4. přednáška LS 2006/2007

- Dokončíme rekapitulaci základních pojmů a metod týkajících se konečných automatů a regulárních jazyků započatou na minulé přednášce.
- Přidáme několik dalších tématek a příkladů k diskusi.
- Připomeneme si strukturu a obsah 1. zápočtové písemky.
- Naznačíme několik dalších věcí souvisejících s konečnými automaty.

(Jazyková) ekvivalence stavů a automatů

Nalezněme všechny dvojice stavů q, q', pro něž platí
$L_q^{out-to-F} = L_q'^{out-to-F}$.

	a	b
$\rightarrow$0	0	1
$\leftarrow$1	1	2
$\leftarrow$2	3	1
3	2	4
4	2	3

	a	b
$\rightarrow$5	5	6
6	7	5
$\leftarrow$7	7	9
8	9	8
$\leftarrow$9	8	7

Jaké známe metody (alespoň jednu) k zjištění, zda dva automaty A, A' jsou ekvivalentní, tedy $L(A) = L(A')$?

(Zobecněný) nedeterministický automat

	a	b
$\rightarrow$ 1	2	1
2	2	3
3	4	3
4	5	5
5	1	5

Zkonstruujte ZNKA rozpoznávající jazyk $L = \{ uv \mid uav \in L(A) \vee ubv \in L(A) \}$, kde A je KA zadaný uvedenou tabulkou. (Slova jazyka L vzniknou ze slov jazyka $L(A)$ vypadnutím jednoho písmene.)
Nakonec alespoň započněte konstrukci DKA pro jazyk L.

Konstrukce regulárního výrazu k automatu

Idea: zobecnění automatu; regulární výrazy na hranách

. . . . .

(viz např. současné slidy k ÚTI [na webu ing. M. Kota])

Věta. (Kleene) Regulárními výrazy lze reprezentovat právě regulární jazyky.

Induktivní definice regulárních jazyků.

Třída $RJ(\Sigma)$ *regulárních jazyků* nad abecedou Σ je nejmenší třída jazyků nad abecedou Σ, která obsahuje tzv. *elementární jazyky* a je uzavřena na *regulární operace*, tzn.:

- elementární jazyky, tj. $\emptyset$ a $\{a\}$ (pro každé $a \in \Sigma$), patří do $RJ(\Sigma)$,
- jestliže $L_1, L_2 \in RJ(\Sigma)$, pak také
  $L_1 \cup L_2 \in RJ(\Sigma)$,
- jestliže $L_1, L_2 \in RJ(\Sigma)$, pak také $L_1 \cdot L_2 \in RJ(\Sigma)$,
- jestliže $L \in RJ(\Sigma)$, pak také $L^* \in RJ(\Sigma)$.

Rozšířené regulární výrazy

Z uzávěrových vlastností třídy regulárních jazyků plyne:
regulární výrazy by mohly obohatit např. symboly pro průnik a doplněk, třeba $\&$, $\neg$, kde

- $[(\alpha\&\beta)] = [\alpha] \cap [\beta]$,
- $[\neg(\alpha)] = \Sigma^* - [\alpha]$ (abeceda Σ z kontextu)

Umožní se zkrácení výrazů:
např. $(0+1)^*1(0+1)^*$ lze nahradit výrazem $\neg(0^*)$

Ztrácí se ovšem "lineární" převod RV $\rightarrow$ ZNKA !

Substituce jazyků

Σ, Δ ... abecedy; pro $a \in \Sigma \ldots \sigma(a) \subseteq \Delta^*$

Rozšiřme *substituci* $\sigma : \Sigma \to \mathcal{P}(\Delta^*)$ na $\sigma : \Sigma^* \to \mathcal{P}(\Delta^*)$:

- $\sigma(\varepsilon) = \{\varepsilon\}$
- $\sigma(uv) = \sigma(u) \cdot \sigma(v) \quad (u, v \in \Sigma^*)$

Dále definujme pro $L \subseteq \Sigma^*$: $\sigma(L) = \bigcup_{w \in L} \sigma(w)$
(Říkáme: $\sigma(L)$ *vznikl* z L *substitucí* σ.)

Jestliže každé $\sigma(a)$ je regulární jazyk, jedná se o *regulární substituci*.

Jestliže každé $\sigma(a)$ obsahuje právě jedno slovo, σ se nazývá *homomorfismus* ($\sigma : \Sigma^* \to \Delta^*$).

Uzavřenost REG na regulární substituce

Tvrzení. Je-li $L \subseteq \Sigma^*$ regulární jazyk a $\sigma : \Sigma \rightarrow \mathcal{P}(\Delta^*)$ regulární substituce, pak $\sigma(L)$ je regulární jazyk.

Důkaz.
Nechť α a α_a (pro vš. $a \in abec(L)$) jsou regulární výrazy takové, že

- $[\alpha] = L$
- $[\alpha_a] = \sigma(a)$

Dosadíme-li do regulárního výrazu α za každý výskyt symbolu a regulární výraz α_a, dostaneme regulární výraz reprezentující $\sigma(L)$.

Ukázka 1. zápočtové písemky

Příklad 1 je zamýšlen asi na 15 minut (3 body). Bude prověřovat např. schopnost konstrukce jednoduchého DKA či NKA či regulárního výrazu, nebo převod NKA na DKA, minimalizaci DKA, apod.

