Týden 8

Přednáška

Turingovy stroje, (výpočetní) problémy

Připomněli jsme si, co bylo cílem (třiadvacetiletého) Alana Turinga, když v roce 1936 zaváděl tzv. Turingův stroj, a nastínili jsme význam tzv. *Church-Turingovy teze*, která se stručně dá vyjádřit sloganem

$$\textit{algoritmus} = \textit{Turingův stroj}.$$

Sestrojili jsme konkrétní Turingův stroj M_1 s dvěma koncovými stavy q_{accept} a q_{reject}, který přijímá (nebezkontextový) jazyk

$$\{a^n b^n c^n \mid n \geq 1\}\,.$$

Snažili jsme se tedy přehledně zkonstruovat množinu instrukcí

$(q_0, a) \rightarrow (q_1, \bar{a}, +1)$
$(q_1, x) \rightarrow (q_1, x, +1)$ pro $x \in \{a, \bar{b}\}$
$(q_1, b) \rightarrow (q_2, \bar{b}, +1)$
...

Zároveň jsme připomněli definici Turingova stroje jako struktury $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$; speciálně jsme si uvědomili, že přechodová funkce

$$\delta : (Q - F) \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, +1\}$$

je, de facto, příslušnou množinou instrukcí ...
Připomněli jsme si, že na výpočet Turingova stroje M se dá pohlížet jako na posloupnost konfigurací $K_0 \vdash_M K_1 \vdash_M K_2 \vdash_M \cdots$.
(V našem konkrétním případě stroje M_1 např. platí $(q_0aaabbbccc) \vdash (\bar{a}q_1aabbbccc) \vdash (\bar{a}aq_1abbbccc) \vdash (\bar{a}aaq_1bbbccc) \vdash (\bar{a}aa\bar{b}q_2bbccc) \vdash \cdots \vdash (\bar{a}\bar{a}\bar{a}\bar{b}\bar{b}\bar{b}\bar{c}\bar{c}\bar{c}q_{accept}\Box)$; ale např. $(q_0aabbbccc) \vdash^* \bar{a}\bar{a}\bar{b}\bar{b}q_{reject}b\bar{c}\bar{c}c$).)
Uvědomili jsme si, že náš *stroj* (tedy algoritmus) *řeší problém*

NÁZEV: *Příslušnost k jazyku* $L = \{a^n b^n c^n \mid n \geq 1\}$

VSTUP: $w \in \{a, b, c\}^*$

VÝSTUP: ANO, když $w \in L$, NE jinak.

Jinými slovy, náš *stroj* (algoritmus) *rozhoduje (rozhodovací, neboli* ANO/NE*) problém*

NÁZEV: *Příslušnost k jazyku* $L = \{a^n b^n c^n \mid n \geq 1\}$

VSTUP: $w \in \{a, b, c\}^*$

OTÁZKA: Je $w \in L$?

Pak jsme si načrtli, jak by pracoval Turingův stroj M_2, který řeší následující problém (jenž není typu ANO/NE):

NÁZEV: *Zdvojení slova v abecedě* $\{a, b\}$.

VSTUP: $w \in \{a, b\}^*$.

VÝSTUP: ww.

Uvědomili jsme si, že každý Turingův stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ přirozeně definuje *částečné zobrazení*

$$f_M : \Sigma^* \to \Gamma^* .$$

Zobrazení je obecně částečné (tedy ne nutně totální) proto, že pro některé vstupy $w \in \Sigma^*$ se výpočet M nemusí zastavit a příslušný výstup tedy není definován; výrazem $!M(w)$ se zpravidla označuje fakt, že M se pro vstup w zastaví (a hodnota výstupu $f_M(w)$, někdy označovaná přímo jako $M(w)$, je definována).

Připomněli jsme si další konkrétní problémy (jako např. SAT [problém splnitelnosti booleovských formulí] a problém minimální kostry grafu) a přirozené způsoby kódování vstupů a výstupů slovy ve vhodné abecedě (která lze nakonec zakódovat „binárně“, tedy pomocí slov v abecedě $\{0, 1\}$).

