JAO 2017 - Zadania z * 1

Michał Szafraniuk

26 kwietnia 2017

Zadanie 1 Rozstrzygamy, iż ten jezyk nie jest regularny. Załóżmy przeciwnie. Z lematu o pompowaniu wynika, że istnieje stała r taka, że dla dowolnego słowa w z tego języka o długości co najmniej r istnieje rozbicie $w = xyz$ o własnościach: $y \neq \epsilon, |xy| \leqslant r, xy^k z \in L$ dla dowolnego k. Rozważmy słowo $u = b^r acb^r$. Oczywiście $u \in L$, gdyż przy rozcięciu $u = b^r a \cdot cb^r$ suma wystąpień a oraz b dla lewego podsłowa równa się sumie wystąpień b i c dla prawego podsłowa (nazwijmy te sumy dla wygody *indeksami*) i wynosi $r + 1$. Rozbicie z lematu o pompowaniu daje nam $y = b^i$ dla pewnego i, przy czym $1 \leqslant i \leqslant r$. Pompujemy $k \geqslant 2$ razy dostając (przy pierwotnym rozcięciu) $b^r b^{ik} a \cdot cb^r$. Oczywiście zaburzyły się indeksy więc szukamy poprawnego rozcięcia - jedyna droga to wydłużanie prawego podsłowa kosztem lewego. Przesuńmy narazie rozcięcie o jeden w lewo: $b^r b^{ik} \cdot acb^r$. Lewy indeks wynosi teraz $r + ik$ a prawy wciąż $r + 1$. Wobec ograniczeń na i oraz k lewy indeks jest wciąż ostro większy od prawego. Przesuwamy rozcięcie j razy, za każdym przesunięciem zwiększając prawy indeks o jeden i zmniejszając lewy o jeden. Po j przesunięciach mamy $r + ik - j$ vs $r + 1 + j$ i załóżmy, że jest dobrze tj. lewy indeks równy jest prawemu. Stąd dostajemy $ik = 2j + 1$. Ponieważ k możemy dobrać dowolnie to kładąc k parzyste dostajemy sprzeczność, gdyż lewa strona równania jest parzysta a prawa nieparzysta. Każde dalsze przesuwanie rozcięcia w lewo generuje indeks lewy mniejszy od prawego. Wskazaliśmy tym samym spompowane słowo spoza języka a więc nie jest on regularny.

Zadanie 2 Niech $\mathcal{R}$ oznacza klasę języków regularnych nad wszystkimi skończonymi alfabetami. Z definicji (lub z odpowiednich twierdzeń, zależnie co jak definiujemy a co dowodzimy) języki regularne nad Σ to zbiory z rodziny $\mathcal{R}_\Sigma$, gdzie $\mathcal{R}_\Sigma$ to najmniejsza rodzina języków nad Σ, która zawiera $\emptyset$ oraz $\{a\}$ dla każdego $a \in \Sigma$ a także jest zamknięta na sumę, konkatenację oraz gwiazdkę. Chcemy rozstrzygnąć czy $\mathcal{F} = \mathcal{R}$. Rozpatrujemy obie konstrukcje $\mathcal{F}$:

- $\mathcal{F}$ zamknięta na (skończoną) sumę, dopełnienie i konkatenację

  Skądinąd wiadomo, że rodzina $\mathcal{R}_\Sigma$ jest również zamknięta na dopełnienie oraz że każdy skończony język jest regularny. A zatem $\mathcal{F} \subset \mathcal{R}$. Pytamy teraz czy $\mathcal{R} \subset \mathcal{F}$? Intuicyjnie gdyby tak było, tzn. gdyby $\mathcal{R} = \mathcal{F}$ to dostalibyśmy alternatywną definicję klasy języków regularnych i co więcej, ta definicja byłaby bardziej “kanoniczna” niż definicja wyjściowa, gdyż dopełnienie wydaje się być bardziej “kanoniczne” niż operacja gwiazdki. Zatem podejrzewamy, że $\mathcal{F}$ jest raczej właściwym podzbiorem $\mathcal{R}$.

Wstępnie macamy $\mathcal{F}$ aby dokładniej zrozumieć z czym mamy do czynienia: $\emptyset \in \mathcal{F}$ bo wystarczy przeciąć dwa rozłączne skończone języki, biorąc dopełnienie zbioru pustego dostajemy $\Sigma^* \in \mathcal{F}$. Klasę $\mathcal{F}$ możemy podzielić na podklasy języków skończonych i nieskończonych. Te pierwsze lądują w $\mathcal{F}$ z automatu ale te drugie wprowadzamy do $\mathcal{F}$ za pomocą skończonej liczby ww operacji. Intuicyjnie zastanawiam się jak wygląda indukcyjnie konstrukcja $\mathcal{F}$: najpierw pojawią się w niej skończone języki, w pewnym momencie pojawia się pierwszy język nieskończony. Zauważmy, że musi on powstać z operacji dopełnienia bo zarówno suma jak i konkatenacja na skończonych językach daje skończone języki. To "pierwsze dopełnienie" zachowuje pewną strukturę języka, którą być może kolejne operacje będą zachowywać. Szukamy konstrprzykładu więc możemy skonkretyzować rozważania: dla przykładu weźmy najprostszy alfabet $\Sigma = \{a\}$. Bierzemy dopełnienie względem $\{a\}^*$ dla jakiegoś skończonego języka i dostajemy jakiś nieokreślony początek a od pewnego momentu "spójny ogon" postaci $\{a^N, a^{N+1}, ...\}$.

