Týden 3

Přednáška-pondělí

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 $L_1, L_2 \in \mathsf{REG}$, 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 $L_1, L_2 \in \mathsf{REG}$, pak také $L_1 \cdot L_2$, a $(L_1)^*$ jsou v REG.)
>
> Třída REG je uzavřena vůči operaci zrcadlového obrazu. (Je-li tedy $L \in \mathsf{REG}$, pak také $L^R \in \mathsf{REG}$.)

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 "·", 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

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

Začali jsme řešení Cvičení 3.80, naznačili jsme tedy průběh převodu $RV \to ZNKA$ na výše uvedeném výrazu $((01^*0 + 101)^*100 + (11)^*0)^*01$.

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.)

Přednáška-čtvrtek

Uvědomili jsme si, že jsme de facto ukázali jeden směr věty 3.33.

> Regulárními výrazy lze reprezentovat právě regulární jazyky.

Zbývá tedy ukázat, že ke každému konečnému automatu A existuje (lze algoritmicky sestrojit) regulární výraz α_A tak, že jazyk $[\alpha_A]$ je roven jazyku $L(A)$.
Jeden způsob je zachycen příslušnou animací. My jsme si uvedli klasičtější postup indukcí:

Nechť $L = L(A)$ pro (deterministický) KA $A = (Q, \Sigma, \delta, 1, F)$, kde $Q = \{1, 2, \ldots, n\}$. Pro vš. $i, j \in \{1, 2, \ldots, n\}$ definujme

$$R_{ij} = \{w \in \Sigma^* \mid i \xrightarrow{w} j\}$$

(tj. jako množinu slov, které převedou A ze stavu i do stavu j).

Dokážeme-li, že každá množina R_{ij} $(i, j \in \{1, 2, \ldots, n\})$ je reprezentovatelná regulárním výrazem, jsme hotovi: stačí pak spojit znaménkem + regulární výrazy pro množiny R_{1j_1}, R_{1j_2}, …, R_{1j_m}, kde $F = \{j_1, j_2, \ldots, j_m\}$.

K úkolu vyjádřit množiny R_{ij} regulárními výrazy přistoupíme induktivně. Nehodí se zde ovšem indukce podle délky slov, použijeme indukci podle (maximálního čísla) povolených "mezistavů": každému $i \xrightarrow{w} j$ odpovídá jistá množina stavů (podmnožina množiny $\{1, 2, \ldots, n\}$), které se vyskytují uvnitř příslušného sledu; např. pro $5 \xrightarrow{a} 17 \xrightarrow{b} 4 \xrightarrow{a} 5 \xrightarrow{a} 2$ se tedy na sledu $5 \xrightarrow{abaa} 2$ vyskytují mezistavy 4,5,17.

Definujme tedy R^k_{ij} $(k = 0, 1, \ldots, n)$ jako množinu slov w, pro něž platí $i \xrightarrow{w} j$ a mezistavy na příslušném sledu tvoří podmnožinu množiny $\{1, 2, \ldots, k\}$ (tedy není tam žádný mezistav větší než k).

Ukážeme-li, že pro každé k je množina R^k_{ij} reprezentovatelná regulárním výrazem, budeme hotovi – je totiž $R_{ij} = R^n_{ij}$. Postupujeme indukcí podle k. Při $k = 0$ nejsou povoleny žádné mezistavy. Tedy $R^0_{ij} \subseteq \Sigma \cup \{\varepsilon\}$ $(a \in R^0_{ij} \Leftrightarrow \delta(i, a) = j$ a $\varepsilon \in R^0_{ij} \Leftrightarrow i = j)$. Indukční krok je zřejmý ze snadno odvoditelného vztahu

$$R^{k+1}_{ij} = R^k_{ij} \;\cup\; \left(\, R^k_{i,k+1}\, (R^k_{k+1,k+1})^*\, R^k_{k+1,j} \,\right).$$

Ukázali jsme si, na čem lze založit důkaz následujících dvou vět.

