Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Prídavná informácia a zložitosť nedeterministických konečných automatov

Zhrnutie naštudovaných materiálov k diplomovke

2016

Šimon Sádovský

Obsah

1 Predošlý výskum problematiky **1**

1.1 Motivácie a definícia problému 1

1.2 Deterministické konečné automaty 2

1.3 Deterministické zásobníkové automaty 6

2 Techniky určovania dolnej hranice počtu stavov NKA **7**

2.1 Techniky oblbovacích množín 7

2.2 Technika dvojklikového hranového pokrytia 10

Kapitola 1

Predošlý výskum problematiky

V tejto kapitole sa budeme zaoberať predošlým výskumom, ktorý na našej fakulte prebiehal v minulých rokoch a na ktorý chcem mojou prácou nadviazať. Konkrétne ide o práce, ktoré študovali identický problém pre deterministické konečné automaty [1] a pre deterministické zásobníkové automaty [4].

1.1 Motivácie a definícia problému

Pred tým ako zadefinujeme skúmaný problém formálne, venujme sa chvíli motiváciám vedúcim k nášmu problému. K skúmanému problému viedli dve úvahy a to nasledovné. Prvou je otázka užitočnosti prídavnej informácie pri rozpoznávaní jazyka. Volne povedané, ak automatu našepkám, že vstup, ktorý ide rozpoznávať patrí do nejakého poradného jazyka, viem tým zabezpečiť, že na rozpoznávanie pôvodného jazyka stačí automat menšej zložitosti? Táto úvaha sa dá prerozprávať nasledovne. Viem jazyk rozpoznávať nejakým automatom menšej zložitosti, ak pred automat umiestnim akýsi filter, ktorý spôsobí, že ak slovo patrí do poradného jazyka, tak ho pustí na vstup automatu a nechá automat rozhodnúť o jeho akceptácii a ak do poradného jazyka nepatrí, tak rovno rozhodne, že toto slovo akceptované nebude. Aby sme neboli príliš abstraktný, uvažujme, že chceme rozpoznávať jazyk $\{w \in a^*|a\ mod\ 6 = 0\}$ a chceme ho rozpoznávať deterministickým konečným automatom. Lahko vidno, že minimálny DKA pre tento jazyk má 6 stavov. Čo ak však automatu našepkám, že dĺžka vstupu je delitelná tromi? Resp. pred automat umiestnim filter, ktorý pustí iba slová dĺžky, ktorá je delitelná tromi? Vtedy mi stačí vziať DKA s dvomi stavmi. To nás vedie k nasledovnej definícii. Spomeňme ešte, že pod slovom automat teraz myslíme akýkolvek výpočtový model, nie nutne iba deterministický konečný automat, prípadne nedeterministický konečný automat.

Definícia 1. *Nech L_1 je jayzk. Nech A je automat. Jazyk akceptovaný automatom A*

s poradným jazykom L_1 je jazyk

$$L(A, L_1) = L(A) \cap L_1$$

Ako sme videli, sú prípady, keď poradný jazyk existuje. Pochopitelne, poradný jazyk pre jazyk L existuje vždy a je ním práve jazyk L. Avšak tento prípad z pochopitelných príčin nie je príliš zaujímavý. Takisto ak $L \subseteq \Sigma^*$, tak ak ako poradný jazyk zvolím Σ^*, tak som automatu príliš nepomohol.

Druhou úvahou, ktorá vedie k velmi podobnému problému je, či viem rozložiť automat rozpoznávajúci jazyk na dva, ktoré sú nejakým spôsobom jednoduchšie ako pôvodný automat, pričom prienik jazykov ktoré rozpoznávajú jednotlivé jednoduchšie automaty je pôvodný jazyk. Pre prepojenie s prvou úvahou si stačí uvedomiť, že ak sa to dá tak potom sa rozpoznávanie jazyka dá realizovať tak, že najprv dáme slovo rozpoznať prvému jednoduchšiemu automatu. Ak neakceptuje, neakceptujeme a ak akceptuje, dáme ho rozpoznať aj druhému automatu a podla jeho odpovede sa rozhodneme. Teda prvý jednoduchší automat plní funkciu filtra z prvej úvahy a navyše, definuje poradný jazyk. To nás vedie k nasledovnej definícii

Definícia 2. *Nech A je automat. Potom dva automaty A_1, A_2 také, že $L(A) = L(A_1) \cap L(A_2)$ nazveme rozklad automatu A. Ak sú automaty A_1, A_2 navyše oba jednoduchšie ako A, rozklad je netriviálny a hovoríme, že A je rozložitelný.*

A práve táto otázka, otázka existencie netriviálneho rozkladu nedeterministického konečného automatu nás bude zaujímať. Avšak, táto otázka sa dá posunúť ešte o úroveň vyššie, konkrétne na otázku rozložitelnosti jazyka.

