Deterministické zásobníkové automaty

Peter Kostolányi

25. apríla 2017

Definícia

Zásobníkové automaty môžu využívať dve formy nedeterminizmu. Jednak môže byť pre daný stav q, vstupný symbol (prípadne prázdne slovo) z a zásobníkový symbol Z definovaný viac ako jeden prechod – množina $\delta(q, z, Z)$ môže obsahovať viac ako jeden prvok. Alebo môže byť pre stav q a zásobníkový symbol Z definovaný prechod na písmeno c a súčasne prechod na ε – množiny $\delta(q, c, Z)$ a $\delta(q, \varepsilon, Z)$ môžu byť obidve neprázdne. Aj v tomto prípade ide o druh nedeterminizmu, pretože nie je jasné, či má výpočet pokračovať krokom na písmeno, alebo krokom na ε.

Deterministické zásobníkové automaty sú zásobníkové automaty zbavené týchto dvoch foriem nedeterminizmu. To znamená, že na každý vstupný symbol (alebo prázdne slovo) z môže z každého stavu a na každý zásobníkový symbol viesť najviac jeden prechod. Ak je navyše pre daný stav a zásobníkový symbol definovaný prechod na písmeno, nemôže byť pre túto dvojicu definovaný žiaden prechod na ε.

Definícia 1. *Deterministický zásobníkový automat* (DPDA) je (nedeterministický) zásobníkový automat $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že platí:

(*i*) Pre všetky $q \in K$, všetky $z \in \Sigma \cup \{\varepsilon\}$ a všetky $Z \in \Gamma$ obsahuje množina $\delta(q, z, Z)$ nanajvýš jeden prvok.

(*ii*) Ak je pre nejaké $q \in K$, $c \in \Sigma$ a $Z \in \Gamma$ množina $\delta(q, c, Z)$ neprázdna, tak je množina $\delta(q, \varepsilon, Z)$ prázdna.

Poznámka 1. Deterministické zásobníkové automaty by bolo možné definovať aj ako usporiadanú sedmicu $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde $\delta\colon K \times (\Sigma \cup \{\varepsilon\}) \cup \Gamma \nrightarrow K \times \Gamma^*$ je *čiastočná* funkcia spĺňajúca analógiu podmienky (*ii*).

Pri definícii 1 nie sú potrebné ďalšie definície konfigurácie, kroku výpočtu a akceptovaných jazykov $L(A)$ a $N(A)$ – tieto sa „zdedia“ od zásobníkových automatov. To možno chápať ako jej výhodu oproti alternatívnemu prístupu cez čiastočné prechodové funkcie. Nevýhodou je naopak „nutnosť“ pracovať s jednoprvkovými množinami namiesto ich prvkov. V nasledujúcom ale od tejto „nutnosti“ upustíme a namiesto $\delta(p, z, Z) = \{(q, \gamma)\}$ budeme písať $\delta(q, z, Z) = (q, \gamma)$, čo je odôvodnené očividnou korešpondenciou medzi oboma alternatívnymi definíciami.

Definícia 2. Jazyk L je *deterministický bezkontextový*, ak existuje deterministický zásobníkový automat A akceptujúci jazyk L *stavom* – pre automat A teda musí platiť $L(A) = L$. Triedu všetkých deterministických bezkontextových jazykov označujeme $\mathscr{L}_{detCF}$.

Poznámka 2. Definícia jazyka $L(A)$ je rovnaká ako pre (nedeterministické) zásobníkové automaty. Ak sa teda výpočet končí (konečnou alebo nekonečnou) postupnosťou prechodov na ε, na akceptáciu vstupného slova stačí *jediná akceptačná konfigurácia* v tejto časti výpočtu. Na akceptáciu vstupného slova je teda postačujúce, aby ho automat celé dočítal a následne sa dostal do akceptačného stavu. Nie je ale nutné, aby sa automat v tomto akceptačnom stave aj zastavil.

Uzavretosť triedy $\mathscr{L}_{detCF}$ na komplement

Dokážeme teraz, že trieda $\mathscr{L}_{detCF}$ *je uzavretá* na komplement. Dôsledkom je totiž nerovnosť tried $\mathscr{L}_{detCF}$ a $\mathscr{L}_{CF}$ – trieda $\mathscr{L}_{CF}$ na komplement uzavretá nie je. Keďže z definície deterministických zásobníkových automatov vyplýva $\mathscr{L}_{detCF} \subseteq \mathscr{L}_{CF}$, dôsledkom uzavretosti triedy $\mathscr{L}_{detCF}$ na komplement je vlastná inklúzia $\mathscr{L}_{detCF} \subsetneq \mathscr{L}_{CF}$.

Pri deterministických *konečných* automatoch stačí na akceptovanie komplementu vymeniť akceptačné a neakceptačné stavy pôvodného automatu. Pre deterministické *zásobníkové* automaty ale takáto priamočiara konštrukcia akceptáciu komplementu nezaručuje. Jedným z dôvodov je skutočnosť, že výpočet DPDA sa môže „zaseknúť“ pred dočítaním vstupného slova. V takom prípade je vstup zamietnutý nielen pôvodným automatom, ale aj automatom s vymenenými akceptačnými a neakceptačnými stavmi, ktorý tak nemôže akceptovať komplement pôvodného jazyka.

Deterministický zásobníkový automat sa môže „zaseknúť“ z dvoch príčin. Prvou je prázdnosť oboch množín $\delta(q, c, Z)$ a $\delta(q, \varepsilon, Z)$ pre nejaké $q \in K$, $c \in \Sigma$ a $Z \in \Gamma$. Druhou možnou príčinou je vyprázdnenie zásobníka pred dočítaním vstupného slova.

V nasledujúcom dokážeme, že ku každému deterministickému zásobníkovému automatu existuje ekvivalentný, ktorý sa nikdy „nezasekne“.

Definícia 3. Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je deterministický zásobníkový automat. Automat A je „nezasekávajúci sa“, ak platí:

(i) Pre všetky $q \in K$, $c \in \Sigma$ a $Z \in \Gamma$ je buď $\delta(q, c, Z) \neq \emptyset$, alebo $\delta(q, \varepsilon, Z) \neq \emptyset$.

(ii) Automat A v žiadnom výpočte nevyprázdni zásobník.

Lema 1. *Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je DPDA. Potom existuje „nezasekávajúci sa“ DPDA $A' = (K', \Sigma', \Gamma', \delta', q_0', Z_0', F')$ taký, že $L(A') = L(A)$.*

Dôkaz. Platnosť podmienky (i) zaručíme pridaním nového „odpadového“ stavu q_{odp} a dodefinovaním chýbajúcich prechodov tak, aby viedli do stavu q_{odp}, v ktorom automat A' dočíta vstup. Platnosť podmienky (ii) zaručíme pridaním nového zásobníkového symbolu D reprezentujúceho „falošné dno“. Po „náraze“ na symbol D automat prejde do stavu q_{odp}, v ktorom dočíta vstup.

Formálne: $K' = K \cup \{q_0', q_{odp}\}$ pre nové stavy q_0' a q_{odp}, $\Sigma' = \Sigma$, $\Gamma' = \Gamma \cup \{Z_0', D\}$ pre nové symboly Z_0' a D, $F' = F$ a prechodová funkcia δ' je daná nasledovne:

$$
\begin{aligned}
\delta'(q_0', \varepsilon, Z_0') &= (q_0, DZ_0), && \\
\delta'(q_0', \varepsilon, Z) &= (q_{odp}, Z) && \forall Z \in \Gamma \cup \{D\}, \\
\delta'(p, \varepsilon, Z_0') &= (q_{odp}, Z_0') && \forall p \in K \cup \{q_{odp}\}, \\
\delta'(p, z, Z) &= (q, \gamma) && \forall p, q \in K\ \forall z \in \Sigma \cup \{\varepsilon\}\ \forall Z \in \Gamma\ \forall \gamma \in \Gamma^* : \delta(q, z, Z) = (q, \gamma), \\
\delta'(p, c, Z) &= (q_{odp}, Z) && \forall p \in K\ \forall c \in \Sigma\ \forall Z \in \Gamma : \delta(p, c, Z) = \delta(p, \varepsilon, Z) = \emptyset, \\
\delta'(p, \varepsilon, D) &= (q_{odp}, D) && \forall p \in K, \\
\delta'(q_{odp}, c, Z) &= (q_{odp}, Z) && \forall c \in \Sigma\ \forall Z \in \Gamma \cup \{D\}.
\end{aligned}
$$

Prechod v prvom riadku slúži na vytvorenie „falošného dna“. Je zrejmé, že prechody v druhom a treťom riadku sa nepoužijú v žiadnom výpočte – slúžia iba na dodefinovanie chýbajúcich prechodov tak, aby výsledný automat vyhovoval definícii 3. □

Na akceptáciu komplementu ale nestačí ani výmena akceptačných a neakceptačných stavov „nezasekávajúceho sa“ automatu. Výpočet totiž môže skončiť aj v „epsilonovom cykle“ – počnúc nejakou konfiguráciou môže automat robiť iba kroky na ε, pričom zvyšok vstupného slova nikdy nedočíta. Problém je potom rovnaký, ako v prípade „zaseknutia“.

Ukážeme, že každý deterministický zásobníkový automat možno transformovať na ekvivalentný, ktorý je „nezasekávajúci sa“ a súčasne neobsahuje „epsilonové cykly“. „Epsilonové cykly“ pritom môžu byť dvoch druhov:

1. Buď zásobník počas „epsilonového cyklu“ rastie nad všetky medze.[1]

2. Alebo sa od určitého bodu „epsilonového cyklu“ začnú konfigurácie opakovať. Ak totiž zásobník nerastie nad všetky medze, môže obsahovať najviac K symbolov, kde K je konštanta. Keďže automat robí iba kroky na ε, neprečítaná časť vstupného slova sa počas „epsilonového cyklu“ nemení. Konfigurácií s obmedzenou výškou zásobníka a rovnakou neprečítanou časťou vstupného slova je však konečne veľa, preto sa po čase začnú opakovať.

