastnícími posluchači. Nepůjde o promítání a recitaci toho, co mají všichni k dispozici v základním studijním textu (na nějž se vždy v popisu průběhu výuky odkazujeme, není-li řečeno jinak) a v příslušných animacích. Je **velmi žádoucí**, aby si posluchači vždy sami prošli příslušné odkazované partie v textu, uvědomili si případné problémy a nejasnosti a připravili se na cvičení, kde je primární prostor pro vyřešení případných problémů a nejasností. (Cvičení pochopitelně nemůže nahradit vaše samostudium.)

Konečné automaty

Připomněli jsme si pojem konečného automatu jako (jednoduchého) *modelovacího nástroje* a ukázali souvislost s programy s „malou“ pamětí.

Konečný automat při vyhledávání vzorku v textu

Zkonstruovali jsme konečný automat, který je základem (efektivního) algoritmu pro vyhledání všech výskytů vzorku *abaaba* v (dlouhém) řetězci písmen z abecedy $\{a, b\}$. Řešili jsme tedy „úkol“ U_0 z části 3.1. (studijního textu). Dospěli jsme ke konečnému automatu se stavy $U_0, U_1, \ldots, U_6$. Při konstrukci se může osvědčit značení

$U_0 \ldots |abaaba \ldots 0$ (potenciálně jsme právě přečetli “0-tou pozici” vzorku)

$U_1 \ldots |a|baaba \ldots 0, 1$ (potenciálně jsme právě přečetli 1. nebo 0. pozici vzorku)

$\ldots$

$U_6 \ldots |a|ba|aba| \ldots 0, 1, 3, 6$ (přečetli jsme 6., 3., 1., nebo 0. pozici vzorku)

Použili jsme obecný algoritmus zachycený tímto pseudokódem (pro vstup *patt*: array[1..d] of char):

$Next[0, patt[1]] := 1$; $\forall x \in \Sigma \setminus \{patt[1]\} : Next[0, x] := 0$; $Sec[1] := 0$;

for $i := 1$ to $d-1$ do

$Next[i, patt[i+1]] := i+1$;
$\forall x \in \Sigma \setminus \{patt[i+1]\} : Next[i, x] := Next[Sec[i], x]$;
$Sec[i+1] := Next[Sec[i], patt[i+1]]$;

$\forall x \in \Sigma : Next[d, x] := Next[Sec[d]; x];$

Demonstrovali jsme tak de facto variantu algoritmu "Knuth - Morris - Pratt". Jak je více ilustrováno na slidech, KMP-algoritmus nesestrojuje tento automat explicitně, klíčová je funkce Sec ... Také jsme si uvědomili několik souvislostí ohledně časové složitosti algoritmu ...

Připomněli jsme si pak definici konečného automatu jako matematické struktury

$$A = (Q, \Sigma, \delta, q_0, F)$$

a definici jazyka rozpoznávaného (přijímaného) automatem A

$$L(A) = \{w \in \Sigma^* \mid \text{ slovo } w \text{ je přijímáno } A\} = \{w \in \Sigma^* \mid q_0 \xrightarrow{w} F\}$$

a uvědomili jsme si, že jsme vlastně zkonstruovali automat přijímající jazyk

$$\{\, w \in \{a, b\}^* \mid w \text{ má sufix } abaaba \,\}\,.$$

Modulární postup při návrhu konečných automatů

Zkonstruovali jsme konečný automat (resp. uvedli jsme hlavní myšlenky příslušné konstrukce) rozpoznávající jazyk

$$L_0 = \{w \in \{0, 1\}^* \mid bn(w) \text{ je dělitelné třemi a } w \text{ neobsahuje podřetězec } 101\}\,,$$

kde $bn(w)$ znamená číslo, pro něž je w jeho binárním zápisem (např. $bn(0101) = 5$; definujeme také $bn(\varepsilon) = 0$).
Postupovali jsme tak, že jsme nejdříve zkonstruovali (4-stavový) automat A_2' pro jazyk

$$L_2' = \{w \in \{0, 1\}^* \mid w \text{ obsahuje podřetězec } 101\}\,,$$

ten jsme transformovali na (4-stavový) automat A_2 rozpoznávající doplněk jazyka L_2', tj. jazyk

$$L_2 = \{0, 1\}^* - L_2' = \{w \in \{0, 1\}^* \mid w \text{ neobsahuje podřetězec } 101\}\,,$$

pak jsme zkonstruovali (3 stavový) A_1 pro jazyk

$$L_1 = \{w \in \{0, 1\}^* \mid bn(w) \mod 3 = 0\}$$

(uvědomili jsme si, že $bn(uv) = bn(u) \cdot 2^{|v|} + bn(v)$ (speciálně $bn(u0) = bn(u) \cdot 2$ a $bn(u1) = bn(u) \cdot 2 + 1$), takže z dosud přečteného prefixu u si stačí pamatovat hodnotu $bn(u) \mod 3$) a na závěr jsme naznačili konstrukci „kartézského součinu", která k automatům A_1, A_2 vyrobí (12-stavový) automat A rozpoznávající $L(A_1) \cap L(A_2) = L_0$.

Induktivní definice

Pobavili jsme se o induktivních definicích a ilustrovali jsme příkladem zavedení značení (či ternárního predikátu) $q \xrightarrow{w} q'$, jak uvedeno v „matematické poznámce“ na str. 58.

