JFiZO – rozwiązanie zadania nr 40 XL

J-23

5 kwietnia 2020

Zadanie 40 XL.

Udowodnij, że dla każdego (dostatecznie dużego) n istnieje PDFA

$$\mathcal{A} = \langle \Sigma = \{0, 1\}, Q, -, -, \delta \rangle$$

taki, że $|Q| = n$ i że csync(Q) jest niepusty ale nie zawiera słowa krótszego niż $p(n)$, gdzie p jest dowolnym wcześniej ustalonym wielomianem.

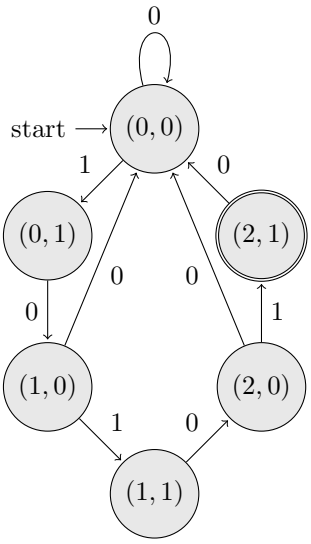
Wskazówka. Suma pierwszych n liczb pierwszych jest mniejsza niż $n^2 \log n$, zaś ich iloczyn zawsze jest większy od $2^{n \log n}$.

Wstęp

Możemy myśleć o tym zadaniu jak o problemie konstrukcji grafu z n wierzchołków, po którym trzeba chodzić „ostrożnie", tylko dozwolonymi krawędziami (co odpowiada częściowości δ). Słowo synchronizujące implementuje strategię wygrywającą ślepego żuczka który rzucony gdziekolwiek na planszę (dowolny stan) musi dojść do „mety", to znaczy jakiegoś ustalonego stanu. Naturalnie jego „ostrożność" zagwarantuje mu, że nigdy nie spadnie z grafu.

Warunek dotyczący długości słowa oznacza, że graf musi być wystarczająco skomplikowany. Konstrukcja PDFA będzie stanowiła relatywnie prostą modyfikację rozwiązania wariantu L. Tak jak wcześniej, korzystamy z zespołu mniejszych PDFA, które wymuszają, by słowo synchronizujące zawierało słowo z przecięcia języków przez nich rozpoznawanych, jednak nie mamy trzeciego znaku którym żuczek deklaruje, że osiągnął stan akceptujący dla danego automatu składowego. Ten problem obchodzimy dodając nowe stany do składowych automatów w taki sposób, by uprzednią semantykę przejść kodować nie pojedynczymi znakami, tylko parami.

Rozwiązanie będzie dwuetapowe – najpierw skonstruujemy interesujący nas PDFA (w ogólnej formie), następnie pokażemy że dla każdego wielomianu p, dla odpowiednio dużego n długość najkrótszego słowa z csync(Q) jest większa niż $p(n)$.



Rysunek 1: Graf PDFA Mp_2 („*mod* 3").

Twierdzenie 1. *Dla każdego $k \in \mathbb{N}$ istnieje ostrożnie synchronizowalny PDFA taki, że $|Q| < 1 + 2k^2 \log k$ i minimalne słowo synchronizujące jest dłuższe niż $2^{1+k \log k}$.*

Dowód. (konstrukcyjny)

Niech p_i oznacza i-tą liczbę pierwszą. Najpierw zbudujmy PDFA M_{p_i} nad alfabetem $\Sigma = \{0, 1\}$. Stanowi on modyfikację automatu akceptującego słowa 1^m gdzie $m = p_i - 1$ $(mod\ p_i)$. Po pierwsze, oprócz przejścia odpowiadającemu zwiększeniu wartości bufora $mod\ p_i$ dodajemy przejście z każdego stanu (na znaku 0) resetujące wartość bufora. Druga modyfikacja, kluczowa w obecnie rozważanym wariancie XL, polega na zduplikowaniu wszystkich stanów poprzez wprowadzenie pomiędzy nimi stanów pośrednich oraz krawędzi w taki sposób, że dla oryginalnych stanów przejścia według znaków 1 i 0 odpowiadają teraz odpowiednio słowom 10 i 00. Formalniej:

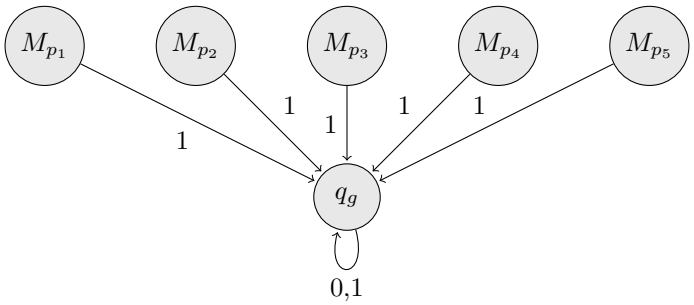
$$M_{p_i} = \langle \Sigma, Q, q_0, F, \delta \rangle$$

$$\Sigma = \{0, 1\}$$

$$Q = \{(x, j) : x \in \{0..p_i\},\ j \in \{0, 1\}\},\ q_0 = (0, 0),\ F = \{(p_i - 1, 1)\}$$

$$\delta((x, j), 0) = \begin{cases} ((x+1) mod\ p_i,\ 0) & \text{dla } j = 1 \\ (0, 0) & \text{wpp.} \end{cases}$$

$$\delta((x, j), 1) = \begin{cases} (x, 1) & \text{dla } j = 0 \\ \bot & \text{wpp.} \end{cases}$$



Rysunek 2: Schemat struktury PDFA CRT_5

Gdzie $\perp$ wyraźnie koduje przypadek nieokreślony. Ustalenie stanu akceptującego będzie miało znaczenie dopiero później. Przykładowy automat został opisany na rysunku 1.

Interesująca nas własność tak skonstruowanego PDFA polega na tym, że niezależnie od wyboru stanu q dla $m = -1 \ (mod \ p_i)$ zachodzi:

$$\hat{\delta}(q, 00(10)^m 1) = (p_i - 1, 1)$$

Własność ta natychmiast wynika z konstrukcji częściowej funkcji przejścia – krawędź 0 jest określona dla każdego stanu i prowadzi albo do wierzchołka postaci $(x, 0)$ albo do stanu początkowego, stąd słowo 00 zawsze prowadzi do q_0. Następnie m-krotne powtórzenie słowa 10 odpowiada przesunięciu się ze stanu $(0, 0)$ do stanu $(m \ mod \ p_i, 0) = (p_i - 1, 0)$. Finalny znak 1 przesuwa nas na stan akceptujący $(p_i - 1, 1)$.

Mając już rodzinę automatów M_{p_i} weźmy k pierwszych reprezentantów, to znaczy zbiór $\{M_{p_i}\}_{i=1}^{k}$. Zbudujemy z nich większy PDFA CRT_k. Konstrukcja wygląda bardzo prosto – dodajemy stan q_g z dwiema pętlami (dla obu znaków) do siebie, oraz łączymy z każdego automatu składowego jego stan akceptujący ze stanem q_g, za pomocą przejścia na znaku 1 (wcześniej nieokreślonego). Przykładowy wynik takiej konstrukcji został przedstawiony na rysunku 2.

Niech $P = \prod_{i=1}^{k} p_i$. Słowem synchronizującym (do stanu q_g) dla CRT_k jest

$$w_s = 00(10)^{P-1}11$$

By się o tym przekonać, rozważmy możliwe stany do synchronizacji:

- Stan q_g

  Jest to nasz stan docelowy, z którego nie da się w żaden sposób wyjść, zatem również po wczytaniu w_s pozostaniemy w stanie q_g.

- Stan w podautomacie M_{p_i}

  Pamiętamy kluczową własność M_{p_i}, że dla $m = -1 \ (mod \ p_i)$ słowo $00(10)^m 1$ przeprowadza ten automat z dowolnego stanu do stanu akceptującego.

Wprost z definicji P wynika $P-1=-1\ (mod\ p_i)$, zatem również w_s bez ostatniego znaku jest instancją słowa prowadzącego do stanu akceptującego. Wspomniany ostatni znak 1 powoduje przejście ze stanu akceptującego do q_g, zgodnie z konstrukcją CRT_k.

Pozostaje argumentacja, że w_s jest minimalnym słowem synchronizującym.