[1] Tým samozrejme nie je myslené, že sa počet symbolov na zásobníku musí zvýšiť v každom kroku výpočtu. Hovoríme, že zásobník rastie nad všetky medze, ak sa pre všetky $n \in \mathbb{N}$ automat aspoň raz dostane do konfigurácie s aspoň n symbolmi na zásobníku.

Cieľom nasledujúcich úvah je zostrojiť automat, ktorý dokáže „epsilonové cykly“ detegovať, pričom po detekcii prejde do „odpadového stavu“, v ktorom dočíta vstup. Najprv ukážeme, že je možné detegovať „epsilonové cykly“ prvého typu.

Nech $(q, w, \gamma Z) \vdash^+ (q, w, \gamma\beta Z)$ je časť výpočtu DPDA A. Ak je počas celej tejto časti výpočtu výška zásobníka väčšia ako $|\gamma|$, tak tento výpočet od γ zjavne „nezávisí“. Ak teda navyše $|\beta| \geq 1$, nutne

$$(q, w, \gamma Z) \vdash^+ (q, w, \gamma\beta Z) \vdash^+ (q, w, \gamma\beta\beta Z) \vdash^+ \ldots,$$

pričom tento výpočet je „epsilonovým cyklom“ prvého typu.

Každý „epsilonový cyklus“ prvého typu zjavne musí pre všetky $N \in \mathbb{N}$ obsahovať podvýpočet, počas ktorého výška zásobníka „narastie“ aspoň o N symbolov. V nasledujúcom ukážeme, že ak zásobník počas výpočtu na ε „narastie“ aspoň o V symbolov, kde V je vhodná konštanta, tak tento výpočet nutne obsahuje podvýpočet typu $(q, w, \gamma Z) \vdash^+ (q, w, \gamma\beta Z)$, kde výška zásobníka je počas celého podvýpočtu väčšia ako $|\gamma|$. Na detekciu „epsilonového cyklu“ prvého typu tak stačí zistiť dostatočný „nárast“ zásobníka.

Nech k je maximálna dĺžka slova, ktoré možno v jednom kroku na ε pridať na zásobník:

$$k := \max\{|\gamma| \mid \gamma \in \Gamma^*;\ \exists p, q \in K\ \exists Z \in \Gamma : \delta(p, \varepsilon, Z) = (q, \gamma)\};$$

v jednom „epsilonovom“ kroku výpočtu teda zásobník „narastie“ najviac o $k-1$ zásobníkových symbolov. Položme $V := (k-1)\cdot(|K|\cdot|\Gamma|+1)$. Ak potom platí $|\beta| \geq |\alpha| + V$, tak výpočet

$$(p, w, \alpha) \vdash^+ (r, w, \beta)$$

musí obsahovať aspoň $|K|\cdot|\Gamma|+1$ konfigurácií takých, že vo všetkých nasledujúcich konfiguráciách uvedeného výpočtu je na zásobníku už len viac symbolov. Z Dirichletovho princípu ďalej vyplýva, že aspoň dve z týchto konfigurácií sa zhodujú v stave a v symbole na vrchu zásobníka. Výpočet $(p, w, \alpha) \vdash^+ (r, w, \beta)$ preto obsahuje podvýpočet

$$(q, w, \gamma Z) \vdash^+ (q, w, \gamma\eta Z),$$

počas ktorého je na zásobníku vždy viac ako $|\gamma|$ symbolov a kde $|\eta| > 0$. „Epsilonové cykly“ prvého typu teda skutočne možno detegovať na základe „nárastu“ výšky zásobníka aspoň o V.

Automat realizujúci túto detekciu „nárastu“ zásobníka aspoň o V si v stave pamätá aktuálnu hodnotu „nárastu“, ktorú v každom kroku výpočtu náležite upraví. V prípade, že by táto hodnota mala ísť do záporu, nastaví sa na nulu (stačí totiž, aby v nasledujúcom zásobník „narástol“ aspoň o V v porovnaní s touto aktuálnou konfiguráciou).

„Epsilonové cykly“ druhého typu možno detegovať na základe jednoduchého pozorovania, že v rámci každého takéhoto „ε-cyklu“ pre výšku zásobníka h platí $s \leq h < s + V$, kde s je vhodná konštanta. V opačnom prípade by totiž išlo o „epsilonový cyklus“ prvého typu. Konfigurácií s takto obmedzenou výškou zásobníka a s nezmenenou neprečítanou časťou vstupného slova je však iba konečne veľa – konkrétne $|K|\cdot|\Gamma|^V$. Položme $C := |K|\cdot|\Gamma|^V + 1$.

Automat realizujúci detekciu oboch typov „epsilonových cyklov“ si teda môže v stave pamätať „nárast“ zásobníka a počet krokov na ε. Ak „nárast“ dosiahne hodnotu V, ide o „epsilonový cyklus“ prvého typu. Ak počet krokov na ε dosiahne hodnotu C, ide o „epsilonový cyklus“ druhého typu. Ak by mal ísť „nárast“ zásobníka do záporu, nastaví sa na nulu nielen „nárast“ zásobníka, ale aj počítadlo krokov na ε (lebo v tomto prípade sa mení hodnota s). Pri kroku na písmeno sa vynulujú obidve počítadlá.

Lema 2. *Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je deterministický zásobníkový automat. Potom existuje deterministický zásobníkový automat $A' = (K', \Sigma', \Gamma', \delta', q_0', Z_0', F')$, pre ktorý platí $L(A') = L(A)$ a ktorý každý vstup dočíta do konca a zastane.*

Dôkaz. Bez ujmy na všeobecnosti predpokladajme, že deterministický zásobníkový automat A je „nezasekávajúci sa“.

Automat A' si bude v stave pamätať „nárast zásobníka na ε“ a „počet krokov na ε“. Navyše bude mať „odpadový“ stav, do ktorého prejde po detekcii „epsilonového cyklu“. Množina stavov K' teda bude daná ako

$$K' = (K \times \{0, \ldots, V-1\} \times \{0, \ldots, C-1\}) \cup \{q_{odp}\},$$

kde V a C sú konštanty zavedené vyššie. Vstupná aj pracovná abeceda ostanú nezmenené: $\Sigma' = \Sigma$ a $\Gamma' = \Gamma$. Prechodová funkcia δ' bude daná nasledovne:

1. Pre všetky $p, q \in K$, $c \in \Sigma$, $Z \in \Gamma$ a $\gamma \in \Gamma^*$ také, že $\delta(p, c, Z) = (q, \gamma)$ bude mať A' pre každé $s \in \{0, \ldots, V-1\}$ a $t \in \{0, \ldots, C-1\}$ definovaný prechod $\delta'([p, s, t], c, Z) = ([q, 0, 0], \gamma)$. Pri prechodoch na písmeno sa teda vynulujú obidve počítadlá.

2. Pre všetky $p, q \in K$ a $Z \in \Gamma$ také, že $\delta(p, \varepsilon, Z) = (q, \varepsilon)$ bude mať automat A' pre každé $t \in \{0, \ldots, C-1\}$ definovaný prechod $\delta'([p, 0, t], \varepsilon, Z) = ([q, 0, 0], \varepsilon)$. Ak by teda malo ísť počítadlo „nárastu“ zásobníka do záporu, obidve počítadla sa vynulujú.

3. Pre všetky $p, q \in K$, $Z \in \Gamma$ a $\gamma \in \Gamma^+$ také, že $\delta(p, \varepsilon, Z) = (q, \gamma)$ bude mať A' pre každé $s \in \{0, \ldots, V-1\}$, kde $s + |\gamma| - 1 \geq V$ a pre každé $t \in \{0, \ldots, V-1\}$ definovaný prechod $\delta'([p, s, t], \varepsilon, Z) = (q_{odp}, Z)$. Ak teda „nárast“ zásobníka dosiahne alebo presiahne hodnotu V, automat A' prejde do „odpadového“ stavu.

4. Pre všetky $p, q \in K$, $Z \in \Gamma$ a $\gamma \in \Gamma^*$ také, že $\delta(p, \varepsilon, Z) = (q, \gamma)$ bude mať A' pre každé $s \in \{0, \ldots, V-1\}$, kde $s + |\gamma| - 1 \geq 0$ definovaný prechod $\delta'([p, s, C-1], \varepsilon, Z) = (q_{odp}, Z)$. Ak teda počet krokov na ε dosiahne hodnotu C a „nárast“ zásobníka nejde do záporu, automat A' prejde do „odpadového“ stavu.

5. Pre všetky $p, q \in K$, $Z \in \Gamma$ a $\gamma \in \Gamma^*$ také, že $\delta(p, \varepsilon, Z) = (q, \gamma)$ bude mať A' pre každé $s \in \{0, \ldots, V-1\}$ a $t \in \{0, \ldots, C-1\}$ neošetrené v bodoch 2 až 4 definovaný prechod $\delta'([p, s, t], \varepsilon, Z) = ([q, s + |\gamma| - 1, t + 1], \gamma)$. K „nárastu“ zásobníka sa teda pripočíta hodnota $|\gamma| - 1$ a „počet krokov na ε“ sa zvýši o jedna.

6. Pre všetky $c \in \Sigma$ a $Z \in \Gamma$ bude mať automat A' definovaný prechod $\delta'(q_{odp}, c, Z) = (q_{odp}, Z)$. Automat A' teda môže v stave q_{odp} dočítať vstup.

Ďalej položme $q'_0 = [q_0, 0, 0]$, $Z'_0 = Z_0$ a $F' = F \times \{0, \ldots, V-1\} \times \{0, \ldots, C-1\}$. Tým je konštrukcia automatu A' dokončená. $\square$

Na zostrojenie automatu akceptujúceho komplement nie je postačujúca ani výmena akceptačných a neakceptačných stavov automatu, ktorý každý vstup dočíta do konca a zastane. Problémom sú výpočty, ktoré skončia postupnosťou niekoľkých krokov na ε. Ak je napríklad stav q akceptačný a stav p nie je akceptačný, je výpočet

$$(q_0, w, Z_0) \vdash^* (q, \varepsilon, \gamma) \vdash (p, \varepsilon, \gamma')$$

na slove w akceptačný. Ten istý výpočet je ale akceptačný aj pre automat s vymenenými akceptačnými a neakceptačnými stavmi, v ktorom je akceptačný stav p. Takýto automat tak nemôže akceptovať komplement pôvodného jazyka.