Definícia 3. *Nech L je jazyk a A je minimálny automat rozpoznávajúci L. Potom jazyk L je rozložitelný, ak A je rozložitelný.*

V predchádzajúcom sme spomínali pojmy jednoduchší a minimálny automat. Aby boli tieto pojmy jasné, musíme samozrejme uviesť, o akú mieru zložitosti automatu ide, vzhladom na ktorú môže byť nejaký automat pre jazyk minimálny respektíve jeden automat jednoduchší/zložitejší ako iný. V našej práci sa ako prirodzená miera zložitosti núka počet stavov, ale takisto aj počet prechodov nedeterministického konečného automatu. Na skúmanie druhej otázky a to otázky rozložitelnosti jazyka musíme mať nástroje, pomocou ktorých vieme k jazykom hladať ich minimálne automaty. Tejto otázke sa pre NKA venuje Kapitola 2.

1.2 Deterministické konečné automaty

Ako sme už zmienili, náš problém bol skúmaný pre deterministické konečné automaty v [1]. Tu uvedieme niektoré výsledky tejto práce. Najprv však uveďme definície, vďaka ktorým môžme nahliadnuť, ako presne bol problém pre deterministické konečné automaty definovaný.

Definícia 4. *Hovoríme, že DKA* $A' = (K', \Sigma, \delta', q'_0, F')$ *realizuje stavové správanie DKA* $A = (K, \Sigma, \delta, q_0, F)$ *ak existuje injektívne zobrazenie* $\alpha : K \to K'$ *také, že:*

(a) $(\forall a \in \Sigma)(\forall q \in K) : \delta'(\alpha(q), a) = \alpha(\delta(q, a))$

(b) $\alpha(q_0) = q'_0$

Navyše, hovoríme, že A' *realizuje stavové a akceptačné správanie DKA* A*, ak navyše platí*

(c) $(\forall q \in K) : \alpha(q)\ F' \Leftrightarrow q \in F$.

Definícia 5. *Paralelné spojenie dvoch DKA* $A_1 = (K_1, \Sigma, \delta_1, q_1, F_1)$ *a* $A_2 = (K_2, \Sigma, \delta_2, q_2, F_2)$ *je DKA* $A_1 \parallel A_2 = (K_1 \times K_2, \Sigma, \delta, (q_1, q_2), F_1 \times F_2$ *taký, že platí* $\delta((p_1, p_2), a) = (\delta_1(p_1, a), \delta_2(p_2, a))$.

Definícia 6. *Pár deterministických konečných automatov* (A_1, A_2) *je stavové správanie rozpoznávajúci rozklad (SB-rozklad, z anglického state behavior decomposition) deterministického konečného automatu* A *ak* $A_1 \parallel A_2$ *realizuje stavové správanie DKA* A*. Pár* (A_1, A_2) *je stavové a akceptačné správanie rozpoznávajúci rozklad (ASB-rozklad, z anglického acceptance and state behavior decomposition) DKA* A *ak* $A_1 \parallel A_2$ *realizuje stavové a akceptačné správanie DKA* A*. Ak oba automaty,* A_1 *a* A_2 *majú menej stavov ako* A*, rozklad nazývame netriviálny.*

Nasledujúca veta hovorí, že existencia netriviálneho ASB-rozkladu pre nejaký automat implikuje existenciu netriviálneho poradného jazyka pre tento jazyk.

Veta 1. *Nech* A *je DKA. Ak existuje netriviálny ASB-rozklad* A*, potom existuje* $L_1 \in \mathscr{R}$ *a DKA* A_2 *také, že* $L(A) = L(A_2, L_1)$ *kde navyše existuje DKA pre* L_1*, ktorý má menej stavov ako* A*. Takisto* A_2 *má menej stavov ako* A

Opačná implikácia, ako autor ukazuje, neplatí.

