

## Týden 1

### Přednáška - první část

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 diskusi těchto problémů a nejasností. (Cvičení nenahrazuje vaše samostudium!)

Druhá část přednášky bude typicky zamýšlena pro studenty s hlubším zájmem o matematické pozadí, důkazy vybraných tvrzení apod. Mj. bych zde předpokládal účast všech studentů, kteří potenciálně uvažují o PhD studiu v informatice a/nebo budou mít zájem o předmět Vybrané partie teoretické informatiky v dalším semestru apod. Zvládnutí této problematiky se bude prověřovat (v podstatě jen) 8-bodovým příkladem v závěrečné zkoušce. Je zde pochopitelně také větší prostor pro diskusi než v první části.

### Konečné automaty

Připomněli jsme si pojem konečného automatu a ukázali souvislost s programy s „malou pamětí“. (Viz slidy z přednášky.)

### Konečný automat při vyhledávání vzorku v textu

Po prohlédnutí kódu programu (viz slidy), který realizuje neoptimální vyhledávání jsme zkonstruovali 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  $U_0, U_1, \dots, U_6$ . Při konstrukci se nám osvědčilo značení

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

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

$U_2 \dots |ab|aaba \dots 0, 2$  (přečetli jsme 2. nebo 0. pozici vzorku)

...

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

De facto jsme demonstrovali obecný algoritmus, který se nazývá “Knuth - Morris - Pratt algorithm” a o němž bude pojednáno v jednom z referátů na cvičení.

Připomněli jsme si 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 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.  $bn(0101) = 5$ ).

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

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

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

### Přednáška - druhá část

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.

Také jsme se věnovali stručnému matematickému popisu konstrukce, která ke dvěma automatům  $A_1, A_2$  zkonstruuje automat  $A$  tak, že  $L(A) = L(A_1) \cap L(A_2)$ , a důkazu korektnosti této konstrukce.

Uvědomili jsme si, že konstrukce konečných automatů úzce souvisí s operací kvocientu jazyka podle slova. Ilustrovali jsme definici

$$u \setminus L = \{v \mid uv \in L\},$$

uvedli značení (k danému  $A = (Q, \Sigma, \delta, q_0, F)$ )

$$L_q^{toAcc} = \{v \in \Sigma^* \mid q \xrightarrow{v} F\} \quad (\text{tedy } L(A) = L_{q_0}^{toAcc})$$

a intuitivně jsme pochopili, že

$$\text{když máme } q_0 \xrightarrow{w} q, \text{ tak platí } L_q^{toAcc} = w \setminus L_{q_0}^{toAcc} = w \setminus L(A).$$

Pak jsme 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  $\{w \setminus L \mid w \in \Sigma^*\}$  konečná.

### Partie textu k prostudování

Posluchači by si měli projít Kapitulu 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

Na začátku budou přiděleny **referáty**, jejichž předběžné zadání bude k dispozici na webstránce předmětu (a bude se tam i aktualizovat).

#### Příklad 1.1

Vypište prvních deset slov z jazyka  $L = \{w \in \{a, b\}^* \mid \text{každý výskyt podslova } aa \text{ je ve } w \text{ ihned následován znakem } b\}$ . 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

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 1.3

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 1.4**

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

(Nepovinně: Charakterizujte jazyky  $a \setminus L$ ,  $b \setminus L$ .)

**Příklad 1.5**

Zkonstruujte modulárně konečný automat přijímající jazyk

$L = \{w \in \{0, 1\}^* \mid w \text{ obsahuje podслово } 010 \text{ nebo } |w|_1 \text{ je sudé}\}$ .

Potom se zamyslete, zda půjde počet stavů výsledného automatu zmenšit.

**Příklad 1.6**

(Nepovinně.)

Zjistěte, zda obecně platí některý ze vztahů

$$(uv) \setminus L = u \setminus (v \setminus L),$$

$$(uv) \setminus L = v \setminus (u \setminus L).$$

**Příklad 1.7**

(Nepovinně.)

Dokažte matematickou indukcí, že žádné slovo  $u$  nesplňuje vztah

$$au = ub$$

pro dva různé symboly  $a, b$ . (Podle čeho povedete indukci ?)