Riešenia prvej sady bodovaných domácich úloh

9. apríla 2018

Poznámka na úvod: Tento materiál je neoficiálny. Ak obsahuje chybu, vaše riešenie s rovnakou chybou nie je správne. Ak natrafíte na chybu, cvičiaci bude veľmi vďačný, ak ho na ňu upozorníte.

Úloha 1. Dané je tvrdenie: „Ak L je bezkontextový jazyk a R je regulárny jazyk, tak existuje a-prekladač M taký, že $M(L) = R$.“ Zistite a dokážte, pre ktoré dvojice L, R toto tvrdenie platí. Vaše tvrdenie poriadne dokážte.

Riešenie. Ukážeme, že tvrdenie platí pre všetky dvojice (L, R) také, že L je neprázdny bezkontextový jazyk a R je ľubovoľný regulárny jazyk a taktiež pre dvojicu $(\emptyset, \emptyset)$.

V prvom rade si uvedomíme, že pre ľubovolný a-prekladač M platí $M(\emptyset) = \emptyset$. Toto je naozaj ľahko vidno z definície prekladu a-prekladačom. A intuitícia taktiež našepkáva, že ak nemám nič, čo by som prekladal, sotva niečo preložím. Uporozňujeme, že v intuitívnom prístupe nič znamená prázdna množnia, nie jazyk obsahujúci iba prázdne slovo. Prázdne slovo už niečo je. Z tohoto nám vyplýva, že tvrdenie platí pre dvojicu $(\emptyset, \emptyset)$ a taktiež, že neplatí pre žiadnu dvojciu $(\emptyset, R)$ kde R je neprázdny regulárny jazyk.

Teraz ostáva dokázať, že pre dvojice (L, R) také, že L je neprázdny bezkontextový jazyk a R je ľubovoľný regulárny jazyk tvrdenie platí. Toto dokážeme priamo konštrukciou potrebného a-prekladača. Intuitívne si najprv všimnime, že na a-prekladač sa vieme pozerať ako na vylepšený konečný automat. Taktiež ako konečný automat má stavy, medzi ktorými sa pohybuje na základe toho, čo číta na vstupe, pričom vstup číta práve raz, z ľava do prava a nič na vstupe neprepisuje. Iba si môže občas niečo poznačiť na výstupnú pásku. V tomto momente si môžeme všimnúť ďalšiu vec. Čo ak potrebujem pomocou a-prekladača „vygenerovať“ nejaký regulárny jazyk R? V prvom kroku našej úvahy sa zamyslíme nad tým, ako by šiel R vygenerovať z jediného slova - ε. Kedže R je regulárny, existuje nejaký DKA A taký, že $L(A) = R$. Náš a-prekladač by mohol namiesto čítania, ktoré daný DKA robil, zapisovať na pásku, pričom by menil stav podla toho, čo práve zapísal a všetky prechody by robil čítajúc zo vstupu ε. Pre priblíženie tejto myšlienky sa môžeme pozrieť na vec pomocou prechodových diagramov. Daný prechodový diagram a-prekladača (ktorý nám naozaj jednoznačne definuje prechodovú funkciu a-prekladača) spravíme z prechodového diagramu DKA A tak, že jednoducho na každú šípku napíšeme, že číta ε a zapisuje to, čo pomocou danej šípky DKA A čítal. Teraz sa už dá jednoducho nahliadnuť, že nami popísaná konštrukcia bude fungovať. Ešte je pred nami jeden krok. My nechceme ľubovoľný R „generovať“ z $\{\varepsilon\}$, ale z ľubovolného neprázdneho bezkontextového L. Tu nám stačí uvedomiť si, že ak najprv v počiatočnom stave konštruovaného a-prekladača prečítame celý vstup, nech je akýkoľvek a potom prejdeme do počiatočného stavu a-prekladača, ktorý z ε „generuje“ celý jazyk R, tak máme presne to čo potrebujeme.

Poďme formálne. Nech L je neprázdny bezkontextový jazyk a R je ľubovoľný regulárny jazyk. Potom existuje DKA $A = (K_A, \Sigma_A, \delta_A, q_0, F_A)$ taký, že $L(A) = R$. Definujeme a-prekladač $M = (K_A \cup \{q_{zahod}\}, \Sigma_L, \Sigma_A, H, q_{zahod}, F_A)$ kde prechodová funkcia H je definovaná nasledovne:

- $(q_{zahod}, a, \varepsilon, q_{zahod}) \in H$ pre $a \in \Sigma_L$
- $(q_{zahod}, \varepsilon, \varepsilon, q_0) \in H$
- $(q, \varepsilon, a, p) \in H$ pre p, q $K_A, a \in \Sigma_A, \delta_A(q, a) = p$
- iné prechody H neobsahuje.

