Týden 7

Přednáška

Na rozšiřující přednášce minulý týden jsme se věnovali hlavně zásobníkovým automatům … převod mezi variantami přijímání prázdným zásobníkem a koncovým stavem, deterministické zásobníkové automaty, návrh ZA pro jazyk $\{w \in \{a,b\}^* \mid w = w^R\}$ (samozřejmě nedeterministického ZA, protože deterministický ZA přijímající tento jazyk neexistuje).

Nebezkontextové jazyky

Metodou obrázků derivačních stromů z části 5.3. jsme si ukázali, proč jazyk

$$L = \{a^n b^n c^n \mid n \geq 0\}$$

není bezkontextový. Porozuměli jsme tak pumping lemmatu (neboli $uvwxy$-teorému) pro bezkontextové jazyky:

> *Věta.* Nechť L je bezkontextový jazyk. Pak existuje přirozené číslo n tž. každé slovo $z \in L$, $|z| \geq n$, (tedy každé 'dlouhé' slovo z jazyka L) lze psát ve tvaru $z = uvwxy$, přičemž platí
>
> - $vx \neq \varepsilon$,
> - $|vwx| \leq n$,
> - pro vš. $i \geq 0$ je $uv^i wx^i y \in L$.

Intuici k poznání nebezkontextových jazyků jsme si posílili zvážením jazyků

$$L_1 = \{ww \mid w \in \{0,1\}^*\},$$
$$L_2 = \{\, 0^m 1^n 0^m \mid m = 2n \,\},$$

u nichž obou jsme usoudili, že ne všechna jejich dlouhá slova lze napsat v patřičném tvaru $uvwxy$, a že tedy nejsou bezkontextové. (U jazyka L_1 jde např. o slova tvaru $0^n 1^n 0^n 1^n$.)

Uzávěrové vlastnosti třídy CFL

Připomněli jsme si, že snadno umíme ukázat, že CFL je uzavřena vůči sjednocení.
Pak jsme si všimli, že jazyky $L_1 = \{a^i b^j c^k \mid i = j\}$ a $L_2 = \{a^i b^j c^k \mid j = k\}$ jsou bezkontextové, ale $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$ je jazyk, o němž víme, že bezkontextový není.
Tedy CFL není uzavřena na průnik. Díky de Morganovým pravidlům ihned vidíme, že CFL není uzavřena ani na doplněk.
(Jelikož $(L_1 \cap L_2 = (\overline{\overline{L_1} \cup \overline{L_2}})$, tak z uzavřenosti na doplněk by díky uzavřenosti na sjednocení vyplynula uzavřenost na průnik.)

Poznámka. Jak jsme si už řekli, třída DCFL, tj. třída jazyků rozpoznatelných deterministickými zásobníkovými automaty (přijímacími stavy), je vlastní podtřídou CFL. Dá se ukázat, že (pod)třída DCFL je uzavřena vůči doplňku; není ovšem uzavřena vůči průniku (jak dokazují výše uvedené jazyky $\{a^i b^j c^k \mid i = j\}$, $L_2 = \{a^i b^j c^k \mid j = k\}$), a tedy není uzavřena ani vůči sjednocení.

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

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

Naznačili jsme konstrukci konkrétního Turingova stroje M_1 s dvěma koncovými stavy q_{accept} a q_{reject}, který přijímá (nebezkontextový) jazyk

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

Jedná se vlastně o sestavení (jako vždy, pokud možno přehledného, okomentovaného) programu s jediným typem instrukcí; jeho začátek může vypadat následovně.

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

Zároveň jsme připomněli definici Turingova stroje jako struktury $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ (kde $\Sigma \subseteq \Gamma$ a $\Gamma - \Sigma$ vždy obsahuje speciální symbol $\Box$); speciálně jsme si uvědomili, že přechodová funkce

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

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 $(q_0 aaabbbccc) \vdash (\bar{a} q_1 aabbbccc) \vdash (\bar{a}\bar{a} q_1 abbbccc) \vdash (\bar{a}\bar{a}\bar{a} q_1 bbbccc) \vdash (\bar{a}\bar{a}\bar{a}\bar{b} q_2 bbccc) \vdash \cdots \vdash (\bar{a}\bar{a}\bar{a}\bar{b}\bar{b}\bar{b}\bar{c}\bar{c}\bar{c} q_{accept}\Box)$; ale např. $(q_0 aabbbccc) \vdash^* \bar{a}\bar{a}\bar{b}\bar{b} q_{reject} b\bar{c}\bar{c}c)$.)
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).

(Výpočetní) problémy, rozhodovací (ANO/NE) problémy, ...

Připomněli jsme si obecné definice a konkrétní problémy, jako např. SAT [problém splnitelnosti booleovských formulí], problém minimální kostry grafu (i v rozhodovací verzi).
Je samozřejmé, že po pochopení problému musí být každý schopen uvést konkrétní příklady instancí (vstupů) a příslušných výstupů ... .
Také jsme diskutovali přirozené způsoby kódování vstupů a výstupů slovy ve vhodné abecedě (která lze nakonec zakódovat „binárně“, tedy pomocí slov v abecedě $\{0, 1\}$).