Riešenie tohto problému bude spočívať v zavedení špeciálnych stavov, ktoré sa budú môcť vyskytovať iba na konci postupností prechodov na ε. V nich si bude A' – automat akceptujúci komplement – pamätať informáciu o tom, či sa v predchádzajúcej postupnosti prechodov na ε vyskytol aspoň jeden akceptačný stav. Ak nie, bude takýto stav automatu A' akceptačný.

Veta 1. *Trieda $\mathscr{L}_{detCF}$ je uzavretá na komplement.*

Dôkaz. Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je deterministický zásobníkový automat, ktorý každý vstup prečíta až do konca a zastane. Popíšeme konštrukciu deterministického zásobníkového automatu $A' = (K', \Sigma, \Gamma, \delta', q'_0, Z_0, F')$ takého, že $L(A') = L(A)^C$.

Množina K' bude daná ako $K \times \{0, \ldots, 3\}$ – význam jednotlivých stavov je nasledovný:

- Stav $[q, 0]$ zodpovedá stavu q automatu A, pričom v aktuálnej postupnosti prechodov na ε sa zatiaľ nevyskytol žiaden akceptačný stav.
- Stav $[q, 1]$ zodpovedá stavu q automatu A, pričom v aktuálnej postupnosti prechodov na ε sa vyskytol aspoň jeden akceptačný stav.
- Stav $[q, 2]$ má rovnaký význam ako $[q, 0]$, ale automat A' v ňom môže byť iba na konci postupnosti prechodov na ε.
- Stav $[q, 3]$ má rovnaký význam ako $[q, 1]$, ale automat A' v ňom môže byť iba na konci postupnosti prechodov na ε.

Ak $q_0 \in F$, tak $q_0' = [q_0, 1]$ (v danej prázdnej postupnosti prechodov na ε sa už vyskytol akceptačný stav), v opačnom prípade $q_0' = [q_0, 0]$. Prechodová funkcia δ' bude daná nasledovne:

1. Pre všetky $p, q \in K$, $Z \in \Gamma$ a $\gamma \in \Gamma^*$ také, že $\delta(p, \varepsilon, Z) = (q, \gamma)$ bude $\delta'([p, 0], \varepsilon, Z) = ([q, 0], Z)$ ak $q \notin F$, $\delta'([p, 0], \varepsilon, Z) = ([q, 1], Z)$ ak $q \in F$ a v oboch prípadoch $\delta'([p, 1], \varepsilon, Z) = ([q, 1], Z)$. Druhý komponent sa teda môže zmeniť z 0 na 1 v prípade, že je stav q akceptačný.
2. Pre všetky $p \in K$ a $Z \in \Gamma$ s $\delta(p, \varepsilon, Z) = \emptyset$ bude $\delta'([p, 0], \varepsilon, Z) = ([p, 2], Z)$ a $\delta'([p, 1], \varepsilon, Z) = ([p, 3], Z)$. Na konci postupnosti prechodov na ε sa teda druhý komponent zmení z 0 na 2 a z 1 na 3.
3. Pre všetky $p, q \in K$, $c \in \Sigma$, $Z \in \Gamma$ a $\gamma \in \Gamma^*$ s $\delta(p, c, Z) = (q, \gamma)$ bude $\delta'([p, 2], c, Z) = ([q, k], \gamma)$ a $\delta'([p, 3], c, Z) = ([q, k], \gamma)$, kde $k = 1$ ak $q \in F$ a $k = 0$ ak $q \notin F$. Krok na písmeno sa teda môže vykonať iba zo stavu s druhým komponentom 2 alebo 3 a po jeho vykonaní sa druhý komponent nastaví na 0 resp. 1 podľa toho, či je nový stav automatu A akceptačný.

Za množinu akceptačných stavov automatu A' vezmeme $F' = F \times \{2\}$. Automat A' teda akceptuje, ak automat A dočíta vstupné slovo a v celej „záverečnej postupnosti krokov na ε“ sa nevyskytne žiaden akceptačný stav. □

Pozícia triedy $\mathscr{L}_{detCF}$ v Chomského hierarchii

Veta 2. *Platí $\mathscr{L}_{detCF} \subsetneq \mathscr{L}_{CF}$.*

Dôkaz.

- $\subseteq$: Deterministický zásobníkový automat je definovaný ako špeciálny zásobníkový automat. Preto každý jazyk v $\mathscr{L}_{detCF}$ je aj v $\mathscr{L}_{CF}$.
- $\not\supseteq$: Keby platilo $\mathscr{L}_{detCF} \supseteq \mathscr{L}_{CF}$, triedy $\mathscr{L}_{detCF}$ a $\mathscr{L}_{CF}$ by sa rovnali. Trieda $\mathscr{L}_{detCF}$ je však uzavretá na komplement, kým trieda $\mathscr{L}_{CF}$ nie je. Preto $\mathscr{L}_{detCF} \not\supseteq \mathscr{L}_{CF}$. □

Veta 3. *Platí $\mathscr{R} \subsetneq \mathscr{L}_{detCF}$*

Dôkaz.

- $\subseteq$: Aj keď deterministické konečné automaty z formálneho hľadiska nie sú deterministickými zásobníkovými automatmi, zjavne ich tak možno interpretovať. Presnejšie: ku každému DKA $A = (K, \Sigma, \delta, q_0, F)$ možno zostrojiť (zjavne ekvivalentný) DPDA $A' = (K, \Sigma, \Gamma, \delta', q_0, Z_0, F)$, kde $\Gamma = \{Z_0\}$ a pre všetky $p, q \in K$ a $c \in \Sigma$ je $\delta'(p, c, Z_0) = (q, Z_0)$ práve vtedy, keď $\delta(p, c) = q$.
- $\not\supseteq$: Nech $L \in \{a^n b^n \mid n \in \mathbb{N}\}$. Ako je známe, $L \notin \mathscr{R}$. Platí ale $L \in \mathscr{L}_{detCF}$, keďže L je akceptovaný deterministickým zásobníkovým automatom $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ daným nasledovne: $K = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, Z_0\}$, $F = \{q_0, q_3\}$ a

$$\begin{aligned} \delta(q_0, a, Z_0) &= (q_1, Z_0 a), & \delta(q_1, a, a) &= (q_1, aa), \\ \delta(q_1, b, a) &= (q_2, \varepsilon), & \delta(q_2, b, a) &= (q_2, \varepsilon), \\ \delta(q_2, \varepsilon, Z_0) &= (q_3, \varepsilon). \end{aligned}$$

□

Tvrdenie $\mathscr{L}_{detCF} \subsetneq \mathscr{L}_{CF}$ (veta 2) sme dokázali poukázaním na rozdielne uzáverové vlastnosti týchto dvoch tried. V nasledujúcom dokážeme pre dva *konkrétne* jazyky ich príslušnosť do rozdielu $\mathscr{L}_{CF} - \mathscr{L}_{detCF}$. Kým v dôkaze prvého z týchto tvrdení opäť využijeme uzavretosť triedy $\mathscr{L}_{detCF}$ na komplement, druhé tvrdenie bude možné interpretovať ako alternatívny dôkaz vety 2.

Tvrdenie 1. *Jazyk* $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}^C$ *je v* $\mathscr{L}_{CF} - \mathscr{L}_{detCF}$.

Dôkaz. Ako bolo dokázané na cvičeniach v zimnom semestri, $L \in \mathscr{L}_{CF}$. Keby platilo $L \in \mathscr{L}_{detCF}$, dôsledkom uzavretosti triedy $\mathscr{L}_{detCF}$ na komplement by bolo $L^C \in \mathscr{L}_{detCF} \subseteq \mathscr{L}_{CF}$. Pritom ale $L^C = \{a^n b^n c^n \mid n \in \mathbb{N}\} \notin \mathscr{L}_{CF}$, čo je spor. □

Tvrdenie 2. *Jazyk* $L = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2n} \mid n \in \mathbb{N}\}$ *je v* $\mathscr{L}_{CF} - \mathscr{L}_{detCF}$.

Dôkaz. Dôkaz tvrdenia $L \in \mathscr{L}_{CF}$ prenechávame čitateľovi ako jednoduché cvičenie. V nasledujúcom dokážeme, že $L \notin \mathscr{L}_{detCF}$.

Za účelom sporu predpokladajme, že $L \in \mathscr{L}_{detCF}$. Potom existuje deterministický zásobníkový automat A taký, že $L(A) = L$. Nech A' je *nedeterministický* zásobníkový automat, ktorý je ako A, ale v akceptačných stavoch má možnosť „prejsť do svojej kópie“, v ktorej sú všetky prechody na b zmenené na prechody na c (konštrukciu automatu A' prenechávame čitateľovi). Je zrejmé, že takýto automat akceptuje jazyk $L' = L \cup \{a^n c^n \mid n \in \mathbb{N}\} \cup \{a^n c^{2n} \mid n \in \mathbb{N}\} \cup \{a^n b^n c^n \mid n \in \mathbb{N}\}$. Pomocou pumpovacej lemy je však ľahko možné dokázať, že $L' \notin \mathscr{L}_{CF}$, čo je spor. □

Základné uzáverové vlastnosti triedy $\mathscr{L}_{detCF}$

Ako bolo dokázané vyššie, trieda $\mathscr{L}_{detCF}$ *je* uzavretá na komplement. V nasledujúcom preskúmame ďalšie uzáverové vlastnosti triedy $\mathscr{L}_{detCF}$.

