Týden 3

Přednáška

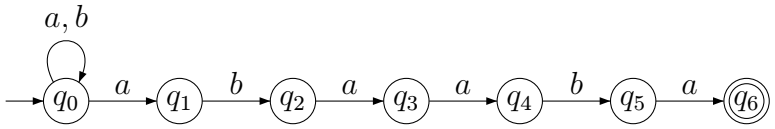
Na rozšiřující přednášce minulý týden jsme se věnovali zejména větě

"jazyk L je regulární $\Leftrightarrow$ množina levých kvocientů L podle všech slov je konečná"

a jejím důsledkům a aplikacím.
Na začátku této přednášky ještě připomeneme problém minimalizace konečného automatu a příslušné souvislosti, včetně důsledku pro zjišťování ekvivalence dvou konečných automatů (izomorfismus ekvivalentních minimálních automatů, normovaný tvar, ...).

Nedeterministické konečné automaty

Definici nedeterministického konečného automatu (NKA)
$A = (Q, \Sigma, \delta, I, F)$,
kde $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ ($\mathcal{P}(Q)$ značí množinu všech podmnožin množiny Q),
$I \subseteq Q$,
jsme si osvětlili na následujícím příkladu (kde $I = \{q_0\}$, $\delta(q_0, a) = \{q_0, q_1\}$, $\delta(q_2, b) = \emptyset$, atd.)



a uvědomili jsme si, že pro automat A na obrázku je $L(A) = \{w \mid \exists q \in I : q \xrightarrow{w} F\}$ roven jazyku, pro nějž jsme konstruovali (deterministický) konečný automat na první přednášce, totiž jazyku

$$L(A) = \{w \in \{a, b\}^* \mid w \text{ má sufix } abaaba\}.$$

(Predikát $q \xrightarrow{w} q'$ definujeme induktivně takto: 1/ $q \xrightarrow{\varepsilon} q$; 2/ jestliže $q' \in \delta(q, a)$ a $q' \xrightarrow{u} q''$, tak $q \xrightarrow{au} q''$. Dále $q \xrightarrow{w} Q'$ je zkratka za $\exists q' \in Q' : q \xrightarrow{w} q'$.)
Připomněli jsme si, že existuje algoritmus, který převede zadaný nedeterministický konečný automat na ekvivalentní deterministický; tento algoritmus je popsán ve studijním textu (formou „knoflíkové hry"); přitom se konstruují jen *dosažitelné stavy* v příslušném DKA (což může v konkrétních případech ušetřit mnoho stavů).
Formálně lze stručně zachytit konstrukci (i s nedosažitelnými stavy) takto:

> k NKA $A = (Q, \Sigma, \delta, I, F)$ uvažujme
> DKA $A' = (\mathcal{P}(Q), \Sigma, \delta', I, F')$, kde $F' = \{K \subseteq Q \mid K \cap F \neq \emptyset\}$ a (pro každé $K \subseteq Q, a \in \Sigma$ položíme) $\delta'(K, a) = \bigcup_{q \in K} \delta(q, a)$.

Indukcí (podle délky slov w) lze rutinně ověřit, že platí

pro každé $q \in Q$, $w \in \Sigma^*$ v DKA A' máme $\{q\} \xrightarrow{w} K$, kde $K = \{q' \mid q \xrightarrow{w} q' \text{ v NKA } A\}$.

Z toho snadno vyvodíme, že

$$\{w \in \Sigma^* \mid \exists q \in I : q \xrightarrow{w} F \text{ v NKA } A\} = \{w \in \Sigma^* \mid I \xrightarrow{w} F' \text{ v DKA } A'\},$$

a tedy že $L(A) = L(A')$.

Zobecněné nedeterministické konečné automaty (s ε-přechody)

Zatím neformálně jsme uvedli příklad regulárního výrazu, konkrétně $(aa)^*(bb)^*(cc)^*$ a shodli se, že ho chápeme jako reprezentaci jazyka $L = \{a^{2k}b^{2\ell}c^{2m} \mid k, \ell, m \geq 0\}$.
Pak jsme si uvědomili, jak elegantně lze z automatů pro jazyky $\{a^{2k} \mid k \geq 0\}$, $\{b^{2\ell} \mid \ell \geq 0\}$, $\{c^{2m} \mid m \geq 0\}$ vytvořit automat přijímající jazyk L, když použijeme *ε-přechody*.
Uvedli jsme takto zobecněný nedeterministický konečný automat (ZNKA), tedy strukturu $A = (Q, \Sigma, \delta, I, F)$, kde je patřičně rozšířen definiční obor přechodové funkce:

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)\,.$$

Přirozeně jsme došli k tomu, jak zde budeme chápat značení $q \xrightarrow{w} q'$ (či $q \xrightarrow{w} F$ apod.).