Tvrdíme, že $M(L) = R$. Zdôvodnenie:

$\subseteq$: Nech $w \in M(L)$. Nech štandardne $w = a_1 \dots a_n$. Potom muselo existovať slovo $u = b1 \dots b_m \in L$ a nasledovný výpočet a-prekladača M: $(q_{zahod}, b_1, \varepsilon, q_{zahod}) \dots (q_{zahod}, b_m, \varepsilon, q_{zahod})$ $(q_{zahod}, \varepsilon, \varepsilon, q_0)(q_0, \varepsilon, a_1, p_1) \dots (p_{n-1}, \varepsilon, a_n, p_n)$, kde $p_n \in F_A$. Na základe toho, ako je definovaná funkcia H a ako súvisí s prechodovou funkciou δ_A vidno, že existuje akceptačný výpočet DKA A - $(q_0, a_1 \dots a_n) \vdash (p_1, a_2 \dots a_n) \vdash \dots \vdash (p_n, \varepsilon)$, z čoho vidno, že $w \in R$.

$\supseteq$: Nech $w \in R$. Nech štandardne $w = a_1 \dots a_n$. Teda existuje akceptačný výpočet DKA A - $(q_0, a_1 \dots a_n) \vdash (p_1, a_2 \dots a_n) \vdash \dots \vdash (p_n, \varepsilon)$, kde $p_n \in F_A$. Z predpokladu vieme taktiež, že musí existovať nejaké $u = b1 \dots b_m \in L$. Na základe toho, ako je definovaná funkcia H a ako súvisí s prechodovou funkciou δ_A vidno, že musí existovať nasledovný výpočet a-prekladača M - $(q_{zahod}, b_1, \varepsilon, q_{zahod}) \dots (q_{zahod}, b_m, \varepsilon, q_{zahod})(q_{zahod}, \varepsilon, \varepsilon, q_0)(q_0, \varepsilon, a_1, p_1) \dots (p_{n-1}, \varepsilon, a_n, p_n)$, kde $p_n \in F_A$. Z toho vidno, že $w \in M(L)$.

□

Úloha 2. Nech L je jazyk nad Σ. Definujeme operáciu `pred`:

$$\texttt{pred}(L) = \{w \mid \exists u \in \Sigma^* : uw \in L, |u| = |w|\}.$$

a) (4b) Formálne dokážte, že trieda $\mathscr{L}_{ECS}$ je uzavretá na túto operáciu.

b) (1b) Zdôvodnite, že aj trieda jazykov rozpoznávaná deterministickými lineárne ohraničenými automatmi je uzavretá na túto operáciu. Konštrukciu popíšte slovne.

Riešenie. Za riešenie ďakujem Julke Froncovej, ktorej riešenie po miernej úprave a drobnom doplnení uvádzam.

a) Uzavretosť dokážeme konštrukciou LBA A', ktorý bude fungovať nasledovne: A' bude používať dvojposchodové symobly. Nech A je LBA pre jazyk L. Náš LBA A' dostane na vstup slovo w, nedeterministicky si tipne slovo u a napíše ho na druhú stopu. Nasledovne na uw, ktoré má teraz napísané na sebou v dvoch stopách, simuluje pôvodný LBA A. Dokážeme, že $L(A') = pred(L)$. Formálne:

$A' = (K', \Sigma', \Gamma', \delta', [tip], F')$
$K' = \{[tip], [return]\} \cup \{K \times \{[backward], [forward], 0, 1\}\}$
$\Sigma' = \Sigma$
$\Gamma' = \Gamma \times \Gamma$
$F' = F \times \{0, 1\}$