Věta. Je-li automat redukovaný a nemá nedosažitelné stavy, pak je minimální.

Věta. Dva minimální automaty, které přijímají tentýž jazyk (a jsou tedy ekvivalentní), jsou izomorfní, tedy stejné až na pojmenování stavů.

Uvažujme dva konečné automaty $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1\}$, $A_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2\}$ takové, že $L(A_1) = L(A_2) = L$; tedy $L^{toAcc}_{q_{01}} = L^{toAcc}_{q_{02}}$.

Připomeňme, že pro každé $w \in \Sigma^*$ existuje právě jeden stav $q_1 \in Q_1$ a právě jeden stav $q_2 \in Q_2$ tak, že $q_{01} \xrightarrow{w} q_1$, $q_{02} \xrightarrow{w} q_2$. (Přesněji psáno $q_{01} \xrightarrow{w}_{A_1} q_1$, $q_{02} \xrightarrow{w}_{A_2} q_2$.) Zřejmě přitom platí $w \backslash L = L^{toAcc}_{q_1} = L^{toAcc}_{q_2}$.

Ke každému dosažitelnému stavu $q_1 \in Q_1$ (v automatu A_1) tedy existuje dosažitelný stav $q_2 \in Q_2$ (v automatu A_2) tak, že $L^{toAcc}_{q_1} = L^{toAcc}_{q_2}$. Když je tedy A_1 bez nedosažitelných stavů a je redukovaný, což znamená, že $L^{toAcc}_{q_1} \neq L^{toAcc}_{q'_1}$ pro každé $q_1 \neq q'_1$ v Q_1, musí mít A_2 alespoň tolik stavů jako A_1 (tedy $|Q_1| \leq |Q_2|$). Platnost první věty je tedy jasná.

Když A_1, A_2 jsou minimální (a tedy i redukované a bez nedosažitelných stavů), tak zobrazení $f : Q_1 \to Q_2$ definované tak, že $L^{toAcc}_{q_1} = L^{toAcc}_{f(q_1)}$, je příslušným izomorfismem. (Ověřte detailně sami.)

Partie textu k prostudování

Kompletní část 3.9. (nedeterministické konečné automaty), 3.10. (uzávěrové vlastnosti třídy regulárních jazyků), 3.12. (regulární výrazy) a 3.8. (regulární a neregulární jazyky). (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

Prezentace referátů

Referát č. 3
Připomeňte, co je to homomorfismus a co je to izomorfismus mezi (relačními a algebraickými) strukturami (uveďte několik konkrétních příkladů). Pak vysvětlete, co to znamená, že dva konečné automaty jsou izomorfní.

Popište (rychlý) algoritmus testující, zda dané dva automaty jsou izomorfní.

(Dodatečná poznámka. Omezte se na automaty, u nichž jsou všechny stavy dosažitelné.)

Referát č. 4
Vysvětlete, jaký problém řeší tzv. Knuth-Morris-Prattův algoritmus (Knuth-Morris-Pratt algorithm) a ilustrujte jej např. na příkladu z přednášky z 1. týdne.

Příklad 3.1

Připomeňme si, co jsou regulární výrazy (standardní v teorii). Zjistěte, zda platí
$\big[(011 + (10)^*1 + 0)^*\big] = \big[011(011 + (10)^*1 + 0)^*\big]$
$\big[((1 + 0)^*100(1 + 0)^*)^*\big] = \big[((1 + 0)100(1 + 0)^*100)^*\big]$

Příklad 3.2

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

Příklad 3.3

Proveďte syntaktický rozbor výrazu (01*0 + 10)*10 (konstrukcí syntaktického stromu a odpovídajícího lineárního zápisu používajícího „procedury“ $Union, Conc, Iter$) a aplikujte algoritmus převodu RV → ZNKA.

K výslednému ZNKA alespoň započněte konstrukci ekvivalentního DKA.

==============================

Diskutujte případné problémy a nejasnosti týkající se zápočtové písemky.