Teraz uvedieme nutnú a postačujúcu podmienku pre existenciu netriviálneho SB-respektíve ASB-rozkladu pre daný DKA. Najprv však potrebujeme zopár definícií:

Definícia 7. *Nech* M *je konečná množina a* $\pi = \{M_1, M_2, \ldots, M_k\}$ *je množina po dvoch disjunktných neprázdnych podmnožín množiny* M *takých, že* $M = \bigcup_{i=1}^{k} M_i$*. Potom* π *nazývame rozklad množiny* M

Pripomeňme tým, čo flákali úvodné prednášky z diskretných štruktúr, že každý rozklad množiny M definuje reláciu ekvivalencie na M a vice versa. Množiny M_i nazývame bloky alebo triedy ekvivalencie.

Definícia 8. *Rozklad π množiny stavov deterministického konečného automatu A má vlastnosť substitúcie (substitution property, S.P.), ak*

$$\forall p, q \in K : p \equiv_\pi q \Rightarrow (\forall a\Sigma : \delta(p, a) \equiv_\pi \delta(q, a))$$

Definícia 9. *Nech π_1, π_2 sú rozklady množiny M, potom:*

(a) $\pi_1.\pi_2$ *je rozklad množiny M taký, že* $a \equiv_{\pi_1.\pi_2} b \Leftrightarrow a \equiv_{\pi_1} b \wedge a \equiv_{\pi_2} b$

(b) $\pi_1 + \pi_2$ *je rozklad množiny M taký, že $a \equiv_{\pi_1+\pi_2} b$ práve vtedy, keď existuje postupnosť $a = a_0, a_1, \ldots, a_n = b$ taká, že $a_i \equiv_{\pi_1} a_{i+1} \vee a_i \equiv_{\pi_2} a_{i+1}$ pre $0 \leq i \leq n-1$*

(c) $\pi_1 \preceq \pi_2$ *platí, ak $(\forall x, y \in M) : x \equiv_{\pi_1} y \Rightarrow x \equiv_{\pi_2} y$.*

Lahko vidno, že relácia definová v predchádzajúcej definícii je čistočné usporiadanie.

Označenie 1. *Triviálne rozklady množiny $M = \{m_0, m_1, ..., m_n\}$ $\{\{m_0\}, \{m_1\}, \ldots, \{m_n\}\}$ a $\{\{m_0, m_1, ..., m_n\}\}$ budeme označovať 0 a 1.*

Definícia 10. *Hovoríme, že rozklady $\pi_1 = \{R_1, \ldots, R_k\}$ a $\pi_2 = \{S_1, \ldots, S_l\}$ množiny stavov DKA A separujú akceptačné stavy A ak existujú indexy $i_1, \ldots, i_r$ a $j_1, \ldots, j_s$ také, že platí $(R_{i_1} \cup \ldots R_{i_r}) \cap (S_{j_1}, \ldots, S_{j_s}) = F$.*

Teraz môžme uviesť jeden z najpodstatnejších výsledkov uvedených v [1] a tým je nutná a postačujúca podmienka existencia oboch typov rozkladu.

Veta 2. *Deterministický konečný automat $A = (K, \Sigma, \delta, q_0, F)$ má netriviálny SB-rozklad práve vtedy keď existujú dva netriviálne S.P. rozklady π_1, π_2 množiny stavov automatu A také, že $\pi_1.\pi_2 = 0$. Tento rozklad je navyše ASB-rozklad práve vtedy keď π_1 a π_2 separujú akceptačné stavy automatu A.*

Ďalší prístup k rozkladom, ktorý autor ponúka sa zakladá na nasledujúcej myšlienke. Ak má byť rozklad na niečo užitočný, tak výsledok výpočtu menších automatov by mal nejak vypovedať o tom, aký bol výsledok výpočtu pôvodného automatu. Môže nás zaujímať, v akom stave skončil výpočet pôvodného automatu alebo iba to, či pôvodný automat akceptoval alebo nie. Taktiež je otázkou, či potrebujeme prístup k informácii o tom, v akom stave menšie automaty skončili svoje výpočty alebo nám stačí iba vedieť či akceptovali alebo neakceptovali.