$$\forall a \in \Sigma, \forall b \in \Sigma : \delta'([tip], a) \ni ([tip], (b, a), 1)$$
$$\delta'([tip], \$) \ni ([return], \$, -1)$$
$$\forall a \in \Gamma' : \delta'([return], a) \ni ([return], a, -1)$$
$$\delta'([return], c) \ni ([q_0, 0], c, 1)$$
$$\forall q, p \in K; \forall a, b, d \in \Gamma; \Delta \in \{-1, 0, 1\} \delta(q, a) \ni (p, d, \Delta) \Rightarrow \delta'([q, 0], (a, b)) \ni ([p, 0], (d, b), \Delta)$$
$$\forall q, p \in K; \Delta \in \{0, 1\} : \delta(q, c) \ni (p, c, \Delta) \Rightarrow \delta'([q, 0], c) \ni ([p, 0], c, \Delta)$$
$$\forall q \in K : \delta'([q, 0], \$) \ni ([q, backward], \$, -1)$$
$$\forall q \in K; \forall a \in \Gamma' : \delta'([q, backward], a) \ni ([q, backward], a, -1)$$
$$\forall q \in K : \delta'([q, backward], c) \ni ([q, 1], c, 1)$$
$$\forall q, p \in K; \forall a, b, d \in \Gamma; \Delta \in \{-1, 0, 1\} : \delta(q, b) \ni (p, d, \Delta) \Rightarrow \delta'([q, 1], (a, b)) \ni ([p, 1], (a, d), \Delta)$$
$$\forall q, p \in K; \Delta \in \{-1, 0\} : \delta(q, \$) \ni (p, \$, \Delta) \Rightarrow \delta'([q, 1], \$) \ni ([p, 1], \$, \Delta)$$
$$\forall q \in K : \delta'([q, 1], c) \ni ([q, forward], c, 1)$$
$$\forall q \in K; \forall a \in \Gamma' : \delta'([q, forward], a) \ni ([q, forward], a, 1)$$
$$\forall q \in K : \delta'([q, forward], \$) \ni ([q, 0], \$, -1)$$

$\subseteq$: Na vstup A' príde slovo w. Náš automat ho celé prejde a pritom nedeterministicky rozhodne u, ktoré zapíše na pásku tak, že vytvorí dvojité symboly, kde na prvom mieste je

tipnuté písmenko a na druhom je písmenko z pôvodného slova, ktoré práve prešiel. Následne prejde na začiatok tejto pásky a nastaví sa na počiatočný stav automatu A a simuluje ho tak, že v stave si pamätá, s ktorým symbolom z dvojice pracovných symbolov, ktoré práve číta, pracuje. Keď robí s prvým a narazí na koniec pásky, prejde na začiatok s tým, že si pamätá stav, v ktorom skončil a začne pracovať s druhými symbolmi dvojice. Takisto, ak pri práci s druhými symbolmi narazí na začiatok pásky, presunie sa na koniec a prepne sa na prácu s prvými symbolmi. Akceptuje, ak sa na dostane do stavu $[q, i], q \in F, i \in \{0, 1\}$. Ak akceptoval, znamená to, že pôvodný automat A akceptuje slovo uw a teda platí $w \in pred(L)$. $\supseteq$: Majme slovo $w \in pred(L)$. To znamená, že existuje u také, že $uw \in L$. Vieme, že náš automat A' na slove w funguje tak, že si vie nedeterministicky toto slovo u tipnúť a potom na slove uw simulovať automat A. Vieme tiež, že aceptuje práve vtedy, kedy by automat A akceptoval uw, čo sme chceli dosiahnuť a teda platí $w \in L(A')$.

b) V prípade deterministických LBA nemôžeme slovo u nedeterministicky tipovať. To, čo nám pomôže je, že budeme v lexikografickom poradí postupne generovať všetky potenciálne slová u a po každom vygenerovaní budeme simulovať prácu stroja A na uw, pričom ak akceptuje, akceptujeme, ak neakceptuje, vygenerujeme ďalšie potenciálne u v poradí a opakujeme simuláciu. Taktiež si musíme uvedomiť, že LBA A sa môže počas simulácie zacykliť. To ale vieme ošetriť jednoducho tak, že budeme počítať koľko krokov spraví a ak ich spraví príliš veľa, tak simuláciu skončíme, lebo vieme, že už bude cykliť do nekonečna. To, koľko je príliš veľa nechávame na rozmyslenie čitateľa. Teda deterministický LBA A', ktorý pre daný deterministický LBA A akceptuje jazyk $pred(A)$ bude fungovať nasledovne. Bude potrebovať 5-stopové symobli. Na prvej stope bude počas celého výpočtu zapamätané slovo w. Druhá stopa sa bude používať na lexikografické generovanie všetkých potenciálnych slov u. Na tretiu a štvrtú stopu sa vždy po vygenerovaní nasledujúceho u skopíruje u a w a bude prebiehať simulácia stroja A na uw analogicky ako v nedeterministickom prípade. Piata páska sa bude používať na počítanie krokov simulácie, aby sme predišli zacykleniu sa.

□