Fakt 1. *Niech L będzie językiem nad $\Sigma = \{a\}$. Jeśli $L \in \mathcal{F}$ to L albo jest skończony albo ma spójny ogon.*

Dowód. Indukcja po strukturze konstrukcji L. Jeśli $L = \emptyset$ lub $L = \{a\}$ to w oczywisty sposób fakt zachodzi. Suma i konkatenacja języka skończonego / z ogonem daje język skończony / z ogonem. Dopełnienie języka skończonego / z ogonem daje język z ogonem / skończony. □

Teraz wystarczy wziąć $L_{aa} = \{aa\}^*$. Nie jest on skończony i nie ma spójnego ogona ale jest regularny. Ostatecznie zatem $\mathcal{R} \neq \mathcal{F}$.

- $\mathcal{F}$ zamknięta na sumę, dopełnienie, konkatenację i rzutowanie

  $\mathcal{R}$ jest zamknięta na rzutowanie: dla dowolnego $L \in \mathcal{R}$ bierzemy deterministyczny automat A akceptujący L i każdego przejście po symbolu, który jest usuwany przez rzutowanie zastępujemy ϵ-przejściem - czyli $\mathcal{F} \subset \mathcal{R}$ ...

Zadanie 3 Rozstrzygamy, iż taki język nie istnieje. Weźmy język regularny L i niech A będzie deterministycznym automatem skończonym takim, że $L = L(A)$. Będziemy patrzeć na A jak na graf. Jeśli L jest skończony to jego funkcja gęstości $g_L(n) = \Theta(1)$. Załóżmy zatem, że L jest nieskończony. Z lematu o pompowaniu wiemy, że dla dostatecznie długich słów z L w obliczeniach akceptujących automatu A zaczną pojawiać się cykle (przynajmniej jeden). Mamy dwie możliwości: albo cykle w A są stanowo rozłączne albo nie są.

W pierwszym przypadku graf A jest drzewem z doklejonymi rozłącznymi cyklami [1] a dowolne obliczenie to ścieżka po grafie A od stanu początkowego do jakiegoś stanu końcowego i być może po cyklach. Niech ζ reprezentuje rodzinę obliczeń z dokładnością do krotności spompowania cykli tzn.

[1] Zdaje się, że prof. Diks nazywa takie grafy kaktusami.

konkretne obliczenie z rodziny ζ to skończony ciąg zwykłych, pojedynczych tranzycji o niepowtarzających się stanach przepleciony z pewnym podciągiem spompowanych cykli rozłącznych:

$$\langle q_0, q_1, ..., q_{l_1}, C_1^{i_1}, q_{l_1+1}, ..., q_{l_k}, C_k^{i_k}, q_{l_{k+1}}, ..., q_{m+1}, q_F \rangle$$

gdzie przejścia odbywają się po jakichś konkretnych symbolach $a_1, a_2, ...$ W obliczeniu z ζ mamy $m+1$ stanów "zwykłych" czyli m tranzycji nienależących do żadnego cyklu oraz k rozłącznych cykli. Interesuje nas ile n-słów możemy wygenerować przy pomocy ζ. Oczywiście m tranzycji upycha m symboli w słowie więc do obsadzenia zostaje $n-m$ niejednoznacznie określonych pozycji. Będziemy szacować z góry więc możemy grubo przybliżać: przyjmując, że każdy niespompowany cykl dokłada tylko jeden symbol możemy ograniczyć z góry liczbę n-słów przez liczbę niemalejących funkcji ze zbioru $\{1..n-m\}$ w zbiór $\{1..k\}$ a takich funkcji jest $\binom{n-m+k-1}{k}$ co asymptotycznie daje złożoność wielomianową dla ζ. Zauważmy jednak, że skoro nasz automat jest skończony, to takich rodzin obliczeń tj. przeplecionych ciągów tranzycji niecyklowych i cykli będzie także skończona ilość co w efekcie oznacza, że funkcja gęstości dla całego języka będzie również wielomianowa.

Druga możliwość jest taka, że istnieje co najmniej jedno obliczenie akceptujące, w którym są cykle wierzchołkowo nierozłączne. Tym razem szukamy dobrego ograniczenia z dołu więc możemy redukować. Weźmy więc takie obliczenie i załóżmy, że ma dwa cykle o co najmniej jednym wspólnym wierzchołku (a jednocześnie te cykle muszą być różne gdyż wpp albo spompowaliśmy po prostu pierwszy cykl i mamy przypadek pierwszy albo wyszliśmy z pierwszego cyklu ale wtedy jeżeli na niego wróciliśmy to mamy już domknięty inny cykl). Oznacza to, że istnieje obliczenie akceptujące wyglądające następująco:

$$\langle q_0, \alpha_1, C_1, \alpha_2, C_2, \alpha_3, C_1, \alpha_4, q_F \rangle$$

gdzie α_i są niecyklowymi ciągami tranzycji. Taki układ pozwala nam pomopować nie tylko cykle ale również cykle cykli:

$$\langle q_0, \alpha_1, (C_1^{i_1}, \alpha_2, C_2^{i_2}, \alpha_3)^j, \alpha_4, q_F \rangle$$

Rozważmy teraz dowolnie długie n-słowo. Niecyklowe tranzycje zajmują jakąś stałą liczbę miejsc w n-słowie a więc zostaje wciąż $\Theta(n)$ miejsc do obsadzenia. Długości $|C_1|, |C_2|, |\alpha_2|, |\alpha_3|$ są ograniczone przez stałą np. rozmiarem pierwotnie rozważanego obliczenia. Możemy zatem przyjąć, że generujemy ciągi kul czerwonych lub czarnych o długości $\Theta(n)$. Wszystkich takich ciągów jest $2^{\Theta(n)}$ czyli w tym przypadku $g_L(n)$ jest wykładnicza.

Wniosek: dowolny język regularny ma albo funkcję gęstości wielomianową albo wykładniczą, co w sumie jest dosyć zaskakujące.