Prvá možnosť je, že chceme rozložiť DKA na dva menšie automaty tak, že z informácie o stavoch v ktorých výpočty menších automatov na danom vstupnom slove skončili budeme vedieť identifikovať stav, v ktorom skončil výpočet pôvodného automatu na tom istom vstupe. Tento rozklad je sformalizovaný nasledovne:

Definícia 11. *Pár deterministických konečných automatov* (A_1, A_2) *nazývame stav-identifikujúci rozklad (SI-rozklad) DKA* A *ak existuje zobrazenie* $\beta : K_1 \times K_2 \to K$ *taký, že platí:*

$$(\forall w \in \Sigma^*) : \beta(\overline{\delta_1}(q_1, w), \overline{\delta_2}(q_2, w)) = \overline{\delta}(q_0, w)$$

Ak navyše $|K_1| \leq |K|$ *a* $|K_2| \leq |K|$*, tento rozklad je netriviálny.*

Ďalšia možná požiadavka, ktorú na rozklad môžme mať je, že chceme aby oba menšie automaty akceptovali slovo práve vtedy keď ho akceptuje pôvodný automat. Tým pádom môžme nahradiť výpočet pôvodného automatu výpočtom týchto dvoch menších automatov. Tento prístup zachytáva nasledujúca definícia.

Definícia 12. *Pár deterministických konečných automatov* (A_1, A_2) *nazývame akceptáciu-identifikujúca dekompozícia (AI-dekompozícia) DKA* A *ak platí* $L(A) = L(A_1) \cap L(A_2)$*. Ak navyše* $|K_1| \leq |K|$ *a* $|K_2| \leq |K|$*, tento rozklad je netriviálny.*

Pomerne priamočiaro vidno, že koncept AI-dekompozície je ekvivalentný existencii netriviálneho poradného jazyka.

Tretia, najslabšia, požiadavka, ktorú môžme mať na rozklad DKA je vyžadovať, že mosí existovať spósob, ako rozhodnúť, či pôvodný automat akceptuje daný vstup založený na informácii o tom, v akých stavoch skončil výpočet v oboch menších automatoch.

Definícia 13. *Pár deterministických konečných automatov* (A_1, A_2) *nazývame slabá akceptáciu-identifikujúca dekompozícia (wAI-dekompozícia) DKA* A *ak existuje relácia* $R \subseteq K_1 \times K_2$ *taká, že platí:*

$$(\forall w \in \Sigma^*) : R(\overline{\delta_1}(q_1, w), \overline{\delta_2}(q_2, w)) \Leftrightarrow w \in L(A)$$

Ak navyše $|K_1| \leq |K|$ *a* $|K_2| \leq |K|$*, tento rozklad je netriviálny.*

Nasledujúce tvrdenia ohladom týchto typov rozkladov boli v práci dokázané.

Veta 3. *Ak* (A_1, A_2) *je SI-rozklad DKA* A*, potom je to taktiež wAI-rozklad DKA* A*. Ak* (A_1, A_2) *je AI-rozklad DKA* A*, potom je to taktiež wAI-rozklad DKA* A*.*

Veta 4. *Nech* A *je minimálny DKA, nech* (A_1, A_2) *je jeho AI-rozklad. Potom* (A_1, A_2) *je taktiež SI-rozklad DKA* A*.*

Veta 5. *Nech* A *je DKA, nech* π_1, π_2 *sú netriviálne S.P. rozklady množiny stavov* A *také, že separujú akceptačné stavy* A*. Potom existuje netriviálny AI-rozklad automatu* A*.*

Veta 6. *Nech* A *je DKA, nech* π_1, π_2 *sú netriviálne S.P. rozklady množiny stavov* A *také, že* $\pi_1 . \pi_2 \preceq \{F, K - F\}$ *. Potom existuje netriviálny wAI-rozklad automatu* A*.*

V práci sú dokázané pochopitelne viaceré tvrdenia, ktoré tu ale neuvedieme. Na záver časti o deterministických konečných automatoch spomeňme ešte, že boli takisto skúmané nerozložitelné automaty a jazyky a taktiež bol ponúknutý koncept stupňa rozložitelnosti, ktorý sa dá nahliadnuť nasledovne. Ak viem automat rozložiť na dva menšie polovičnej velkosti, pomohol som si viac ako keď som ho rozložil na dva, ktorých velkosť je len o jedna menšia ako velkosť pôvodného.

1.3 Deterministické zásobníkové automaty

V práci [4] bola skúmaná otázka užitočnosti regulárnej prídavnej informácie pri výpočte deterministických zásobníkových automatov. Pre nás je zaujímavý spôsob akým sa autor pozeral na rozložitelnosť jazyka, ktorý je možné rozpoznávať deterministickým zásobníkovým automatom.

Definícia 14. *Nech* $L, L_1 \in \mathscr{L}_{eDPDA}$. *Nech* $L_2 \in \mathscr{R}$ *a nech platí* $L = L_1 \cap L_2$. *Nech* A *resp.* A_1 *sú DPDA rozpoznávajúce jazyky* L *resp.* L_1 *a nech* A_2 *je minimálny DKA rozpoznávajúci* L_2. *Hovoríme, že deterministický bezkontextový jazyk* L *je netriviálne rozložitelný na jazyky* L_1 *a* L_2 *ak automat* A_1 *má ostro menej stavov ako* A.