Veta 4. *Trieda* $\mathscr{L}_{detCF}$ *nie je uzavretá na zjednotenie.*

Dôkaz. Jazyky $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ a $L_2 = \{a^n b^{2n} \mid n \in \mathbb{N}\}$ sú obidva deterministické bezkontextové – deterministický zásobníkový automat akceptujúci jazyk L_1 sme skonštruovali v rámci dôkazu vety 3 a rovnaký automat možno ľahko upraviť, aby akceptoval jazyk L_2. Jazyk $L_1 \cup L_2$ však nie je deterministický bezkontextový (tvrdenie 2). □

Platnosť predchádzajúceho tvrdenia je možné intuitívne zdôvodniť poukázaním na skutočnosť, že deterministický zásobníkový automat (na rozdiel od nedeterministického) nemá v úvode výpočtu možnosť rozhodnúť sa, ktorý z dvoch automatov pre jednotlivé jazyky simulovať. Zhruba povedané, v nasledujúcom dokážeme, že ide o jedinú prekážku na akceptovanie zjednotenia. Dokážeme totiž, že trieda $\mathscr{L}_{detCF}$ je uzavretá na tzv. *označené zjednotenie*, ktoré pre jazyky $L_1, L_2 \subseteq \Sigma^*$ a *nový* symbol $a \notin \Sigma$ možno definovať ako $aL_1 \cup L_2$ – prítomnosť resp. neprítomnosť symbolu a na začiatku slova teda určuje automat, ktorý je potrebné simulovať.

Veta 5. *Trieda* $\mathscr{L}_{detCF}$ *je uzavretá na označené zjednotenie.*

Dôkaz. Dokážeme, že ak $L_1 \subseteq \Sigma_1^*$ a $L_2 \subseteq \Sigma_2^*$ sú deterministické bezkontextové jazyky, je deterministický bezkontextový aj jazyk $aL_1 \cup L_2$, kde $a \notin \Sigma_1 \cup \Sigma_2$.

Nech $A_1 = (K_1, \Sigma_1, \Gamma_1, \delta_1, q_{0,1}, Z_{0,1}, F_1)$ je deterministický zásobníkový automat, pre ktorý platí $L(A_1) = L_1$ a $A_2 = (K_2, \Sigma_1, \Gamma_2, \delta_2, q_{0,2}, Z_{0,2}, F_2)$ je deterministický zásobníkový automat taký, že $L(A_2) = L_2$. Predpokladajme navyše, že automat A_2 je v normálnom tvare, v ktorom sa stav $q_{0,2}$ môže vyskytnúť iba na začiatku výpočtu. Automat $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ akceptujúci jazyk $aL_1 \cup L_2$ zostrojíme takto: $K = K_1 \cup K_2$, $\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{a\}$, $\Gamma = \Gamma_1 \cup \Gamma_2$, $q_0 = q_{0,2}$, $Z_0 = Z_{0,2}$, $F = F_1 \cup F_2$. Prechodová funkcia δ je pritom daná nasledovne:

$$
\begin{aligned}
\delta(q_{0,2}, a, Z_{0,2}) &= (q_{0,1}, Z_{0,1}), \\
\delta(q, z, Z) &= \delta_1(q, z, Z), \qquad \forall q \in K_1\ \forall z \in \Sigma \cup \{\varepsilon\}\ \forall Z \in \Gamma_1, \\
\delta(q, z, Z) &= \delta_2(q, z, Z), \qquad \forall q \in K_2\ \forall z \in \Sigma \cup \{\varepsilon\}\ \forall Z \in \Gamma_2.
\end{aligned}
$$

□

Poznámka 3. Občas je výhodnejšie pracovať s mierne odlišnou definíciou označeného zjednotenia ako jazyka $aL_1 \cup bL_2$, kde a, b sú rôzne symboly, ktoré nemusia byť nové. Drobnou úpravou dôkazu vety 5 možno ľahko odvodiť uzavretosť triedy $\mathscr{L}_{detCF}$ aj na takto definovanú operáciu.

Veta 6. *Trieda $\mathscr{L}_{detCF}$ nie je uzavretá na prienik.*

Dôkaz. Trieda $\mathscr{L}_{detCF}$ je uzavretá na komplement a podľa De Morganových zákonov platí identita $L_1 \cup L_2 = (L_1^C \cap L_2^C)^C$. Ak by teda bola trieda $\mathscr{L}_{detCF}$ navyše uzavretá na prienik, bola by uzavretá aj na zjednotenie, čo je spor s vetou 4. □

Poznámka 4. Predchádzajúcu vetu možno alternatívne dokázať aj pomocou jednoduchého protipríkladu. Ľahko totiž vidieť, že jazyky $L_1 = \{a^n b^m c^m \mid n, m \in \mathbb{N}\}$ a $L_2 = \{a^n b^n c^m \mid n, m \in \mathbb{N}\}$ sú obidva deterministické bezkontextové. Ich prienik $L_1 \cap L_2 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ však nie je ani bezkontextový. Takýmto spôsobom teda dostávame dokonca o niečo silnejšie tvrdenie, než v znení predchádzajúcej vety.

Veta 7. *Trieda $\mathscr{L}_{detCF}$ je uzavretá na prienik s regulárnym jazykom.*

Dôkaz. Rovnako ako pre bezkontextové jazyky. □

Veta 8. *Trieda $\mathscr{L}_{detCF}$ nie je uzavretá na zreťazenie.*

Dôkaz. Nech L_1, L_2 sú deterministické bezkontextové jazyky také, že ich zjednotenie $L_1 \cup L_2$ nie je deterministické bezkontextové. Ak a je nový symbol, jazyk $L_3 := aL_1 \cup L_2$ je deterministický bezkontextový. Jazyk $\{a\}^*$ je regulárny, a teda tiež deterministický bezkontextový. Dokážeme, že zreťazenie týchto jazykov – jazyk $\{a\}^* \cdot L_3$ – nie je v $\mathscr{L}_{detCF}$.

Za účelom sporu predpokladajme, že $\{a\}^* \cdot L_3 \in \mathscr{L}_{detCF}$. Z uzavretosti triedy $\mathscr{L}_{detCF}$ na prienik s regulárnym jazykom dostávame, že aj jazyk

$$L_4 := \{a\}^* \cdot L_3 \cap a(\Sigma_{L_1} \cup \Sigma_{L_2})^* = aL_1 \cup aL_2$$

je v $\mathscr{L}_{detCF}$. Potom existuje DPDA $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(A) = L_4$. Bez ujmy na všeobecnosti možno predpokladať, že $\delta(q_0, a, Z_0) \neq \emptyset$ a $\delta(q_0, c, Z_0) = \emptyset$ pre všetky $c \neq a$.[2] Potom možno zostrojiť DPDA $A' = (K', \Sigma, \Gamma, \delta', q_0', Z_0, F)$, kde $K' = K \cup \{q_0'\}$, q_0' je nový stav, $\delta'(q_0', \varepsilon, Z_0) = \delta(q_0, a, Z_0)$ a $\delta'(q, z, Z) = \delta(q, z, Z)$ pre všetky $q \in K$, $z \in \Sigma \cup \{\varepsilon\}$ a $Z \in \Gamma$. Očividne $L(A') = L_1 \cup L_2$, čo je spor s predpokladom, že jazyk $L_1 \cup L_2$ nie je deterministický bezkontextový. □

Veta 9. *Trieda $\mathscr{L}_{detCF}$ nie je uzavretá na iteráciu.*

Dôkaz. Nech $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ a $L_2 = \{a^n b^{2n} \mid n \in \mathbb{N}\}$. Obidva jazyky sú deterministické bezkontextové, ale jazyk $L_1 \cup L_2$ deterministický bezkontextový nie je.

Je jednoduchým cvičením dokázať, že pre všetky $L \in \mathscr{L}_{detCF}$ platí aj $L \cup \{\varepsilon\} \in \mathscr{L}_{detCF}$. Z uzavretosti na označené zjednotenie vyplýva, že je deterministický bezkontextový aj jazyk

$$L_3 := \{c\} \cup cL_1 \cup L_2 = c(L_1 \cup \{\varepsilon\}) \cup L_2,$$

kde c je nový symbol. Ukážeme, že jazyk L_3^* nie je deterministický bezkontextový.

Za účelom sporu predpokladajme, že jazyk L_3^* je deterministický bezkontextový. Z uzavretosti triedy $\mathscr{L}_{detCF}$ na prienik s regulárnym jazykom potom vyplýva, že aj jazyk

$$L_4 := L_3^* \cap ca^+b^+ = (cL_1 \cup cL_2) - \{c\}$$

je deterministický bezkontextový. Je triviálnym cvičením dokázať, že potom je deterministický bezkontextový aj jazyk $cL_1 \cup cL_2$. Rovnako ako v dôkaze vety 8 by následne bolo možné uzavrieť, že je deterministický bezkontextový aj jazyk $L_1 \cup L_2$, čo je spor s voľbou jazykov L_1 a L_2. □

[2] Ľahko vidieť, že výpočet automatu A môže začínať nanajvýš konečne dlhou postupnosťou prechodov na ε. Prvý prechod na písmeno možno pri troche ostražitosti „presunúť" hneď na začiatok výpočtu. Ďalej možno očividne ľahko docieliť, aby sa konfigurácia so stavom q_0 a symbolom Z_0 na vrchu zásobníka mohla vyskytnúť iba na začiatku výpočtu. V takom prípade sú prechody zo stavu q_0 na $c \neq a$ a Z_0 zbytočné a možno ich odstrániť.