Simulace mezi variantami Turingových strojů

Zavedli jsme přirozenou definici dvoupáskového Turingova stroje (2P-TS) jako struktury $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, kde

$$\delta : (Q - F) \times \Gamma \times \Gamma \to Q \times \Gamma \times \{-1, 0, +1\} \times \Gamma \times \{-1, 0, +1\} ,$$

a zkonstruovali jsme takový stroj, který řeší problém „Zdvojení slova“. Sestrojili jsme přitom následující množinu instrukcí.

$$
\begin{aligned}
&(q_0, x, \Box) \to (q_0, x, +1, x, +1) \text{ pro } x \in \{a, b\}\\
&(q_0, \Box, \Box) \to (q_1, \Box, -1, \Box, 0)\\
&(q_1, x, \Box) \to (q_1, x, -1, \Box, 0) \text{ pro } x \in \{a, b\}\\
&(q_1, \Box, \Box) \to (q_2, \Box, +1, \Box, 0)\\
&(q_2, x, \Box) \to (q_2, x, +1, x, +1) \text{ pro } x \in \{a, b\}\\
&(q_2, \Box, \Box) \to (q_{halt}, \Box, 0, \Box, 0)
\end{aligned}
$$

Pak jsme si ujasnili, jak lze obecný dvoupáskový Turingův stroj (2P-TS) M simulovat jednopáskovým dvouhlavým Turingovým strojem (1P-2H-TS) M'. Stroje M, M' mají tedy stejné *(vstupně/výstupní) chování*, tj. realizují tutéž (částečnou) funkci $f_M = f_{M'}$.

(Idea spočívá v použití „dvoustopé pásky“, tedy místo abecedy Γ stroje M má stroj M' abecedu $\Gamma' = \Gamma \cup (\Gamma \times \Gamma)$. První hlava mění jen „horní stopu“, druhá jen „dolní stopu“. Přitom se musí ošetřit případný zapisovací konflikt, když hlavy stojí na stejném políčku pásky.)

Navázali jsme náčrtem simulace obecného 1P-2H-TS M' standardním strojem, tedy jednopáskovým jednohlavým Turingovým strojem.

(Ten si na pásce musí označovat místa, kde by stály simulované hlavy, a „trochu se naběhá“.)

Sekce pro hlubší zájemce – důkazy

Vrátili jsme se k pumping lemmatu pro bezkontextové jazyky.
Věta. ($uvwxy$-teorém)
Nechť L je bezkontextový jazyk. Pak existují přirozená čísla p, q tž. každé slovo $z \in L$ delší než p lze psát ve tvaru $z = uvwxy$, přičemž platí $vx \neq \varepsilon$, $|vwx| \leq q$ a $uv^iwx^iy \in L$ pro vš. $i \geq 0$.
Vezmeme-li $n = \max\{p + 1, q\}$, lze přeformulovat takto:
Věta. Nechť L je bezkontextový jazyk. Pak existuje přirozené číslo n tž. každé slovo $z \in L$, $|z| \geq n$, lze psát ve tvaru $z = uvwxy$, přičemž platí $vx \neq \varepsilon$, $|vwx| \leq n$ a $uv^iwx^iy \in L$ pro vš. $i \geq 0$.
Precizace „obrázkového důkazu“.
Nechť $L = L(G)$ pro BG $G = (\Pi, \Sigma, S, P)$ v Chomského NF, tedy s pravidly typu $X \longrightarrow YZ$, $X \longrightarrow a$.
Všimněme si: když na větvi deriv. stromu pro $z \in L$ jsou dva výskyty téhož neterminálu, řekněme A, pak
$S \Rightarrow^* uAy \Rightarrow^* uvAxy \Rightarrow^* uvwxy = z$
pro nějaké $u, v, w, x, y \in \Sigma^*$, $vx \neq \varepsilon$.
Nechť $|\Pi| = k$. Všimněme si dále:
- na větvi délky alespoň $k + 1$ (tedy s alespoň $k + 2$ vrcholy) jsou jistě alespoň dva výskyty téhož neterminálu (poslední vrchol už může být označen terminálem);
- má-li derivační strom pro $z \in \Sigma^*$ všechny větve kratší než $k + 1$, pak nutně $|z| \leq 2^{k-1}$ (počet listů binárního stromu hloubky $k-1$; v poslední úrovni, při přepisu neterminálu na terminál, už k zdvojnásobení nedochází);
- v deriv. stromě pro $z \in L$, $|z| > 2^{k-1}$, určitě existují dva různé vrcholy v_1, v_2 na stejné větvi (v_1 blíž ke kořeni) označené stejným neterminálem, přičemž podstrom s kořenem v_1 má hloubku nejvýš $k + 1$ [podstrom s kořenem v_1 už nemá vlastní podstrom, u nějž by došlo k opakování neterminálu na jedné větvi] – tedy má nejvýše 2^k listů.
Můžeme proto vzít: $p = 2^{k-1}$, $q = 2^k$.