(Levý) kvocient jazyka podle slova

Uvědomili jsme si, že konstrukce konečných automatů úzce souvisí s operací (levého) kvocientu jazyka podle slova. Uvedli jsme definici

$$u\backslash L = \{v \mid uv \in L\}$$

a vyslovili následující větu (3.18 z části 3.8., str. 87).

Věta. Jazyk $L \subseteq \Sigma^*$ je regulární (tzn. přijímaný konečným automatem) právě tehdy, když je množina kvocientů $\{w\backslash L \mid w \in \Sigma^*\}$ konečná.

O důkazu jsme zatím nehovořili.

Intuici k poznání regulárních jazyků jsme si posílili příklady (na slidech).

Partie textu k prostudování

Posluchači by si měli projít Kapitolu 2 (Formální jazyky) a sekce 3.1., 3.2., 3.3. (Tzn. udělat si přinejmenším dobrou první představu a zamyslet se nad příklady, speciálně těmi plánovanými na cvičení, ať se mohou na cvičení aktivně účastnit a případné problémy si tam objasnit.)

Cvičení

Příklad 1.1

Vypište prvních deset slov z jazyka $L = \{w \in \{a, b\}^* \mid$ každý výskyt podslova aa je ve w ihned následován znakem $b\}$. Odkazujeme se k uspořádání $<_L$, tedy slova máme uspořádána (vzestupně) podle délky a v rámci stejné délky abecedně, přičemž předpokládáme abecední uspořádání $a < b$.

Příklad 1.2

Definujte operaci zřetězení jazyků $L_1 \cdot L_2$.
Připomeňte si, že pro jazyk L lze definovat L^* (iterace jazyka L) např. takto:
$L^* = \{w \mid$ existuje n a slova $v_1, v_2, \ldots, v_n$ v L tak, že $w = v_1 v_2 \ldots v_n\}$
(z případu $n = 0$ vyplývá, že $\varepsilon \in L^*$).

Příklad 1.3

Charakterizujte slova jazyka $L = L_0 \cup L_1$, kde L_0 je množina všech slov obsahujících více 0 než 1 (tedy $L_0 = \{ w \in \{0,1\}^* \mid |w|_0 > |w|_1 \}$) a L_1 je množina všech slov obsahujících více 1 než 0 (tedy $L_1 = \{ w \in \{0,1\}^* \mid |w|_0 < |w|_1 \}$).
Jaká slova patří do iterace jazyka L, tedy do L^* ?

Příklad 1.4

Jaká slova patří do jazyka L_1^R, tj. jazyka, který je zrcadlovým obrazem jazyka $L_1 = \{\varepsilon, a, abb, baaba\}$?
Jaký je jazyk L_2^R pro $L_2 = \{w \mid |w|_a \bmod 2 = 0\}$?
Platí pro tyto jazyky $(L_1 L_2)^R = (L_1)^R (L_2)^R$?
Platí $(L_1 L_2)^R = (L_2)^R (L_1)^R$? Platí tento vztah obecně pro jakékoli jazyky L_1, L_2 ?

Příklad 1.5

Navrhněte konečný automat A tak, že $L(A) = L$, kde $L = \{w \in \{a,b\}^* \mid$ každý výskyt podslova aa je ve w ihned následován znakem $b\}$.
Charakterizujte jazyky $a\backslash L$, $b\backslash L$.

Příklad 1.6

Zkonstruujte modulárně konečný automat přijímající jazyk
$L = \{ w \in \{0,1\}^* \mid w$ obsahuje podslovo 010 nebo $|w|_1$ je sudé $\}$.
Později se zamyslíme, zda půjde počet stavů výsledného automatu zmenšit.

Příklad 1.7

Zjistěte, zda obecně platí některý ze vztahů
$(uv)\backslash L = u\backslash(v\backslash L)$,
$(uv)\backslash L = v\backslash(u\backslash L)$.
Dokažte svá zjištění.

Příklad 1.8

Vybudujte si intuici o (ne)regulárních jazycích promyšleným zodpovězením otázek na s. 88 a 89. Ty se týkají (ne)regularity následujících jazyků. (Zápisem w^R značíme zrcadlový obraz [reversal] slova w; induktivně lze definovat takto: $\varepsilon^R = \varepsilon$; $(au)^R = u^R a$.)
$\{w \in \{a,b\}^*; |w|_a \bmod 2 = 0\}$
$\{w \in \{a,b\}^*; w$ začíná nebo končí dvojicí stejných písmen $\}$
$\{w \in \{a,b\}^*; |w|_a < |w|_b\}$

$\{w \in \{a,b,c\}^*$; jestliže w neobsahuje podřetězec abc, pak končí $bca\}$
$\{w \in \{a,b\}^*; |w|_a > |w|_b$ nebo w končí $baa\}$
$\{w \in \{a,b\}^*; |w|_a > |w|_b$ nebo $|w|_b \geq 2\}$
$\{u$; ex. $w \in \{a,b\}^*$ tak, že $u = ww^R\}$, stručněji také psáno $\{ww^R; w \in \{a,b\}^*\}$,
$\{w \in \{a,b\}^*; w = w^R\}$
$\{w \in \{a\}^*; w = w^R\}$
$\{ww; w \in \{a,b\}^*\}$
$\{ww; w \in \{a\}^*\}$
$\{w \in \{a,b\}^*$; rozdíl počtů znaků a a znaků b ve w je větší než 100$\}$
$\{w \in \{a,b\}^*$; součin $|w|_a$ a $|w|_b$ je větší nebo roven 100$\}$
$\{w \in \{a\}^*; |w|$ je prvočíslo $\}$