Táto definícia umožnuje, aby zložitosť použitého DKA bola výrazne väčšia ako zložitosť pôvodného DPDA. Avšak tu je dôležité si uvedomiť, že DKA je istým spôsobom vždy jednoduchší ako DPDA. V práci autor ukazuje triedy dobre rozložitelných a aj nerozložitelných jazykov. My sa touto prácou viac zaoberať nebudeme, nakolko náš záujem sa týka hlavne triedy regulárnych jazykov.

Kapitola 2

Techniky určovania dolnej hranice počtu stavov NKA

V tejto kapitole uvedieme techniky, pomocou ktorých budeme schopný určovať dolné hranice pre počet stavov nedeterministického konečného automatu pre daný jazyk. Pre deterministické konečné automaty máme k dispozícii `Mihill-Nerodovú vetu`, ktorá vždy dokáže určiť tesnú spodnú hranicu pre počet stavov potrebných pre deterministický konečný automat rozpoznávajúci daný jazyk. Pri nedeterministických konečných automatoch je situácia horšia. Takúto silnú techniku nemáme k dispozícii. Avšak máme k dispozícii techniky, ktoré nám poskytujú aspoň nejaké, nie nutne tesné, dolné hranice pre počet stavov potrebných pre nedeterministický konečný automat rozpoznávajúci daný jazyk. V kapitole uvedieme tri techniky - Techniku oblbovacích množín (z anglického Fooling set technique), techniku rozšírených oblbovacích množín (z anglického Extended fooling set technique) a techniku dvojklikového hranového pokrytia (z anglického Biclique edge cover technique). Kapitola čerpá z [5], [2] a [3].

2.1 Techniky oblbovacích množín

Tieto dve pomerne jednoduché techniky využívajú fakt, že ak nedeterministickému konečnému automatu, ktorý má rozpoznávať daný jazyk povolíme príliš málo stavov, tak nutne musí popliesť nejaké dva výpočty, ktoré mal od seba rozlíšiť (a teda sa nám podarilo automat oblbnúť, od toho názov techniky). Čo to znamená málo stavov a popliesť výpočty hovoria uvedené vety.

Veta 7 (Technika oblbovacích množín)**.** *Nech $L \subseteq \Sigma^*$ je regulárny jazyk. Nech $n \in \mathbb{N}$. Nech existuje množina párov $P = \{(x_i, y_i) | 1 \leq i \leq n\}$ taká, že:*

(a) $x_i y_i \in L$ *pre* $1 \leq i \leq n$

(b) $x_i y_j \notin L$ *pre* $1 \leq i, j \leq n$ *a* $i \neq j$

Potom každý NKA rozpoznávajúci L má aspoň n stavov. Množinu P s danými vlastnosťami nazývame oblbovacia množina pre jazyk L.

Dôkaz. Sporom. Nech platia predpoklady tvrdenia a nech existuje NKA A ktorý má menej stavov ako n. Pozrime sa na výpočty automatu A na slovách x_iy_i pre $1 \leq i \leq n$. Podľa definície množiny P musí platiť $(q_{0_A}, x_iy_i) \vdash^* (p_i, y_i) \vdash^* (q_{i_F}, \varepsilon)$ kde $p_i \in K_A$ a $q_{i_F} \in F_A$. Pozrime sa teraz pozornejšie na stavy p_i. Nakolko platí, že automat A má menej stavov ako je n, musí platiť, že existujú také $k \neq l$, že $p_k = p_l$. Potom však platí, že $(q_{0_A}, x_ky_l) \vdash^* (p_l, y_l) \vdash^* (q_{i_F}, \varepsilon)$. Potom však $x_ky_l \in L$ čo je spor s definíciou množiny P. Teda A má aspoň n stavov. □

Drobnou úpravou tejto vety dostaneme silnejšie tvrdenie.

Veta 8 (Technika rozšírených oblbovacích množín)**.** *Nech $L \subseteq \Sigma^*$ je regulárny jazyk. Nech $n \in \mathbb{N}$. Nech existuje množina párov $P = \{(x_i, y_i)|1 \leq i \leq n\}$ taká, že:*

(a) $x_iy_i \in L$ *pre* $1 \leq i \leq n$

(b) $x_iy_j \notin L$ *alebo* $x_jy_i \notin L$ *pre* $1 \leq i, j \leq n$ *a* $i \neq j$

Potom každý NKA rozpoznávajúci L má aspoň n stavov. Množinu P s danými vlastnosťami nazývame rozšírená oblbovacia množina pre jazyk L.