Załóżmy istnienie $w'_s \in csync(Q(\mathrm{CRT}_k)), |w'_s| < |w_s|$. Takie słowo koduje strategię wydostania się z dowolnego automatu składowego do stanu q_g. Aby to uczynić wymagane jest najpierw osiągnięcie stanu akceptującego. Zauważmy, że w obrębie ustalonego automatu M_{p_i} słowo 00 koduje powrót do stanu początkowego, natomiast 11 koduje „żądanie" opuszczenia automatu, zdefiniowane jedynie dla stanu $(p_i-1,0)$. Na tej podstawie można stwierdzić, że w'_s zawiera dokładnie po jednym wystąpieniu słów 00 i 11. Jest tak ponieważ po wyjściu z automatu dalsze żądania są zbędne, oraz wielokrotny powrót do stanu początkowego może zostać zamieniony na tylko jeden (ostatni), bez utraty synchronizacji. Ostrożność synchronizacji wymusza by w'_s zaczynało się znakiem 0, gdyż w przeciwnym razie przejście dla pewnych stanów będzie nieokreślone, stąd w'_s zaczyna się słowem 00. Analogicznie w'_s kończy się słowem 11, ponieważ w przeciwnym razie możliwe by było usunięcie sufiksu za wystąpieniem 11, co stoi w sprzeczności z minimalnością. Dla pewnego $n \in \mathbb{N}$ otrzymujemy zatem:

$$w'_s = 00(10)^n 11$$

gdzie w obrębie dowolnego ustalonego automatu M_{p_i}

$$\hat{\delta}((0,0),(10)^n) = (p_i-1,0)$$

Równanie to oznacza, że niezależnie od ustalonego automatu składowego

$$n = -1\ (mod\ p_i)$$

czyli spełnia układ kongruencji

$$\begin{aligned} n &= -1\ (mod\ p_1)\\ n &= -1\ (mod\ p_2)\\ &...\\ n &= -1\ (mod\ p_k) \end{aligned}$$

Na podstawie chińskiego twierdzenia o resztach wiemy, że istnieje dokładnie jedno rozwiązanie w przedziale $[0, P-1]$. Tym rozwiązaniem jest $P-1$. $|ws'| < |w_s|$ implikuje $n < P-1$, zatem otrzymujemy sprzeczność.

Na koniec obliczmy liczbę stanów CRT_k oraz długość w_s.

$$|Q(\mathrm{CRT}_k)| = 1 + \sum_{i=1}^{k} p_i < 1 + k^2 \log k$$

$$|w_s| = 4 + 2(P-1) = 2 + 2\prod_{i=1}^{k} p_i > 2^{1+k\log k}$$

□

Dysponując rodziną PDFA CRT_k o zadanych wcześniej własnościach możemy pokazać, że wzrost długości minimalnego słowa ostrożnie synchronizującego jest większy niż wielomianowy. Niech $k = \sqrt{\frac{n}{\log n}}$. Wtedy ograniczenie na liczbę stanów CRT_k ma postać

$$n > 1 + 2k^2 \log k = 1 + \frac{n}{\log n} \log(\frac{n}{\log n}) = 1 + n - \frac{n}{\log n} \log\log n$$

stąd otrzymujemy nierówność

$$n \log\log n > \log n$$

która zachodzi dla odpowiednio dużego n. Wiemy zatem, jakiej wielkości CRT_k możemy skonstruować mając dane n stanów[1]. Ustalmy zatem $p(n) = O(n^d)$. Zbadajmy teraz asymptotykę ograniczenia dolnego na długość minimalnego słowa ostrożnie synchronizującego:

$$2^{1+\sqrt{\frac{n}{\log n}} \log \sqrt{\frac{n}{\log n}}} / n^d = e^{(\log_e 2)(1+\sqrt{\frac{n}{\log n}} \log \sqrt{\frac{n}{\log n}}) - d \log n}$$

dla odpowiednio dużego n:

$$n > d^2 (\log n)^3$$

wtedy

$$e^{(\log_e 2)(1+\sqrt{\frac{n}{\log n}} \log \sqrt{\frac{n}{\log n}}) - d \log n} > 1$$

zatem dla odpowiednio dużego n długość minimalnego słowa ostrożnie synchronizującego $\mathrm{CRT}_{\sqrt{\frac{n}{\log n}}}$ jest większa od dowolnego wielomianu.

[1] By pozostać w zgodności z treścią zadania należy skomentować, że aby CRT_k miał dokładnie n stanów, możemy go powiększyć dodając odpowiednią liczbę stanów dookoła stanu q_g, tak by nie naruszyć jego synchronizowalności.