Veta 10. *Trieda $\mathscr{L}_{detCF}$ nie je uzavretá na kladnú iteráciu.*

Dôkaz. Podobne ako pre iteráciu. $\square$

Veta 11. *Trieda $\mathscr{L}_{detCF}$ nie je uzavretá na reverz.*

Dôkaz. Nech $L_1 = \{b^n a^n \mid n \in \mathbb{N}\}$ a $L_2 = \{b^{2n} a^n \mid n \in \mathbb{N}\}$. Obidva jazyky sú zjavne deterministické bezkontextové. Nech $L_3 := L_1 \cup dL_2$ je označené zjednotenie jazykov L_1 a L_2, ktoré je tiež deterministické bezkontextové. Dokážeme, že jazyk

$$L_3^R = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2n} d \mid n \in \mathbb{N}\}$$

nie je deterministický bezkontextový. V zásade stačí zreplikovať dôkaz tvrdenia 2. Nech teda A je DPDA akceptujúci jazyk L_3^R. Nech A' je *nedeterministický* zásobníkový automat, ktorý je ako A, ale v akceptačných stavoch má navyše možnosť začať čítať namiesto symbolov b symboly c. Takýto automat zjavne akceptuje jazyk

$$L_4 := L_3^R \cup \{a^n c^n \mid n \in \mathbb{N}\} \cup \{a^n c^{2n} d \mid n \in \mathbb{N}\} \cup \{a^n b^n c^n d \mid n \in \mathbb{N}\},$$

o ktorom možno pomocou pumpovacej lemy dokázať, že nie je bezkontextový – spor. $\square$

Veta 12. *Trieda $\mathscr{L}_{detCF}$ nie je uzavretá na homomorfizmus.*

Dôkaz. Nech $L_1, L_2 \in \mathscr{L}_{detCF}$ sú jazyky také, že $L_1 \cup L_2 \notin \mathscr{L}_{detCF}$. Nech a je nový symbol. Jazyk $L_3 := aL_1 \cup L_2$ je deterministický bezkontextový.

Uvažujme teraz homomorfizmus $h\colon (\Sigma_{L_1} \cup \Sigma_{L_2} \cup \{a\})^* \to (\Sigma_{L_1} \cup \Sigma_{L_2})^*$ taký, že $h(a) = \varepsilon$ a pre všetky $c \in \Sigma_{L_1} \cup \Sigma_{L_2}$ je $h(c) = c$. Potom ale $h(L_3) = L_1 \cup L_2$, čo nie je deterministický bezkontextový jazyk. $\square$

Poznámka 5. Podobne možno dokázať, že trieda $\mathscr{L}_{detCF}$ nie je uzavretá ani na nevymazávajúci homomorfizmus. Nech $L_1, L_2 \in \mathscr{L}_{detCF}$ sú jazyky také, že $L_1 \cup L_2 \notin \mathscr{L}_{detCF}$ a nech a, b sú nové symboly. Jazyk $aL_1 \cup bL_2$ je deterministický bezkontextový. Pomocou nevymazávajúceho homomorfizmu ho však očividne možno „prerobiť" na jazyk $aL_1 \cup aL_2$. Ak by ale tento jazyk bol deterministický bezkontextový, bol by deterministický bezkontextový aj jazyk $L_1 \cup L_2$ (argumentácia je rovnaká ako v dôkaze vety 8).

Veta 13. *Trieda $\mathscr{L}_{detCF}$ je uzavretá na inverzný homomorfizmus.*

Dôkaz. Rovnako ako pre bezkontextové jazyky. $\square$

Akceptácia prázdnym zásobníkom

Ako bolo dokázané na jednej z prednášok v zimnom semestri, (nedeterministické) zásobníkové automaty akceptujúce stavom sú ekvivalentné (nedeterministickým) zásobníkovým automatom akceptujúcim prázdnym zásobníkom. V nasledujúcom ukážeme, že pre *deterministické* zásobníkové automaty tieto dva módy akceptácie ekvivalentné nie sú, pričom akceptácia stavom je ostro silnejšia. Táto skutočnosť je priamym dôsledkom *bezprefixovosti* jazykov akceptovaných deterministickými zásobníkovými automatmi prázdnym zásobníkom.

Definícia 4. Jazyk $L \subseteq \Sigma^*$ je *bezprefixový*, ak pre všetky $u \in \Sigma^*$ a $v \in \Sigma^+$ platí: ak $u \in L$, tak $uv \notin L$.

Veta 14. *Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je deterministický zásobníkový automat. Potom je jazyk $N(A)$ bezprefixový.*

Dôkaz. Nech $u \in N(A)$ a $v \in \Sigma^+$. Potom pre nejaké $q \in K$ musí platiť $(q_0, u, Z_0) \vdash^* (q, \varepsilon, \varepsilon)$ a v dôsledku toho aj $(q_0, uv, Z_0) \vdash^* (q, v, \varepsilon)$. Automat A deterministický – výpočet automatu A na slove uv sa teda *musí* dostať do konfigurácie (q, v, ε). V tejto konfigurácii je zásobník prázdny a automat A nemôže urobiť žiaden ďalší krok výpočtu. Z neprázdnosti slova v potom vyplýva, že automat A nikdy nedočíta slovo uv; preto $uv \notin N(A)$. $\square$

Môžeme teraz pristúpiť k samotnému porovnaniu oboch módov akceptácie. Najprv dokážeme, že akceptácia stavom nie je slabšia, než akceptácia prázdnym zásobníkom. Ak teda pre nejaký jazyk L existuje DPDA A taký, že $L = N(A)$, tak existuje aj DPDA A' taký, že $L = L(A')$.

Veta 15. *Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je deterministický zásobníkový automat. Potom existuje deterministický zásobníkový automat $A' = (K', \Sigma', \Gamma', \delta', q_0', Z_0', F')$ taký, že $L(A') = N(A)$.*

Dôkaz. Konštrukcia je rovnaká ako pre (nedeterministické) zásobníkové automaty. Automat A' teda zostrojíme nasledovne: $K' = K \cup \{q_0', q_F\}$, kde q_0' a q_F sú nové stavy, $\Sigma' = \Sigma$, $\Gamma' = \Gamma \cup \{Z_0', D\}$, kde Z_0' a D sú nové symboly, $F' = \{q_F\}$ a prechodová funkcia δ' je daná nasledovne:

$$\begin{aligned} \delta'(q_0', \varepsilon, Z_0') &= (q_0, DZ_0), & \\ \delta'(q, z, Z) &= \delta(q, z, Z), & \forall q \in K\ \forall z \in \Sigma \cup \{\varepsilon\}\ \forall Z \in \Gamma, \\ \delta'(q, \varepsilon, D) &= (q_F, \varepsilon), & \forall q \in K. \end{aligned}$$

V poslednom riadku definície prechodovej funkcie je prepísanie symbolu D na ε očividne nepodstatné. Dôležitý je len prechod do (jediného) akceptačného stavu a skutočnosť, že zo stavu q_F nie sú definované žiadne ďalšie prechody. □

Akceptácia prázdnym zásobníkom je ale slabšia, než akceptácia stavom, čo vyplýva z nasledujúcej vety.

Veta 16. *Existuje jazyk $L \in \mathscr{L}_{detCF}$, pre ktorý neexistuje žiaden deterministický zásobníkový automat A taký, že $N(A) = L$.*

Dôkaz. Napríklad jazyk $L = \{a, b\}^*$ je regulárny, a teda deterministický bezkontextový. Jazyk L však nie je bezprefixový, a preto podľa vety 14 neexistuje žiaden deterministický zásobníkový automat akceptujúci jazyk L prázdnym zásobníkom. □

Bezprefixovosť jazyka sa dá jednoduchým spôsobom zaručiť pridaním špeciálnej „zarážky“ \$ na koniec každého slova. Dokážeme, že pre každý deterministický bezkontextový jazyk je takto upravený jazyk akceptovaný nejakým DPDA prázdnym zásobníkom.

Veta 17. *Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je deterministický zásobníkový automat. Potom existuje deterministický zásobníkový automat $A = (K', \Sigma', \Gamma', \delta', q_0', Z_0', F')$ taký, že $N(A) = L(A)\$$, kde \$ je nový symbol.*

Dôkaz. Bez ujmy na všeobecnosti môžeme predpokladať, že automat A je „nezasekávajúci sa“. To okrem iného znamená, že automat A v žiadnom svojom výpočte nevyprázdni zásobník. Potom stačí pridať do každého akceptačného stavu prechod na \$, ktorý „umožní“ vyprázdniť zásobník.

Formálne, $K' = K \cup \{q_{kon}\}$, kde q_{kon} je nový stav, $\Sigma' = \Sigma \cup \{\$\}$, $\Gamma' = \Gamma$, $q_0' = q_0$, $Z_0' = Z_0$, $F' = \emptyset$ a prechodová funkcia δ' je daná nasledovne:

$$\begin{aligned} \delta'(q, z, Z) &= \delta(q, z, Z), & \forall q \in K\ \forall z \in \Sigma \cup \{\varepsilon\}\ \forall Z \in \Gamma, \\ \delta'(q, \$, Z) &= (q_{kon}, Z), & \forall q \in F\ \forall Z \in \Gamma, \\ \delta'(q_{kon}, \varepsilon, Z) &= (q_{kon}, \varepsilon), & \forall Z \in \Gamma. \end{aligned}$$

□

Nasledujúca veta poskytuje úplnú charakterizáciu triedy jazykov akceptovaných deterministickými zásobníkovými automatmi prázdnym zásobníkom.

Veta 18. *Nech $L \in \mathscr{L}_{detCF}$. Deterministický zásobníkový automat A taký, že $N(A) = L$ potom existuje práve vtedy, keď je jazyk L bezprefixový.*

Dôkaz.

- $\Rightarrow$: Vyplýva z vety 14.
- $\Leftarrow$: V deterministickom zásobníkovom automate akceptujúcom bezprefixový jazyk sú prechody vedúce z akceptačných stavov očividne zbytočné a možno ich odstrániť. Po ich odstránení možno postupovať podobne ako v dôkaze vety 17 – prechod umožňujúci vyprázdniť zásobník už ale nebude na \$, ale na ε. □

Normálny tvar deterministických zásobníkových automatov

Pri niektorých konštrukciách a dôkazoch môžu byť užitočné tvrdenia o normálnych tvaroch deterministických zásobníkových automatov kladúcich obmedzenia na slová, ktoré možno v jednom kroku pridať na zásobník. V nasledujúcom sformulujeme dve takéto tvrdenia (jedno slabšie, druhé silnejšie), pričom zakaždým uvedieme len myšlienku dôkazu a (priamočiare) konštrukcie budú prenechané čitateľovi ako jednoduché cvičenie.