---

1. Zkonstruujte co nejpřehledněji deterministický konečný automat rozpoznávající následující jazyk L.

$L = \{w \in \{0,1\}^* \mid$ číslo s binárním zápisem w je dělitelné pěti a w neobsahuje podřetězec $010\}$

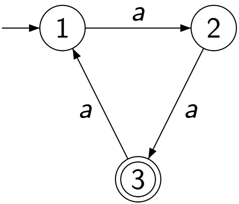
Každý z příkladů 2,3,4 je zamýšlen na 3-4 minuty a je hodnocen 1 bodem. Tyto příklady mají prověřit kratší formou něco z výše uvedeného a také např. znalost uzávěrových vlastností třídy regulárních jazyků, schopnost rozlišení regulárních jazyků od neregulárních, apod.

---

2. K následujícímu regulárnímu výrazu zkonstruujte ekvivalentní nedeterministický konečný automat.

$$(a + b)^* aab + ba(a + b)^*$$

3. Proč je následující automat minimalizovaný ? (Uveďte slova rozlišující jednotlivé stavy.)



4. Označte zřetelně všechny z následujících jazyků, které jsou regulární. (Připomínáme, že $|w|$ označuje délku slova w a $|w|_a$ označuje počet písmen a ve w. Dále w^R označuje zrcadlový obraz slova w.)

- $L_1 = \{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$
- $L_2 = \{w \in \{a, b\}^* \mid |w|_a \bmod 2 = 0\}$
- $L_3 = \{w \in \{a, b, c\}^* \mid w \text{ neobsahuje podřetězec } abc\}$
- $L_4 = \{w \in \{a, b\}^* \mid w \text{ začíná nebo končí dvojicí stejných písmen }\}$

Dvoucestné konečné automaty

Dvoucestné konečné automaty (kde $\delta : Q \times \Sigma \rightarrow Q \times \{-1, 0, 1\}$)

rozpoznávají také (jen) regulární jazyky (i v nedeterministické verzi).

Zobecněný sekvenční stroj

(GSM, generalized sequential machine)
$M = (Q, \Sigma, \Delta, \delta, \rho, q_0)$:
Q ... konečná množina *stavů*,
Σ ... konečná *vstupní abeceda*,
Δ ... konečná *výstupní abeceda*,
$\delta : Q \times \Sigma \to Q$ ... *přechodová funkce*,
$\rho : Q \times \Sigma \to \Delta^*$ ... *výstupní* funkce,
$q_0 \in Q$ ... *počáteční stav*.
Stroj M definuje zobrazení ("překlad") $f_M : \Sigma^* \to \Delta^*$:
1/ $f_M(\varepsilon) = \varepsilon$
2/ $f_M(wa) = f_M(w) \cdot \rho(\,\delta(q_0, w), a\,)$
(Pozn.: nedeterministická verze GSM je 'silnější'.)

Konečně stavový překladač (převaděč)

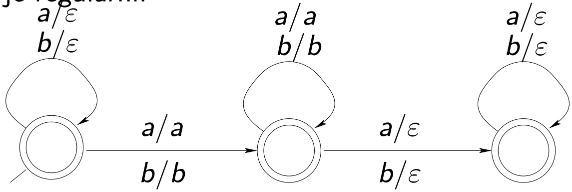
(finite [state] transducer)
$T = (Q, \Sigma, \Delta, \sigma, q_0, F)$:
Q ... konečná množina *stavů*,
Σ ... konečná *vstupní abeceda*,
Δ ... konečná *výstupní abeceda*,
σ ... konečná podmnožina množiny
$Q \times \Sigma^* \times Q \times \Delta^*$
(*"přechodová a výstupní" relace*),
$q_0 \in Q$ ... *počáteční stav*,
$F \subseteq Q$ ... množina *přijímajících stavů*.
Stroj T definuje relaci $R_T \subseteq \Sigma^* \times \Delta^*$:
$(u, v) \in R_T \quad \Leftrightarrow_{df}$ lze psát $u = u_1 u_2 \dots u_n$, $v = v_1 v_2 \dots v_n$ tak, že pro jisté stavy $q_1, q_2, \dots, q_n$ máme: $q_n \in F$ a $(q_{i-1}, u_i, q_i, v_i) \in \sigma$ pro $i = 1, 2, \dots, n$.

FT-relace (neboli racionální relace)

Tvrzení. Je-li R racionální relace, pak také R^{-1} je racionální.
Tvrzení. Je-li R racionální relace a L regulární jazyk, pak $R(L)$ je regulární.
(Podobně $R^{-1}(L)$ je regulární.)
$(R(L) = \{ v \mid \exists u \in L : (u,v) \in R \},$
$R^{-1}(L) = \{ u \mid \exists v \in L : (u,v) \in R \})$
Příklad: Je-li L regulární, pak

$$\mathit{infix}(L) = \{ v \mid \exists u, w : uvw \in L \}$$

je regulární:



Peterson's protocol

Process A:

** noncritical region **

$flag_A := true$

$turn := B$

waitfor

$(flag_B = false \vee turn = A)$

** critical region **

$flag_A := false$

** noncritical region **

Process B:

** noncritical region **

$flag_B := true$

$turn := A$

waitfor

$(flag_A = false \vee turn = B)$

** critical region **

$flag_B := false$

** noncritical region **

Kripkeho struktura (přechodový systém)