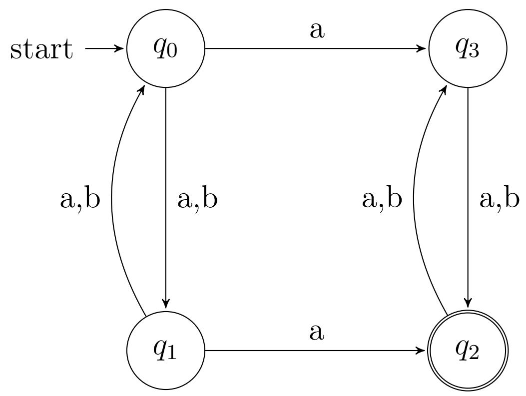
Dôkaz je takmer identický ako dôkaz pre 7 a je triviálne ho rozšíriť tak, aby dokazoval toto tvrdenie, preto ho neuvádzame. Takisto je lahko vidno, že ak je množina oblbovacou množinou pre jazyk L, je aj rozšírenou oblbovacou množinou pre L. Teraz uvedieme niekolko príkladov, aby sme ilustrovali použitie týchto techník.

Príklad 1 (Prevzatý z [3])**.** *Vezmime jazyk všetkých palindromov nad binárnou abecedou dĺžky práve k, formálne $L_{pal_k} = \{w \in \{0,1\}^k | w = w^R\}$. Vezmime množinu:*

$$P = \{(x, 0^{k-2|x|}x^R) | |x| \leq k/2\} \cup \{(x0^{k-2|x|}, x^R) | |x| \leq (k-1)/2\}$$

Použijúc techniku oblbovacích množín teraz vieme, že najmenší NKA rozpoznávajúci L_{pal_k} má aspoň $2^{\lfloor k/2 \rfloor + 1} + 2^{\lfloor (k+1)/2 \rfloor} - 2$. Táto hranica je tesná o čom sa môžme presvedčiť konštrukciou požadovaného automatu.

Prirodzená otázka, ktorá sa ponúka, je: „Ako nájsť čo najväčšiu (rozšírenú) oblbovaciu množinu pre daný jazyk L? “. Algoritmus, pomocou ktorého by sa táto množina dala skonštruovať známy nie je, avšak v [3] autori ponúkajú nasledujúcu heuristiku, ktorá, ako sa zdá, často zafunguje velmi dobre. Najprv skonštruujme NKA rozpoznávajúci jazyk L. Nech pre každý stav q tohto automatu je x_q najkratšie slovo také, že platí $(q_0, x_q) \vdash^* (q, \varepsilon)$ a nech w_q je najkratšie slovo také, že platí $(q, w_q) \vdash^* (q_F, \varepsilon)$, kde q_F je akceptačný stav. Potom zvol P ako nejakú vhodnú podmnožinu $\{(x_q, w_q)|q \in K\}$. Ilustrujme túto heuristiku na nasledujúcom príklade.



Obr. 2.1: Automat pre príklad 2

Príklad 2. *Uvažujme jazyk* $L = \{w \in \{a,b\}^* | \ |w| \ mod \ 2 = 0 \ \wedge \ w \ obsahuje \ 'a'\}$. *Intuícia našepkáva, že minimálny NKA rozpoznávajúci L by mal vyzerať ako ukazuje Obrázok 2.1. Použijme teda zmienenú heuristiku a vezmime páry slov. Pre stav* q_0 *vezmime* (ε, ab)*, pre stav* q_1 *vezmime* (b, a)*, pre stav* q_2 *vezmime* (ba, ε) *a pre stav* q_3 *vezmime* (a, b)*. Tieto 4 páry tvoria rozšírenú oblbovaciu množinu pre jazyk L a teda NKA rozpoznávajúci L má aspoň 4 stavy. Konštrukcia z Obrázku 2.1 dokazuje, že táto hranica je tesná a teda heuristika v tomto prípade perfektne zafungovala.*

Ďalšia prirodzená otázka je, či existuje jazyk, kde maximálna oblbovacia množina je menšia ako nejaká rozšírená oblbovacia množina pre tento jazyk a teda Technika rozšírených oblbovacích množín nám povie viac ako jej slabšia verzia. Že to tak je ilustruje nasledujúci príklad.