> Stručná induktivní definice může vypadat takto:
>
> 1. $q \xrightarrow{\varepsilon} q$,
> 2. když $\delta(q, a) \ni q'$ (pro $a \in \Sigma \cup \{\varepsilon\}$) a $q' \xrightarrow{v} q''$, tak $q \xrightarrow{av} q''$.

I pro ZNKA A lze tedy definovat $L(A) = \{w \in \Sigma^* \mid \exists q \in I : q \xrightarrow{w} F\}$.
Myšlenku důkazu věty 3.24. (s. 103)

> Existuje algoritmus, který ke každému zobecněnému nedeterministickému konečnému automatu A sestrojí ekvivalentní (deterministický) konečný automat A' (tedy $L(A) = L(A')$).

jsme si demonstrovali na výše sestrojeném automatu (a de facto jsme tak provedli Cvičení 3.64, s. 102).

Uzávěrové vlastnosti třídy regulárních jazyků

Uvědomili jsme si, že umíme snadno dokázat věty typu 3.26, 3.27, 3.28 (sekce 3.10), tedy

> Třída REG je uzavřena vůči sjednocení, průniku, doplňku. (Je-li tedy $L_1, L_2 \in$ REG, pak také $L_1 \cup L_2$, $L_1 \cap L_2$, $\overline{L_1}$ jsou v REG.)
>
> Třída REG je uzavřena vůči zřetězení a iteraci. (Je-li tedy $L_1, L_2 \in$ REG, pak také $L_1 \cdot L_2$, a $(L_1)^*$ jsou v REG.)
>
> Třída REG je uzavřena vůči operaci zrcadlového obrazu. (Je-li tedy $L \in$ REG, pak také $L^R \in$ REG.)

Také jsme si uvědomili konstruktivnost těchto tvrzení (tedy existenci příslušných algoritmů). Speciálně jsme si odvodili konstrukce zachycené na obrázku 3.15 na s. 114.

Regulární výrazy

Přesněji jsme uvedli, co myslíme pojmem *regulární výraz* (RV) v našem kontextu, a jakým způsobem reprezentuje výraz α určitý jazyk, označovaný $[\alpha]$. Uvedli jsme tedy Definici 3.30 a 3.31 (v sekci 3.12) a demonstrovali na příkladu (z Cvičení 3.80)

$$((01^*0 + 101)^*100 + (11)^*0)^*01\,,$$

na němž jsme rovněž ilustrovali standardní domluvu o prioritě operátorů, vynechávání znaku "$\cdot$", vynechávání nepotřebných závorek apod.
Uvědomili jsme si, že máme de facto k dispozici všechny prostředky pro návrh algoritmu z Věty 3.22

> Ke každému regulárnímu výrazu α lze sestrojit konečný automat přijímající jazyk $[\alpha]$.

za předpokladu, že umíme (algoritmicky) rozebrat syntaktickou strukturu zadaného výrazu. To znamená, že víme, jak sestavit příslušný syntaktický strom; např. pro výraz

$$(1^*0 + 10)^*1$$

může být takový strom zachycen lineárním zápisem

$$Conc(Iter(Union(Conc(Iter(1), 0), Conc(1, 0))), 1)\,.$$

Regulární a neregulární jazyky

Připomněli jsme si stručně intuici budovanou na minulém cvičení (otázky na s. 88-89). Uvědomili jsme si, že kombinace 'regulární' a 'neregulární' podmínky vždy vyžaduje hlubší zamyšlení. (Např. Otázky 3.51 a 3.52 demonstrují, že sjednocení $L_1 \cup L_2$ dvou jazyků, z nichž jeden je regulární a druhý neregulární, je v některých případech regulární a v jiných neregulární.)

Partie textu k prostudování

Jedná se zejména o části 3.7. (ekvivalence konečných automatů, minimální automaty), 3.8. (regulární a neregulární jazyky), 3.9. (nedeterministické konečné automaty), 3.10. (uzávěrové vlastnosti třídy regulárních jazyků), 3.12. (regulární výrazy).