Lema 3. *Nech A je deterministický zásobníkový automat. Potom existuje deterministický zásobníkový automat $B = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(B) = L(A)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\gamma \in \Gamma^*$ platí: ak $\delta(p, z, Z) = (q, \gamma)$, tak $|\gamma| \leq 2$.*

Počet symbolov na zásobníku automatu B sa teda v jednom kroku výpočtu buď nezmení vôbec, alebo sa zmení práve o jeden symbol. Konštrukcia automatu B je pomerne priamočiara – stačí prechody $\delta(p, z, Z) = (q, \gamma)$ pre $|\gamma| > 2$ rozdeliť na niekoľko prechodov vyhovujúcich podmienke z lemy 3.

Veta 19. *Nech A je deterministický zásobníkový automat. Potom existuje deterministický zásobníkový automat $B = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(B) = L(A)$ a pre všetky $p, q \in K$, $z \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ a $\gamma \in \Gamma^*$ platí: ak $\delta(p, z, Z) = (q, \gamma)$, tak*

- *(i)* $\gamma = \varepsilon$*, alebo*
- *(ii)* $\gamma = Z$*, alebo*
- *(iii)* $\gamma = ZY$ *pre nejaké* $Y \in \Gamma$.

Táto veta je zosilnením lemy 3, keďže hovorí, že v každom kroku výpočtu sa buď odstráni jeden symbol z vrchu zásobníka (operácia „pop“) alebo ostane obsah zásobníka nezmenený alebo sa na zásobník pridá práve jeden symbol Y (operácia „push“). Konštrukcia takéhoto automatu B vychádza z automatu A v normálnom tvare z lemy 3, ktorý je následne upravený tak, aby na dne zásobníka udržiaval „umelé dno“ a aby si symbol na vrchu zásobníka automatu A nepamätal na zásobníku, ale v stave. Ak má teda automat A na zásobníku slovo γZ, automat B má na zásobníku slovo $D\gamma$ a symbol Z si pamätá v stave. Pri takejto reprezentácii je zrejmé, že všetky prechody automatu A možno simulovať tak, aby bola splnená niektorá z podmienok (*i*) až (*iii*).

Predpovedajúce stroje

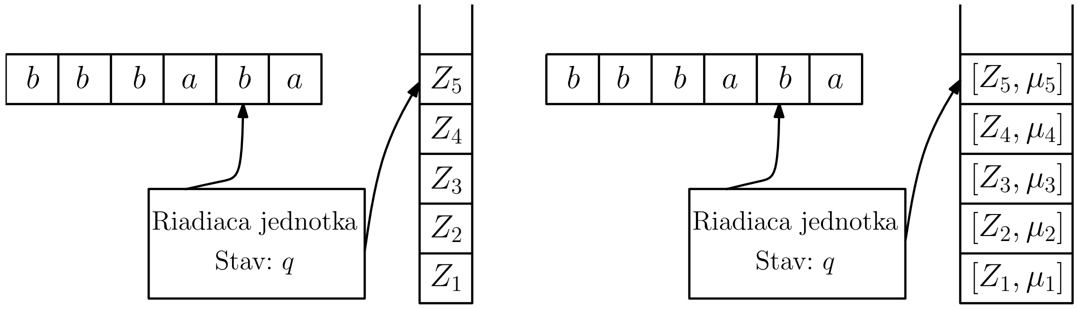
Nech $A = (K_A, \Sigma, \Gamma, \delta_A, p_0, Z_0, F_A)$ je deterministický zásobníkový automat v normálnom tvare z vety 19. Nech $B = (K_B, \Sigma, \delta_B, r_0, F_B)$ je deterministický konečný automat. *Predpovedajúci stroj* (angl. *predicting machine*) $\pi(A, B)$ je deterministický zásobníkový automat, ktorý je ako A, ale na zásobníku si okrem zásobníkových symbolov automatu A pamätá aj ďalšie informácie ohľadom výpočtov automatov A a B. Presnejšie, každý zásobníkový symbol Z v konfigurácii automatu A je v konfigurácii predpovedajúceho stroja $\pi(A, B)$ nahradený nejakou dvojicou $[Z, \mu]$, kde μ je množina dvojíc stavov $(p, r) \in K_A \times K_B$. To je znázornené na obrázku 1.

Uvažujme teraz konfiguráciu z obrázku 1. Dvojica stavov $(p, r) \in K_A \times K_B$ je v množine μ_5 práve vtedy, keď existuje slovo $w \in \Sigma^*$ také, že:

- (*i*) Pre nejaké $p_F \in F_A$ a $\alpha \in \Gamma^*$ platí $(p, w, Z_1 \ldots Z_5) \vdash_A^* (p_F, \varepsilon, \alpha)$.
- (*ii*) Pre nejaké $r_F \in F_B$ platí $(r, w) \vdash_B^* (r_F, \varepsilon)$.

Vo všeobecnosti, ak je predpovedajúci stroj $\pi(A, B)$ v konfigurácii, kde na zásobníku je slovo $[Z_1, \mu_1][Z_2, \mu_2] \ldots [Z_k, \mu_k]$, tak dvojica stavov $(p, r) \in K_A \times K_B$ je v množine μ_k práve vtedy, keď existuje slovo $w \in \Sigma^*$ také, že:

- (*i*) Pre nejaké $p_F \in F_A$ a $\alpha \in \Gamma^*$ platí $(p, w, Z_1 \ldots Z_k) \vdash_A^* (p_F, \varepsilon, \alpha)$.
- (*ii*) Pre nejaké $r_F \in F_B$ platí $(r, w) \vdash_B^* (r_F, \varepsilon)$.



(a) Konfigurácia automatu A. **(b)** Zodpovedajúca konfigurácia stroja $\pi(A, B)$.

Obr. 1: Predpovedajúci stroj $\pi(A, B)$ možno interpretovať ako deterministický zásobníkový automat A so zásobníkovými symbolmi rozšírenými o informáciu ohľadom výpočtov automatov A a B. Tá je daná v podobe množín μ obsahujúcich dvojice stavov $(p, r) \in K_A \times K_B$.

Typické použitie predpovedajúceho stroja je využiť informáciu zapamätanú v množine μ na vrchu zásobníka a zistiť, či obsahuje dvojicu (q, r_0), kde q je aktuálny stav predpovedajúceho stroja (a teda aj automatu A po rovnakom počte krokov výpočtu) a r_0 je počiatočný stav konečného automatu B. Ak áno, slovo u doposiaľ prečítané predpovedajúcim strojom (resp. automatom A) možno predĺžiť o nejaký sufix $v \in L(B)$ tak, že $uv \in L(A)$.[3] Informácie uložené v zásobníkových symboloch tak umožňujú *predpovedať*, či doposiaľ prečítané slovo možno predĺžiť o nejaký sufix z $L(B)$ tak, aby výsledné slovo bolo z $L(A)$.

V nasledujúcom popíšeme *konštrukciu* predpovedajúceho stroja $\pi(A, B)$. Pri nej budeme využívať nasledujúce označenia:

- Pre $p \in K_A$ a $Z \in \Gamma$ označíme symbolom $A_{p,Z}$ automat $(K_A, \Sigma, \Gamma, \delta_A, p, Z, F_A)$. Ide teda o automat A s počiatočným stavom zmeneným na p a počiatočným zásobníkovým symbolom zmeneným na Z_0.

- Pre $p \in K_A$ označíme symbolom $N_p(A)$ jazyk $\{w \in \Sigma^* \mid (p_0, w, Z_0) \vdash_A^* (p, \varepsilon, \varepsilon)\}$. Ide teda o jazyk všetkých slov w, ktoré automat A akceptuje prázdnym zásobníkom tak, že výpočet skončí v stave p. Podobnú notáciu budeme používať aj pre iné zásobníkové automaty.

- Pre $r \in K_B$ označíme symbolom B_r automat $(K_B, \Sigma, \delta_B, r, F_B)$. Ide teda o konečný automat B s počiatočným stavom zmeneným na r.

- Pre $r \in K_B$ označíme symbolom $L_r(B)$ jazyk $\{w \in \Sigma^* \mid (r_0, w) \vdash_B^* (r, \varepsilon)\}$. Ide teda o jazyk akceptovaný konečným automatom B s množinou akceptačných stavov zmenenou na $\{r\}$.

Pre predpovedajúci stroj $\pi(A, B)$ bude platiť $\pi(A, B) = (K_A, \Sigma, \Gamma \times 2^{K_A \times K_B}, \delta, p_0, [Z_0, \mu_0], F_A)$, pričom ostáva špecifikovať μ_0 a prechodovú funkciu δ. Množinu μ_0 definujeme ako

$$\mu_0 = \{(p, r) \in K_A \times K_B \mid L(A_{p,Z_0}) \cap L(B_r) \neq \emptyset\}.$$

Množina μ_0 teda pozostáva z dvojíc $(p, r) \in K_A \times K_B$, pre ktoré existuje slovo $w \in \Sigma^*$ také, že do akceptačného prejde na slovo w ako automat A z konfigurácie so stavom p a obsahom zásobníka Z_0, tak aj automat B zo stavu r. Podmienka kladená na predpovedajúci stroj je tak splnená. Ostáva poznamenať, že množinu μ_0 možno vypočítať algoritmicky, keďže jazyk $L(A_{p,Z_0}) \cap L(B_r)$ je bezkontextový (pretože ide o prienik bezkontextového jazyka s regulárnym) a problém prázdnosti je pre bezkontextové jazyky rozhodnuteľný.

Prechodová funkcia δ bude daná nasledovne (budeme využívať, že automat A je v normálnom tvare z vety 19):

[3] Túto vlastnosť predpovedajúcich strojov teda možno využiť na dôkaz, že trieda $\mathscr{L}_{detCF}$ je uzavretá na pravý kvocient s regulárnym jazykom.