Príklad 3 (Prevzatý z [5]). *Nech* $\Sigma = \{a_i | 1 \leq i \leq n\}$*. Uvažujme konečný jazyk* $L_n = \{a_i a_j \mid 1 \leq i \leq j \leq n\}$*. Lahko vidno, že* $S_n = \{(a_i, a_i) | 1 \leq i \leq n\} \cup \{(\varepsilon, a_1 a_1), (a_1 a_1, \varepsilon)\}$ *je rozšírená oblbovacia množina pre* L_n *a má veľkosť* $n + 2$*. Potom analýza toho, že akákolvek oblbovacia množina pre* L_n *má najviac 3 prvky vyzerá nasledovne:*

(a) Najprv si všimnime, že akákolvek oblbovacia množina môže obsahovať nanajvýš jeden pár typu (ε, w) *a nanajvýš jeden pár typu* (w, ε)*.*

(b) Ďalej si všimnime, že žiadne dva páry typu (a_i, a_j) *a* (a_k, a_l) *pre* $1 \leq i \leq j \leq n$ *a* $1 \leq k \leq l \leq n$ *nemôžu byť súčasne prvkami oblbovacej množiny. BUNV predpokladajme, že* $i \leq k$*. Potom* $a_i a_l \in L_n$*, čo je v spore s definíciou oblbovacej množiny.*

Takže akákolvek oblbovacia množina pre L_n *má nanajvýš 3 prvky. Navyše hranica* $n + 2$ *pre počet stavov NKA je tesná.Čiže v tomto prípade nám dáva silnejšia technika naozaj aj lepší výsledok.*

Taktiež je nutné dodať, že sú prípady, v ktorých nám ani Technika rozšírených oblbovacích množín nefunguje ideálne. V [3] je uvedený jazyk $H_m = (\{0^m\}^+)^C$ kde neexistuje rozšírená oblbovacia množina velkosti viac ako 3. Navyše tiež ukazujú, že akýkolvek NKA pre H_m má aspoň $lg(m+1)$ stavov.

2.2 Technika dvojklikového hranového pokrytia

Poslednou technikou na dokazovanie dolných hraníc pre počet stavov potrebných pre NKA na rozpoznávanie daného jazyka ktorú uvedieme je technika dvojklikového hranového pokrytia. Najprv však potrebujeme zaviesť zopár pojmov z teórie grafov.

Definícia 15. *Bipartitný graf je trojica $G = (X, Y, E)$, kde X a Y sú (nie nutne konečné alebo diskujnktné) množiny vrcholov a $E \subseteq X \times Y$ je množina hrán.*

Definícia 16. *Bipartitný graf $H = (X', Y', E')$ nazývame podgraf grafu G ak $X' \subseteq X, Y' \subseteq Y$, a $E' \subseteq E$. Podgraf H nazveme indukovaný ak $E' = (X' \times Y') \cap E$. Ak máme danú množinu hrán E', podgraf indukovaný množinou hrán E' vzhladom na E je najmenší indukovaný podgraf obsahujúci všetky hrany z E'.*

Definícia 17. *Nech $G = (X, Y, E)$ je bipartitný graf. Množinu $C = \{H_1, H_2, ...\}$ neprázdnych bipartitných podgrafov grafu G nazveme hranové pokrytie grafu G ak každá hrana z G sa nachádza v aspoň jednom podgrafe z C.*

Definícia 18. *Bipartitný graf $G = (X, Y, E)$ nazveme dvojklika, ak platí $E = X \times Y$*

Definícia 19. *Hranové pokrytie C bipartitného grafu G nazveme dvojklikové hranové pokrytie, ak každý z podgrafov v C je dvojklika.*

Definícia 20. *Velkosť najmenšieho dvojklikového hranového pokrytia grafu G nazývame bipartitná dimenzia grafu G a označujeme ju $d(G)$. Ak pre graf neexistuje dvojklikové hranové pokrytie, jeho bipartitná dimenzia je nekonečno.*

Na záver tohto sledu definícii ešte priraďme k akémukolvek jazyku $L \subseteq \Sigma^*$ a množinám $X, Y \subseteq \Sigma^*$ bipartitný graf $G = (X, Y, E_L)$, kde $(x, y) \in E_L$ práve vtedy, keď $xy \in L$.

Teraz môžeme pristúpiť k formulácii Techniky dvojklikového hranového pokrytia.