(Máte si udělat přinejmenším dobrou první představu a zamyslet se nad příklady, speciálně těmi plánovanými na cvičení, ať se můžete na cvičení aktivně účastnit a případné problémy si tam objasnit.)

Cvičení

Příklad 3.1

	0	1
→ 1	1,2	1
2	3	-
3	-	4
④	-	-
→ 5	5	5,6
6	-	7
7	-	8
8	-	9
⑨	9	9

NKA (nedeterministický konečný automat) A zadaný uvedenou tabulkou zadejte jako (matematickou) strukturu $A = (Q, \Sigma, \delta, I, F)$ a pak zakreslete grafem. Charakterizujte co nejjednodušeji jazyk $L(A)$, tedy množinu $\{w \in \Sigma^* \mid \exists q \in I : q \xrightarrow{w} F\}$.

Pak zkonstruujte ekvivalentní DKA (deterministický konečný automat). Dokončete tedy konstrukci následující tabulky.

		0	1	
→	K_0	K_1	K_2	$K_0 = \{1, 5\}$
	K_1	K_3		$K_1 = \{1, 2, 5\}$
	K_2			$K_2 = \{1, 5, 6\}$
	K_3			$K_3 = \{1, 2, 3, 5\}$

(Pro procvičení můžete [samostatně později] výsledný DKA zredukovat a převést do normovaného tvaru.)

Příklad 3.2

Sestrojte (zobecněný) nedeterministický automat rozpoznávající jazyk zadaný regulárním výrazem $(aa)^*(bb)^*(cc)^*$. Pak k tomuto automatu zkonstruujte ekvivalentní deterministický automat.

Příklad 3.3

Připomeňme si, co jsou regulární výrazy (standardní v teorii). Zjistěte, zda platí
$\big[(011 + (10)^*1 + 0)^*\big] = \big[011(011 + (10)^*1 + 0)^*\big]$
$\big[((1 + 0)^*100(1 + 0)^*)^*\big] = \big[((1 + 0)100(1 + 0)^*100)^*\big]$
(Pro regulární výraz α zápisem $[\alpha]$ značíme jazyk reprezentovaný výrazem α.)

Příklad 3.4

Připomeňte si následující induktivní definici regulárních výrazů nad abecedou Σ.

- $\emptyset \in RV(\Sigma), \quad \epsilon \in RV(\Sigma), \quad a \in RV(\Sigma)$ (pro vš. $a \in \Sigma$);
- když $\alpha, \beta \in RV(\Sigma)$, pak také
  - $(\alpha + \beta) \in RV(\Sigma)$,
  - $(\alpha \cdot \beta) \in RV(\Sigma)$,
  - $(\alpha^*) \in RV(\Sigma)$.

Proveďte syntaktický rozbor výrazu $(01^*0+10)^*10$, v němž jsou uplatněna obvyklá pravidla o vynechávání závorek; tedy zkonstruujte příslušný syntaktický strom. Na jeho základě pak zkonstruujte ZNKA přijímající jazyk reprezentovaný uvedeným výrazem.

Příklad 3.5

Zadejte regulárním výrazem jazyk
$L = \{\, w \in \{0,1\}^* \mid$ ve w je sudý počet nul a každá jednička je bezprostředně následována nulou $\}$.

Příklad 3.6

(Nepovinně.)
Při konstrukci ekvivalence $\sim \subseteq Q \times Q$ definované vztahem $q \sim q' \Leftrightarrow L_q = L_{q'}$ (pro automat $A = (Q, \Sigma, \delta, q_0, F)$), jsme de facto postupovali *koinduktivně*: vzali jsme (největší) relaci $R_0 = Q \times Q$, na ni jsme aplikovali určitý monotónní funkcionál $\mathcal{F}$, dostali jsme tedy (menší) relaci $R_1 = \mathcal{F}(R_0)$, pak jsme dostali relaci $R_2 = \mathcal{F}(R_1)$, atd., až jsme dospěli k (největšímu) pevnému bodu R (pro nějž $R = \mathcal{F}(R)$). (Monotónností funkcionálu $\mathcal{F} : 2^{Q\times Q} \to 2^{Q\times Q}$ zde rozumíme vlastnost, že pro $T_1 \subseteq T_2$ máme $\mathcal{F}(T_1) \subseteq \mathcal{F}(T_2)$.) Snažte se funkcionál $\mathcal{F}$ přesně popsat.