- Pre všetky $p, p' \in K_A$, $z \in \Sigma \cup \{\varepsilon\}$ a $Z \in \Gamma$ také, že $\delta_A(p, z, Z) = (p', \varepsilon)$ bude pre všetky $\mu \in 2^{K_A \times K_B}$ definovaný prechod $\delta(p, z, [Z, \mu]) = (p', \varepsilon)$.
- Pre všetky $p, p' \in K_A$, $z \in \Sigma \cup \{\varepsilon\}$ a $Z \in \Gamma$ také, že $\delta_A(p, z, Z) = (p', Z)$ bude pre všetky $\mu \in 2^{K_A \times K_B}$ definovaný prechod $\delta(p, z, [Z, \mu]) = (p', [Z, \mu])$. Obsah zásobníka totiž ostáva nezmenený, a teda aj množina dvojíc stavov z definície množiny μ ostáva nezmenená.
- Pre všetky $p, p' \in K_A$, $z \in \Sigma \cup \{\varepsilon\}$ a $Z, Y \in \Gamma$ také, že $\delta_A(p, z, Z) = (p', ZY)$ bude pre všetky $\mu \in 2^{K_A \times K_B}$ definovaný prechod $\delta(p, z, [Z, \mu]) = (p', [Z, \mu][Y, \eta])$, kde η obsahuje dvojice stavov $(\overline{p}, \overline{r})$ také, že:

  a) Buď $L(A_{\overline{p},Y}) \cap L(B_{\overline{r}}) \neq \emptyset$. To zodpovedá tým slovám $w \in \Sigma^*$, ktoré spĺňajú podmienku z definície predpovedajúceho stroja pre dvojicu $(\overline{p}, \overline{r})$ a zároveň stroj A vo výpočte na slove w nevyužije žiaden zásobníkový symbol, ktorý je na zásobníku pod Y.

  b) Alebo pre nejaké $(s, t) \in \mu$ je $N_s(A_{\overline{p},Y}) \cap L_t(B_{\overline{r}}) \neq \emptyset$. To zodpovedá tým slovám $w \in \Sigma^*$, ktoré spĺňajú podmienku z definície predpovedajúceho stroja pre dvojicu $(\overline{p}, \overline{r})$ a zároveň automat $A_{\overline{p},Y}$ po prvý raz „narazí“ na daný výskyt zásobníkového symbolu Z v stave s, pričom automat $B_{\overline{r}}$ po prečítaní rovnakej časti slova príde do stavu t.

  Opäť pritom využívame skutočnosť, že problém prázdnosti je pre bezkontextové jazyky rozhodnuteľný.

Ďalšie uzáverové vlastnosti triedy $\mathscr{L}_{detCF}$

Keďže štandardné uzáverové vlastnosti triedy $\mathscr{L}_{detCF}$ sú o poznanie horšie, než v prípade tried Chomského hierarchie, pri dôkazoch sa môžu zísť poznatky o uzavretosti tejto triedy na rôzne menej klasické operácie. V nasledujúcom preto dokážeme uzavretosť triedy $\mathscr{L}_{detCF}$ na operácie MIN a MAX, ktoré najprv zadefinujeme.

Definícia 5. Nech L je jazyk. Jazyk MIN(L) je definovaný nasledovne:

$$\mathrm{MIN}(L) = \{w \in L \mid \forall u, v \in \Sigma_L^* : (w = uv \ \wedge \ u \in L) \Rightarrow v = \varepsilon\}.$$

Jazyk MIN(L) teda obsahuje všetky slová w také, že w je v L a súčasne žiaden *vlastný* prefix slova w nie je v L.

Definícia 6. Nech L je jazyk. Jazyk MAX(L) je definovaný nasledovne:

$$\mathrm{MAX}(L) = \{w \in L \mid \forall u \in \Sigma_L^* : wu \in L \Rightarrow u = \varepsilon\}.$$

Jazyk MAX(L) teda obsahuje všetky slová w také, že w je v L a súčasne w nie je *vlastným* prefixom žiadneho slova v L.

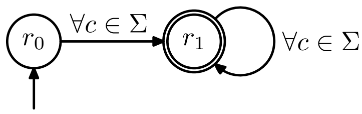
Veta 20. *Trieda $\mathscr{L}_{detCF}$ je uzavretá na operáciu* MIN.

Dôkaz. Nech $L \in \mathscr{L}_{detCF}$ je deterministický bezkontextový jazyk akceptovaný deterministickým zásobníkovým automatom $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$. Jazyk MIN($L$) je akceptovaný deterministickým zásobníkovým automatom $A' = (K, \Sigma, \Gamma, \delta', q_0', Z_0, F)$, ktorý vznikne z A odstránením všetkých prechodov z akceptačných stavov. Pre všetky $q \in K - F$, $z \in \Sigma \cup \{\varepsilon\}$ a $Z \in \Gamma$ teda bude $\delta'(q, z, Z) = \delta(q, z, Z)$ a pre všetky $q \in F$, $z \in \Sigma \cup \{\varepsilon\}$ a $Z \in \Gamma$ bude $\delta'(q, z, Z) = \emptyset$. □

Veta 21. *Trieda $\mathscr{L}_{detCF}$ je uzavretá na operáciu* MAX.

Dôkaz. Nech $L \subseteq \Sigma^*$ je deterministický bezkontextový jazyk. Nech A je deterministický zásobníkový automat taký, že $L(A) = L$. Nech B je deterministický konečný automat taký, že $L(B) = \Sigma^+$, ktorý je daný ako na obrázku 2.

Nech teraz $\pi(A, B)$ je predpovedajúci stroj zodpovedajúci automatom A a B. Nech A' je deterministický zásobníkový automat, ktorý vznikne úpravou stroja $\pi(A, B)$ tak, aby mohol akceptovať práve vtedy, keď je v akceptačnom stave $p \in F$ a súčasne je buď zásobník prázdny, alebo je na jeho vrchu symbol $[Z, \mu]$, kde $[p, r_0] \notin \mu$. Automat A' teda akceptuje slovo u práve vtedy, keď $u \in L$ a zároveň neexistuje žiadne slovo $v \in \Sigma^+$ také, že $uv \in L$. □



Obr. 2: Deterministický konečný automat B.

Ako ukážkovú aplikáciu vety o uzavretosti triedy $\mathscr{L}_{detCF}$ na operáciu MIN v nasledujúcom dokážeme, že jazyk $L = \{ww^R \mid w \in \{a,b\}^*\}$ nie je deterministický bezkontextový.

Tvrdenie 3. *Jazyk $L = \{ww^R \mid w \in \{a,b\}^*\}$ je v $\mathscr{L}_{CF} - \mathscr{L}_{detCF}$.*

Dôkaz. Skutočnosť $L \in \mathscr{L}_{CF}$ je dostatočne známa. Ostáva teda dokázať $L \notin \mathscr{L}_{detCF}$. Za účelom sporu predpokladajme, že $L \in \mathscr{L}_{detCF}$. Je pomerne jednoduchou úlohou dokázať, že v takom prípade je deterministický bezkontextový aj jazyk

$$L_2 := L - \{\varepsilon\} = \{ww^R \mid w \in \{a,b\}^+\}.$$

Keďže je trieda $\mathscr{L}_{detCF}$ uzavretá na prienik s regulárnym jazykom, je deterministický bezkontextový aj jazyk

$$L_3 := L_2 \cap (ab)^*(ba)^*(ab)^*(ba)^* = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i,j \in \mathbb{N};\ i > 0 \ \vee\ j > 0\}.$$

Z uzavretosti triedy $\mathscr{L}_{detCF}$ na operáciu MIN ďalej vyplýva, že je deterministický bezkontextový aj jazyk

$$L_4 := \mathrm{MIN}(L_3) = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid 0 \leq j < i\}.$$

Jazyk $\mathrm{MIN}(L_3)$ je daný uvedeným spôsobom, pretože pre $i \leq j$ je slovo $(ab)^i(ba)^i \in L_3$ prefixom slova $(ab)^i(ba)^j(ab)^j(ba)^i$. V opačnom prípade nemá slovo $(ab)^i(ba)^j(ab)^j(ba)^i$ žiaden prefix z jazyka L_3.

Nech teraz $h\colon \{a,b\}^* \to \{a,b\}^*$ je homomorfizmus taký, že $h(a) = ab$ a $h(b) = ba$. Z uzavretosti triedy $\mathscr{L}_{detCF}$ na inverzný homomorfizmus potom vyplýva, že jazyk

$$L_5 := h^{-1}(L_4) = \{a^ib^ja^jb^i \mid 0 \leq j < i\}.$$

je deterministický bezkontextový. Pomocou pumpovacej lemy však o tomto jazyku možno ľahko dokázať, že nie je ani bezkontextový. Dospeli sme teda k sporu. □

Rozhodovacie problémy pre deterministické bezkontextové jazyky

Na prednáške bolo dokázaných viacero výsledkov o (ne)rozhodnuteľnosti niektorých štandardných problémov pre jazyky tried Chomského hierarchie. V nasledujúcom vyšetríme rozhodnuteľnosť týchto problémov pre deterministické bezkontextové jazyky.