Partie textu k prostudování

Uzávěrové vlastnosti CFL (část 5.2.). Nebezkontextové jazyky (část 5.3.).

Problémy a algoritmy (část 6.1.), Turingovy stroje (část 6.2.).

(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 7.1

Připomeňme, že jako základní jsme (u nedeterministických zásobníkových automatů) brali přijímání slova prázdným zásobníkem. Zásobníkový automat jsme definovali jako jistou strukturu $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$, kde jazykem přijímaným automatem M rozumíme jazyk

$$L(M) = \{\, w \in \Sigma^* \mid \exists q \in Q : (q_0, w, Z_0) \vdash_M^* (q, \varepsilon, \varepsilon) \,\}\,.$$

U verze přijímání přijímacím stavem definujeme zásobníkový automat jako $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde $F \subseteq Q$, a pak definujeme

$$L(M) = \{\, w \in \Sigma^* \mid \exists q \in F, \exists \alpha \in \Gamma^* : (q_0, w, Z_0) \vdash_M^* (q, \varepsilon, \alpha) \,\}\,.$$

Řekněme, že k zás. automatu $M_1 = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ přijímajícímu přijímacím stavem jazyk L chceme sestrojit zás. automat M_2 přijímající jazyk L prázdným zásobníkem. Stačí prostě upravit M_1 tak, že pro každý stav $q \in F$ dodáme instrukci $(q, \varepsilon, X) \to (q, \varepsilon)$ pro každé $X \in \Gamma$? Když ne, zkuste alespoň stručně navrhnout bezpečnou úpravu …

Příklad 7.2

Na přednášce jsme mj. ukázali, že třída CFL není uzavřena na doplněk. Existuje tedy jazyk $L \subseteq \Sigma^*$ (pro nějakou abecedu Σ), který je bezkontextový, přičemž jeho doplněk $\overline{L} = \Sigma^* - L$ bezkontextový není.
Víme, že jazyk $L_1 = \{ww \mid w \in \{a, b\}^*\}$ bezkontextový není. Zkuste prokázat, že $L_2 = \overline{L_1}$ bezkontextový je a představuje tak konkrétní příklad jazyka z CFL, jehož doplněk v CFL není.
Nápověda. Vystihněte co nejjednodušeji, jak vypadá slovo z L_2, které má sudou délku, tedy $2d$ pro nějaké $d \geq 1$. Pak se zkuste zamyslet, zda vám pomůže následující vztah:

$$2d = (d_1 + 1 + d_2) + (d_1 + 1 + d_2) = (d_1 + 1 + d_1) + (d_2 + 1 + d_2)\,.$$

Nakonec si tipněte, jestli se může podařit navrhnout *deterministický* zásobníkový automat přijímající L_2.
(Pomůže vám nějak u předchozí otázky, když se dozvíte, že třída DCFL je uzavřena vůči doplňku?)

Příklad 7.3

(Na základě alespoň intuitivních argumentů) určete, které z daných jazyků
jsou regulární:
jsou bezkontextové, ale ne regulární:
nejsou bezkontextové:

$L_1 = \{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$
$L_2 = \{w \in \{a,b\}^* \mid |w|_a$ je sudé $\}$
$L_3 = \{w \in \{a,b\}^* \mid w$ obsahuje podslovo $abba\}$
$L_4 = \{w \in \{a,b,c\}^* \mid |w|_a = |w|_b = |w|_c\}$
$L_5 = \{w \in \{a,b\}^* \mid |w|_a$ je prvočíslo$\}$
$L_6 = \{\, 0^m 1^n \mid m \leq 2n \,\}$
$L_7 = \{\, 0^m 1^n 0^m \mid m = 2n \,\}$

Příklad 7.4

Na přednášce jsme již příliš nediskutovali o (výpočetních) problémech. Zformulujte, co to je (v našem kontextu) pojem *problém* a pak speciálně rozhodovací (neboli ANO/NE) problém. Projděte mj. závěr podkladu k přednášce (mj. problém SAT [jako vždy, s ilustrativními instancemi], minimální kostra, ...).

Specifikujte také problém jednoznačnosti bezkontextových gramatik. O něm jsme si řekli, že je algoritmicky nerozhodnutelný. Zformulujte definici (algoritmicky) rozhodnutelného (obecně *algoritmicky řešitelného*) problému ...

Příklad 7.5

(Případně až příště.)

Zformalizujte situaci obchodního cestujícího, který má soupis měst k projetí a zná vzdálenost mezi každými dvěma městy, jakožto (výpočetní) problém; jde přitom o nalezení co nejkratší cesty, při níž jsou navštívena všechna města. Pak navrhněte rozhodovací verzi problému (při níž je dán limit). Vždy uveďte příklady konkrétních instancí a příslušných výstupů. (V případě rozhodovací verze uveďte alespoň jednu pozitivní a jednu negativní instanci.)