Veta 9 (Technika dvojklikového hranového pokrytia)**.** *Nech $L \subseteq \Sigma^*$ je regulárny jazyk. Nech existuje bipartitný graf $G = (X, Y, E_L)$ kde sú $X, Y \subseteq \Sigma^*$ (nie nutne konečné množiny). Potom akýkolvek nedeterministický konečný automat rozpoznávajúci L má aspoň tolko stavov, ako je bipartitná dimenzia grafu G.*

Dôkaz. Nech $A = (K, \Sigma, \delta, q_0, F)$ je nejaký NKA rozpoznávajúci L. Ukážeme, že každý nedeterministický konečný automat rozpoznávajúci L indukuje dvojklikové hranové pokrytie grafu G. Pripomíname, že G je graf priradený k množinám X, Y a jazyku L.

Pre každý stav $q \in K$ nech $H_q = (X_q, Y_q, E_q)$, kde $X_q = X \cap \{w \in \Sigma^* | (q_0, w) \vdash^* (q, \varepsilon)\}$, $Y_q = Y \cap \{w | \exists q_F \in F : \ (q, w) \vdash^* (q_F, \varepsilon)\}$ a $E_q = X \times Y$. Teraz ukážeme, že $C = \{H_q | q \in K\}$ je dvojklikové hranové pokrytie grafu G.

(a) Priamo z definície H_q je dvojklika pre každé $q \in K$

(b) Každý bipartitný graf H_q je tiež podgrafom G. Priamo z konštrukcie H_q vyplýva $X_q \subseteq X$ a $Y_q \subseteq Y$. Ostáva ukázať, že $E_q \subseteq E_L$. Nech $x \in X_q$ a nech $yinY_q$. Z toho vyplýva, že $(q_0, x) \vdash^* (q, \varepsilon)$ a $\exists q_F \in F : \ (q, y) \vdash^* (q, \varepsilon)$. Z toho vyplýva, že $xy \in L$ a teda $(x, y) \in E_L$

(c) Ešte ostáva ukázať, že C je hranové pokrytie grafu G. Nech $(x, y) \in E_L$ je lubovolná hrana grafu G. Nájdime túto hranu v nejakom grafe z C. Z toho, ako bol graf G skonštruovaný vyplýva, že $xy \in L$. Nakolko NKA A rozpoznáva jazyk L, existuje nejaký stav $q \in K$ taký, že $(q_0, xy) \vdash^* (q, y) \vdash^* (q_F, \varepsilon)$ kde $q_F \in F$. Potom platí $x \in X_q$ a $y \in Y_q$. Nakolko H_q je dvojklika, lahko vidno, že (x, y) je hranou v E_q čo dokazuje, že C je hranové pokrytie G

Teda sme ukázali, že C je dvojklikové hranové pokrytie grafu G.

Teraz predpokladajme, že existuje nedeterministický konečný automat rozpoznávajúci L taký, že počet jeho stavov je ostro menší ako bipartitná dimenzia G. Podla predchádzajúceho však tento automat indukuje dvojklikové hranové pokrytie grafu G takej velkosti, ako je počet jeho stavov. Potom však existuje dvojklikové hranové pokrytie grafu G, ktorého velkosť je ostro menšia ako bipartitná dimenzia grafu G, čo je v spore s definíciou bipartitnej dimenzie grafu. A teda akýkolvek nedeterministický konečný automat rozpoznávajúci L má aspoň tolko stavov ako $d(G)$. Koniec dôkazu, môžete omdlieť.

$\square$

Literatúra

[1] Peter Gaži. *Parallel decomposition of finite automata.* 2006. Diplomová práca pod vedením prof. Branislava Rovana.

[2] Markus Holzer Hermann Gruber. Finding lower bounds for nondeterministic state complexity is hard. Technical report, Institut fur Informatik, Technische Universitat Munchen, Boltzmannstraße 3, D-85748 Garching bei Munchen, Germany, 2006.

[3] Jefrrey Shallit Ian Glaister. A lower bound technique for the size of nondeterministic finite automata. *Information Processing Letters*, (59):75–77, 1996.

[4] Pavel Labath. *Zjednodušenie výpočtov prídavnou informáciou.* 2010. Diplomová práca pod vedením prof. Branislava Rovana.

[5] Alexandros Palioudakis. Nondeterministic state complexity and quantifying nondeterminism in finite automata. Technical Report Technical Report 2012-596, School of Computing, Queen's University, Kingston, ON, Canada, October 2012.