Veta 22. *Problém príslušnosti komplementu do $\mathscr{L}_{detCF}$ je pre deterministické bezkontextové jazyky triviálny, a teda rozhodnuteľný.*

Dôkaz. Vyplýva z uzavretosti triedy $\mathscr{L}_{detCF}$ na komplement. □

Veta 23. *Problém prázdnosti je pre deterministické bezkontextové jazyky rozhodnuteľný.*

Dôkaz. Vyplýva z rozhodnuteľnosti tohto problému pre bezkontextové jazyky. □

Veta 24. *Problém konečnosti je pre deterministické bezkontextové jazyky rozhodnuteľný.*

Dôkaz. Vyplýva z rozhodnuteľnosti tohto problému pre bezkontextové jazyky. □

Veta 25. *Problém ekvivalencie s regulárnym jazykom je pre deterministické bezkontextové jazyky rozhodnuteľný.*

Dôkaz. Nech A je deterministický zásobníkový automat a B je deterministický konečný automat. Zrejme $L(A) = L(B)$ práve vtedy, keď

$$(L(A) \cap L(B)^C) \cup (L(A)^C \cap L(B)) = \emptyset. \tag{1}$$

Keďže sú triedy $\mathscr{L}_{detCF}$ a $\mathscr{R}$ efektívne[4] uzavreté na komplement, trieda $\mathscr{L}_{detCF}$ je efektívne uzavretá na prienik s regulárnym jazykom a trieda $\mathscr{L}_{CF}$ je efektívne uzavretá na zjednotenie, na základe kódov automatov A a B možno algoritmicky zostrojiť kód (vo všeobecnosti nedeterministického) zásobníkového automatu akceptujúceho jazyk $(L(A) \cap L(B)^C) \cup (L(A)^C \cap L(B))$. Na overenie rovnosti (1) už potom stačí zistiť, či tento automat akceptuje prázdny jazyk, čo je (ako je známe z prednášky) rozhodnuteľný problém. □

Veta 26. *Problém regulárnosti je pre deterministické bezkontextové jazyky rozhodnuteľný.*

Dôkaz. Presahuje rámec týchto poznámok. □

Na dôkaz nerozhodnuteľnosti problému prázdnosti prieniku využijeme tzv. *jazyk platných výpočtov* deterministického Turingovho stroja.

Definícia 7. Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj. *Jazyk platných výpočtov* stroja A je jazyk $L_{PV}(A)$ pozostávajúci zo slov

$$\begin{cases} w_0\#w_1^R\#w_2\#w_3^R\#\ldots\#w_k, & k \text{ párne} \\ w_0\#w_1^R\#w_2\#w_3^R\#\ldots\#w_k^R, & k \text{ nepárne} \end{cases}$$

takých, že:

(*i*) Každé w_i, $i = 0, \ldots, k$, je konfigurácia stroja A (chápaná ako slovo, v ktorom „stav určuje pozíciu hlavy“).

(*ii*) Konfigurácia w_0 je počiatočná.

(*iii*) Konfigurácia w_k je akceptačná.

(*iv*) Pre $i = 1, \ldots, k$ platí $w_{i-1} \vdash w_i$.

Dôkazy nasledujúcich dvoch liem prenechávame čitateľovi ako jednoduché cvičenie.

Lema 4. *Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj. Potom existujú jazyky $L_1, L_2 \in \mathscr{L}_{detCF}$ také, že $L_{PV}(A) = L_1 \cap L_2$. Kódy deterministických zásobníkových automatov akceptujúcich jazyky L_1 a L_2 sú navyše algoritmicky zostrojiteľné na základe kódu stroja A.*

Lema 5. *Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj. Jazyk $L_{PV}(A)$ je prázdny práve vtedy, keď je prázdny jazyk $L(A)$.*

Veta 27. *Problém prázdnosti prieniku je pre deterministické bezkontextové jazyky nerozhodnuteľný.*

Dôkaz. Redukciou problému prázdnosti pre rekurzívne vyčísliteľné jazyky na problém prázdnosti prieniku pre deterministické bezkontextové jazyky.

Nech B je deterministický Turingov stroj. Nech A_1, A_2 sú deterministické zásobníkové automaty také, že $L(A_1) \cap L(A_2) = L_{PV}(B)$. Podľa lemy 4 sú kódy automatov A_1 a A_2 algoritmicky zostrojiteľné na základe kódu stroja B. Podľa lemy 5 je navyše jazyk $L_{PV}(B) = L(A_1) \cap L(A_2)$ prázdny práve vtedy, keď je prázdny jazyk $L(B)$.

Na rozhodnutie prázdnosti jazyka $L(B)$ teda stačí zostrojiť kódy automatov A_1 a A_2 a použiť ich ako vstupy pre „podprogram“ rozhodujúci prázdnosť prieniku pre deterministické bezkontextové jazyky. □

[4] Na základe kódu automatu A možno algoritmicky zostrojiť kód automatu akceptujúceho $L(A)^C$.

Veta 28. *Problém inklúzie je pre deterministické bezkontextové jazyky nerozhodnuteľný.*

Dôkaz. Zrejme $L_1 \cap L_2 = \emptyset$ práve vtedy, keď $L_1 \subseteq L_2^C$. Keďže je trieda $\mathscr{L}_{detCF}$ efektívne uzavretá na komplement, bolo by v prípade rozhodnuteľnosti problému inklúzie možné rozhodovať aj nerozhodnuteľný problém prázdnosti prieniku. □

Na dôkaz nerozhodnuteľnosti problému príslušnosti prieniku deterministických bezkontextových jazykov do $\mathscr{L}_{detCF}$ je potrebná ešte jedna vlastnosť jazyka platných výpočtov Turingovho stroja, ktorej (jednoduchý) dôkaz opäť prenechávame čitateľovi. (Predpokladáme, že deterministický Turingov stroj sa zastaví akonáhle príde do akceptačného stavu.)

Lema 6. *Nech A je deterministický Turingov stroj, ktorý na každom vstupe urobí aspoň tri kroky.*[5] *Jazyk $L_{PV}(A)$ je konečný práve vtedy, keď je konečný jazyk $L(A)$. Ak je jazyk $L(A)$ nekonečný, jazyk $L_{PV}(A)$ nie je bezkontextový.*

Veta 29. *Problém príslušnosti prieniku do $\mathscr{L}_{detCF}$ je pre deterministické bezkontextové jazyky nerozhodnuteľný.*

Dôkaz. Redukciou nerozhodnuteľného problému konečnosti pre rekurzívne vyčísliteľné jazyky na problém príslušnosti prieniku deterministických bezkontextových jazykov do $\mathscr{L}_{detCF}$.

Nech B je deterministický Turingov stroj. Nech A_1 a A_2 sú (efektívne zostrojiteľné) deterministické zásobníkové automaty také, že $L(A_1) \cap L(A_2) = L_{PV}(B)$. Z lemy 6 potom vyplýva, že jazyk $L(B)$ je konečný práve vtedy, keď je prienik $L(A_1) \cap L(A_2)$ deterministický bezkontextový. □

Poznámka 6. Dôsledkom predchádzajúcej vety je aj nerozhodnuteľnosť príslušnosti zjednotenia a zreťazenia deterministických bezkontextových jazykov do $\mathscr{L}_{detCF}$. Čitateľa nabádame, aby tieto tvrdenia aj skutočne dokázal.

Veta 30. *Problém ekvivalencie je pre deterministické bezkontextové jazyky rozhodnuteľný.*

Dôkaz. Presahuje rámec týchto poznámok. □

Poznámka 7. Rozhodnuteľnosť problému ekvivalencie pre deterministické bezkontextové jazyky bola dlhé roky otvoreným problémom patriacim k najvýznamnejším v teoretickej informatike. Od prvej zmienky tohto problému v roku 1966 bolo publikovaných mnoho článkov o jeho rozhodnuteľnosti pre rôzne špeciálne triedy deterministických zásobníkových automatov a na významnosti problému pridával aj jeho súvis s niektorými problémami ekvivalencie programových schém. Prvý pokus o dôkaz rozhodnuteľnosti problému ekvivalencie publikoval v roku 1992 ukrajinský vedec Mejtus v lokálnom časopise (v ruštine). S prvým všeobecne uznaným dôkazom tejto vety prišiel až v roku 1997 Géraud Sénizergues, ktorého kompletný dôkaz bol publikovaný v roku 2001 v článku o 166 stranách. Neskôr boli publikované viaceré jednoduchšie dôkazy, spomedzi ktorých treba spomenúť najmä ten od Petra Jančara publikovaný v roku 2010 (resp. 2012).

Odkazy na literatúru

Tieto poznámky obsahovo vychádzajú z knihy Hopcrofta a Ullmana [2], z ktorej boli prebraté aj niektoré dôkazy. V tejto knihe tiež možno nájsť úplné verzie (čiastočne alebo úplne) vynechaných dôkazov.

Kľúčového historického významu je pre teóriu deterministických zásobníkových automatov predovšetkým článok Ginsburga a Greibachovej [1], v ktorom bolo po prvý raz dokázaných viacero tvrdení z týchto poznámok. Dôkaz rozhodnuteľnosti problému regulárnosti pre deterministické bezkontextové jazyky možno nájsť v článkoch Stearnsa [6] a Valianta [7].

Originál Mejtusovho článku o probléme ekvivalencie je ťažko dostupný, jeho anglický preklad vyšiel ako [4]. Úplná 166-stranová verzia Sénizerguesovho článku s dôkazom rozhodnuteľnosti problému ekvivalencie vyšla v článku [5]. Jančarov zjednodušený dôkaz vyšiel v článku [3].

[5] Zrejme ide o normálny tvar deterministických Turingových strojov.

Literatúra

[1] Ginsburg, S., Greibach, S.: Deterministic Context Free Languages. In *Information and Control*. 1966, vol. 9, no. 6, pp. 620–648.

[2] Hopcroft, J. E., Ullman, J. D.: *Introduction to Automata Theory, Languages, and Computation*. Reading : Addison-Wesley, 1979.

[3] Jančar, P.: Decidability of DPDA Language Equivalence via First-Order Grammars. In *27th Annual IEEE Symposium on Logic in Computer Science (LICS 2012)*. 2012, pp. 415–424.

[4] Meitus, V. Yu.: Decidability of the Equivalence Problem for Deterministic Pushdown Automata. In *Cybernetics and Systems Analysis*. 1992, vol. 28, no. 5, pp. 672–690.

[5] Sénizergues, G.: $L(A) = L(B)$? Decidability Results from Complete Formal Systems. In *Theoretical Computer Science*. 2001, vol. 251, no. 1–2, pp. 1–166.

[6] Stearns, R. E.: A Regularity Test for Pushdown Machines. In *Information and Control*. 1967, vol. 11, no. 3, pp. 323–340.

[7] Valiant, L. G.: Regularity and Related Problems for Deterministic Pushdown Automata. In *Journal of the ACM*. 1975, vol. 22., no. 1, pp. 1–10.