Partie textu k prostudování

Problémy a algoritmy (část 6.1.), Turingovy stroje (část 6.2.). Částečně část 6.4., simulace mezi výpočetními modely, Churchova-Turingova teze.
(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. 16 (Simulace mezi různými variantami Turingových strojů 1)
Důkladně promyslete, popište a vysvětlete následující příklad (s návodem).

Mějme standardní Turingův stroj (předpokládající oboustranně nekonečnou pásku) $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$. Sestrojme k němu Turingův stroj $M' = (Q', \Sigma, \Gamma', \delta', q_0', F')$, který předpokládá jen jednostranně (tj. pravostranně) nekonečnou pásku—tedy z nejlevější buňky (na níž stojí hlava na počátku) nemůže přejít doleva—a přitom simuluje stroj M.
Naznačíme možný způsob konstrukce:
$Q' = \{q_0', q_1\} \cup \{q_x \mid x \in \Sigma\} \cup \{q_U \mid q \in Q\} \cup \{q_D \mid q \in Q\}$
$\Gamma' = \Sigma \cup (\Gamma \times \Gamma) \cup \{\not\cent, \Box\}$
$F' = \{q_U \mid q \in F\} \cup \{q_D \mid q \in F\}$
$\delta'(q_0', x) = (q_x, \not\cent, +1)$ ... pro $x \in \Sigma$
$\delta'(q_x, y) = (q_y, (x, \Box), +1)$ ... pro $x, y \in \Sigma$
$\delta'(q_x, \Box) = (q_1, (x, \Box), -1)$ ... pro $x \in \Sigma$
$\delta'(q_1, z) = (q_1, z, -1)$ ... pro $z \neq \not\cent$
$\delta'(q_1, \not\cent) = ((q_0)_U, \not\cent, +1)$
Obrázkem si znázorněte pásku a (na malém příkladu) počáteční fázi práce stroje M' (pozn.: asi vás napadne pojem 'dvoustopá páska'); doplňte pak instrukce stroje M' (tedy dodefinujte zobrazení δ') tak, aby skutečně simuloval M. (Ještě kousek nápovědy: U v indexu u stavu znamená 'up', D znamená 'down').
Referát č. 17 (Simulace mezi různými variantami Turingových strojů 2)
Představte si Turingův stroj pracující na "čtverečkované rovině" (místo lineární pásky). Vstupní slovo je zapsáno na začátku v jednom řádku, čtecí hlava stojí na jeho začátku (ostatní buňky=čverečky obsahují prázdný znak). Obor hodnot přechodové funkce je nyní rozšířen tak, že možné pohyby hlavy jsou Left, Right, Up, Down.
Stručně a srozumitelně popište, jak je možné simulovat tento "rovinný" stroj klasickým "lineárním" strojem. (Nápověda. Musíte tedy popsat, jak bude mít lineární stroj uložen na pásce obsah oné roviny; stačí mít v každém okamžiku zachycen jen obdélník obsahující všechna políčka roviny, která simulovaný stroj dosud navštívil. Pak musíte popsat, jak bude simulující stroj provádět analogii konkrétních instrukcí simulovaného.)

Příklady

Příklad 8.1

Navrhněte (co nejjednodušší) dvoupáskový Turingův stroj (2P-TS) M řešící problém příslušnosti k jazyku (palindromů) $L = \{w \in \{a, b\}^* \mid w = w^R\}$.

Příklad 8.2

K 2P-TS M z předchozího příkladu sestrojte standardní (jednopáskový, jednohlavový) Turingův stroj M', který řeší tentýž problém. Přitom se snažte používat obecný postup (který k libovolnému 2P-TS M sestrojí ekvivalentní TS M').

Diskuse témat druhé zápočtové písemky

Připomenutí druhé zápočtové písemky, která se bude psát na další přednášce. Připomenutí a diskuse témat zřejmých z ukázkové písemky a řešení případných problémů, se kterými přijdou studenti.