Prezentačná úloha z FoJA

Martin Širáň, 2i4

1 Zadanie

Dokážte, že pre deterministický zásobníkový automat A a regulárny jazyk R je rozhodnuteľné:

- $L(A) = R$
- $L(A) \subseteq R$
- $R \subseteq L(A)$

2 Riešenie

Najprv dokážem platnosť pomocných tvrdení, a až potom samotné tvrdenia v zadaní úlohy. Takže pekne jedno po druhom...

Lema 1. *Doplnok regulárneho jazyka je regulárny jazyk.*
Dôkaz. Uvažujme konečný deterministický automat $M = (K, \Sigma, \delta, q_0, F)$. K automatu M zostrojíme konečný deterministický automat $M^{'}$, pre ktorý bude platiť $L(M^{'}) = L(M)^c$, nasledovne: $M^{'} = (K, \Sigma, \delta, q_0, K \setminus F)$. Triviálne sa dá overiť, že $L(M^{'}) = L(M)^c$. Ak totiž $w \in L(M)$, potom sa v M pre w výpočet dostane do nejakého akceptačného stavu $q \in F$, ktorý však určite nebude akceptačným stavom v $M^{'}$, lebo $q \notin K \setminus F$, čiže $w \notin L(M^{'})$. Naopak, ak $w \notin L(M)$, potom sa výpočet M na w dostane do nejakej konfigurácie, ktorej stav nie je akceptačný, t.j. $q \notin F$, avšak v tomto prípade platí $q \in K \setminus F$, a teda $w \in L(M^{'})$. $\square$

Lema 2. *K ľubovoľnému deterministickému zásobníkovému automatu M existuje deterministický zásobníkový automat $M^{'}$ taký, že:*

- $L(M^{'}) = L(M)$
- $M^{'}$ *pri každom výpočte prečíta celý svoj vstup a potom sa zastaví.*

Dôkaz. Dôkaz tohto nie celkom triviálneho tvrdenia neuvádzam, pretože je podľa môjho názoru už mimo záberu tejto prezentačnej úlohy. Vcelku zrozumiteľný dôkaz sa však dá nájsť napríklad v [1].

Lema 3. *Doplnok deterministického bezkontextového jazyka je deterministický bezkontextový jazyk.*
Dôkaz. Uvažujme deterministický zásobníkový automat $M = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ spĺňajúci predchádzajúcu lemu 2. K automatu M zostrojíme deterministický zásobníkový automat $M^{'}$, pre ktorý bude platiť $L(M^{'}) = L(M)^c$, nasledovne: $M^{'} = (K^{'}, \Sigma, \Gamma, \delta^{'}, q_0^{'}, Z_0, F^{'})$, kde $K^{'} = K \times \{1, 2, 3\}$, $F^{'} = K \times \{3\}$. Ak $q_0 \in F$, tak $q_0^{'} = (q_0, 1)$, inak $q_0 \notin F$ a $q_0^{'} = (q_0, 2)$. Ešte predtým ako sa pozrieme na modifikovanú prechodovú funkciu $\delta^{'}$, treba vysvetliť úlohu "k-čka" v každom stave (q, k). Táto zložka nám označuje, či pôvodný automat M od posledného kroku vykonaného na skutočnom vstupnom symbole (teda nie "ε-e") dosiahol akceptačný stav. Ak áno, tak $k = 1$, ak nie, tak $k = 2$. Ak $k = 3$, stav je automaticky akceptačný pre $M^{'}$, ako vidno z definície $F^{'}$. Teraz uvažujme ľubovoľné $q, p \in K$, $Z \in \Gamma$, $\gamma \in Z^*$, $a \in \Sigma$ a pozrime sa na prechodovú funkciu $\delta^{'}$, vyzerá nasledovne:

- Pre $\delta(q, \varepsilon, Z) = (p, \gamma)$ a $(k = 1 \ \vee \ k = 2)$, definujeme $\delta^{'}((q, k), \varepsilon, Z) = ((p, k^{'}), \gamma)$, kde $k^{'} = 1$ ak $(k = 1 \ \vee \ p \in F)$, inak $(k = 2 \ \wedge \ p \notin F)$, a potom $k^{'} = 2$.
- Pre $\delta(q, a, Z) = (p, \gamma)$, definujeme $\delta^{'}((q, 1), a, Z) = \delta^{'}((q, 3), a, Z) = \delta^{'}((p, k), \gamma)$, kde $k = 1$ ak $p \in F$, inak $p \notin F$ a $k = 2$.
- Pre $\delta(q, \varepsilon, Z) = \emptyset$, definujeme $\delta^{'}((q, 2), \varepsilon, Z) = ((q, 3), Z)$.

Aj keď to možno na prvý pohľad tak neyzerá, ak si všímame len prvé zložky stavov, $M^{'}$ presne simuluje správanie sa M a zložitosť definície $\delta^{'}$ je spôsobená rôznosťou prípadov, v ktorých treba upravovať druhú "informačnú" zložku stavov (ktorej funkciu sme už vysvetlili vyššie). Za pošimnutie ešte stojí prechod do akceptačných stavov. V prípade, že $M^{'}$ je v stave $(q, 2)$ (teda od posledne spracovaného skutočného vstupu sa M nedostal do akceptačného stavu), potom pre $\delta(q, \varepsilon, Z) = \emptyset$ povolíme prechod $\delta^{'}((q, 2), \varepsilon, Z) = ((q, 3), Z)$, kde $(q, 3)$ je akceptačný stav pre $M^{'}$. Žiadnou inou cestou sa $M^{'}$

do akceptačného stavu dostat' nemôže. A teraz dôkaz tvrdenia $L(M^{'}) = L(M)^c$. Nuž ak $w \in L(M)$, potom na w sa M dostane do nejakého akceptačného stavu $q_F \in F$, a teda po prečítaní posledného symbolu w (a prípadne konečnom počte krokov na "ε") sa $M^{'}$ musí dostat' do stavu $(q_F, 1)$, pričom už prečítal celý vstup, teda môže vykonávat' kroky len na ε. Ako vidno z definície $\delta^{'}$, tieto môžu $M^{'}$ priviest' len do stavov $(q, 1)$ (pre nejaké $q \in K$), z ktorých akceptácia nie je možná, teda $w \notin L(M^{'})$. Naopak, ak $w \notin L(M)$, potom po prečítaní posledného symbolu w sa M dostane do nejakého neakceptačného stavu $q_R \in K \setminus F$. Teda $M^{'}$ bezprostredne po spracovaní posledného skutočného vstupného symbolu skončí v stave $(q_R, 2)$. Odtial'to už môže M vykonat' nanjvýš nejaký konečný počet krokov na "ε" prechádzajúc cez neakceptačné stavy a následne sa zaseknút' v nejakom stave $q \in K \setminus F$. Ako však vidno z definície $\delta^{'}$, $M^{'}$ ho bude simulovat' a skončí v stave $(q, 2)$, odkial' (kedže M nemá definovaný dal'ší krok) smie prejst' do akceptačného stavu $(q, 3)$, teda $w \in L(M^{'})$. $\square$

Lema 4. *Prienik deterministického bezkontextového jazyka s regulárnym jazykom je deterministický bezkontextový jazyk.*
Dôkaz. Uvažujme deterministický zásobníkový automat $M_1 = (K_1, \Sigma_1, \Gamma, \delta_1, q_{1_0}, Z_0, F_1)$ a deterministický konečný automat $M_2 = (K_2, \Sigma_2, \delta_2, q_{2_0}, F_2)$. Deterministický zásobníkový automat M_3, pre ktorý platí $L(M_3) = L(M_1) \cap L(M_2)$, zostrojíme nasledovne: $M_3 = (K_1 \times K_2, \Sigma_1 \cup \Sigma_2, \Gamma, \delta_3, (q_{1_0}, q_{2_0}), Z_0, F_1 \times F_2)$. Uvažujme l'ubovol'né $q_1, p_1 \in K_1$, $q_2, p_2 \in K_2$, $a \in \Sigma_1 \cup \Sigma_2$, $Z \in \Gamma$, $\gamma \in \Gamma^*$, prechodová funkcia δ_3 má tvar:

- Pre $\delta_1(q_1, a, Z) = (p_1, \gamma)$ a $\delta_2(q_2, a) = p_2$ definujeme $\delta_3((q_1, q_2), a, Z) = ((p_1, p_2), \gamma)$.
- Pre $\delta_1(q_1, \varepsilon, Z) = (p_1, \gamma)$ definujeme $\delta_3((q_1, q_2), \varepsilon, Z) = ((p_1, q_2), \gamma)$.

Aj ked' tento zápis na prvý pohl'ad vyzerá hrozivo, M_3 sa správa vel'mi jednoducho. V prípade "ne-ε-ového" kroku jednoducho vzhl'adom na prvú zložku svojho stavu, vstupný symbol a zásobníkový symbol odsimuluje jeden krok M_1 a zároveň pre druhú zložku svojho stavu a rovnaký vstupný symbol odsimuluje jeden krok M_2. V prípade, že jeden z automatov M_1 a M_2 nemá definovaný d'alší krok, bude tomu tak aj v automate M_3. V prípade kroku na "ε" ignorujeme výpočet v M_2, pretože ten z definície nemôže robit' kroky na "ε". Odsimuluje sa teda iba krok v M_1. Opät', v prípade že M_1 nemá definovaný d'alší krok, nebude ho mat' definovaný ani automat

M_3. S týmto na zreteli je triviálne ukázat', že $L(M_3) = L(M_1) \cap L(M_2)$. Ak $w \in L(M_1) \cap L(M_2)$, potom výpočet v M_1 na w privedie automat do nejakého akceptačného stavu q_{1F} a v M_2 do nejakého akceptačného stavu q_{2F}. Z konštrukcie M_3 však vidno, že potom aj výpočet M_3 na w príde do stavu $(q_{1F}, q_{2F}) \in F_1 \times F_2$, teda $w \in L(M_3)$. Naopak, ak $w \notin L(M_1)$ alebo $w \notin L(M_2)$, teda M_1 alebo M_2 sa na w bud' zasekne alebo neskončí v akceptačnom stave, potom ako už bolo uvedené vyššie, aj automat M_3 sa bud' zasekne alebo neskončí v akceptačnom stave, teda $w \notin L(M_3)$. □

Lema 5. *Problém prázdnosti bezkontextového jazyka je rozhodnutel'ný.*
Dôkaz. Uvažujme l'ubovol'nú bezkontextovú gramatiku $G = (N, T, P, \sigma)$. Chceme ukázat' existenciu algoritmu rozhodujúceho problém $L(G) = \emptyset$. Najprv dokážeme, že pre G sa dá algoritmicky vygenerovat' (na n krokov) množina H_n, obsahujúca vsetky tie neterminály, z ktorých sa v G dá odvodit' nejaké slovo. Induktívne generovanie H_n vyzerá nasledovne:

- $H_0 = \{\xi \in N;\ (\exists w \in T^*)(\xi \to w \in P)\}$
- $H_{i+1} = H_i \cup \{\xi \in N;\ (\exists w \in (T \cup H_i)^*)(\xi \to w \in P)\}$

Zo samotnej konštrukcie množín H_i vidno, že akonáhle $H_n = H_{n+1}$ (pre nejaké n), potom $H_n = H_{n+k}$ (pre l'ubovol'né prirodzené k). A ked'že z definície bezkontextovej gramatiky vieme, že množina N neterminálov je konečná, mohutnost' H_i nemôže donekonečna narastat', čím je existencia vyššie uvedeného n zaručená. Teraz ešte treba dokázat', že z každého neterminálu obsiahnutého v H_n sa dá vygenerovat' v G nejaké slovo. Matematická indukcia vzhl'adom na počet krokov výpočtu H_n:

- Báza indukcie: triviálne z definície H_0 plynie, že pre l'ubovol'né $\xi \in H_0$ sa dá v G vygenerovat' slovo.
- Indukčný predpoklad: Pre každé $\xi \in H_i$ sa dá v G nájst' odvodenie nejakého slova z ξ.
- Induktívny krok: uvažujme $\xi \in H_{i+1}$. Potom bud' $\xi \in H_i$ (v tomto prípade vzhl'adom na IP niet čo dokazovat') alebo platí $\xi \Rightarrow_G w$, kde $w \in (T \cup H_i)^*$. Avšak podl'a IP, z každého neterminálu patriaceho do H_i vieme v G odvodit' slovo. Teda zjavne aj ξ sa bude dat' v G prepísat' na nejaké slovo.

Keď si to celé zhrnieme, už sme ukázali, že pre danú bezkontextovú gramatiku G existuje n (s horeuvedenými vlastnosťami) a množina neterminálov H_n, ktorá je jednak algoritmicky konečne generovateľná a jednak predstavuje množinu všetkých tých neterminálov v G, z ktorých sa dá v G odvodiť slovo. Algoritmus teraz už len overí prítomnosť počiatočného neterminálu σ v H_n. Ak $\sigma \notin H_n$, potom zjavne $L(G) = \emptyset$, inak $\sigma \in H_n$ a $L(G) \neq \emptyset$. $\square$

Veta 1. *Pre deterministický zásobníkový automat A a regulárny jazyk R je rozhodnuteľné, či $L(A) \subseteq R$.*
Dôkaz. Podľa lemy 1 vieme, že $R^c \in \mathcal{R}$. Ďalej vďaka leme 4 dostávame $L(A) \cap R^c \in \mathcal{L}_{\mathcal{DCF}}$. Vieme, že $\mathcal{L}_{\mathcal{DCF}} \subset \mathcal{L}_{\mathcal{CF}}$, podľa lemy 5 je teda problém $L(A) \cap R^c = \emptyset$ rozhodnuteľný. Avšak platí (vo všeobecnosti pre ľubovoľné množiny), že $L(A) \subseteq R \Leftrightarrow L(A) \cap R^c = \emptyset$. Teda problém $L(A) \subseteq R$ je rozhodnuteľný. $\square$

Veta 2. *Pre deterministický zásobníkový automat A a regulárny jazyk R je rozhodnuteľné, či $R \subseteq L(A)$.*
Dôkaz. Podľa lemy 3 vieme, že $L(A)^c \in \mathcal{L}_{\mathcal{DCF}}$. Ďalej vďaka leme 4 dostávame $R \cap L(A)^c \in \mathcal{L}_{\mathcal{DCF}}$. Vieme, že $\mathcal{L}_{\mathcal{DCF}} \subset \mathcal{L}_{\mathcal{CF}}$, podľa lemy 5 je teda problém $R \cap L(A)^c = \emptyset$ rozhodnuteľný. Avšak platí (vo všeobecnosti pre ľubovoľné množiny), že $R \subseteq L(A) \Leftrightarrow R \cap L(A)^c = \emptyset$. Teda problém $R \subseteq L(A)$ je rozhodnuteľný. $\square$

Veta 3. *Pre deterministický zásobníkový automat A a regulárny jazyk R je rozhodnuteľné, či $L(A) = R$.*
Dôkaz. Podľa viet 1 a 2 vieme, že problémy $L(A) \subseteq R$ a $R \subseteq L(A)$ sú rozhodnuteľné. Avšak platí (vo všeobecnosti pre ľubovoľné množiny), že $L(A) = R \Leftrightarrow L(A) \subseteq R \wedge R \subseteq L(A)$. Teda problém $L(A) = R$ je rozhodnuteľný. $\square$

3 Referencie

- [1] J.E. Hopcroft, J.D. Ullman, *Formal languages and their relation to automata*, Addison-Wesley Publishing Company (1969), slovenský preklad: B. Rovan, P. Mikulecký, *Formálne jazyky a automaty*, Vydavateľstvo Alfa (1978)