Jak dowodzimy poprawność konstrukcji automatu skończonego

Antoni Kościelski

16 marca 2018

1 Zadanie 21

1.1 Treść

Skonstruuj niedeterministyczny automat skończony rozpoznający język tych słów nad $\{0,1\}^2$, które – jako liczba w systemie dwójkowym – dzielą się przez 5, przy czym liczba jest wczytywana począwszy od najmniej znaczącego bitu.

1.2 Idea rozwiązania

Żeby rozwiązać takie zadanie trzeba coś wiedzieć o liczeniu reszt, czyli znać odpowiedni fragment arytmetyki. To pozwali wyobrazić sobie potrzebny algorytm. Algorytm liczenia reszty jest doskonale znany: dzielenie pisemne pozwala znaleźć zarówno iloraz, jak i resztę z dzielenia. Wymaga jednak czytania danej liczby od najbardziej znaczącej cyfry. Analizując ten algorytm można też zauważyć, że reszta pojawiająca się na kolejnym etapie zależy od reszty wcześniejszej, niekoniecznie jednoznacznie, i pojawia się w niewielkiej liczbie przypadków. To stwarza nadzieję na weryfikowanie, czy na końcu otrzymamy określoną resztę przez „cofanie się do tyłu", podobnie jak będziemy to robić w następnym zadaniu. Wiedzy matematyczna podpowiada, że takie cofanie jest szczególnie łatwe przy badaniu podzielności przez liczbę pierwszą.

Zauważmy, że jeżeli liczby n dopiszemy na końcu cyfrę a, to otrzymamy liczbę $2n + a$, która przestaje do $x = 2(n \bmod 5) + a$ modulo 5. Stąd, znając x możemy wyliczyć $n \bmod 5$. Uwzględniając własności reszt modulo 5 otrzymujemy, że n przystaje do $3(x - a)$ modulo 5.

1.3 Konstrukcja żądanego automatu

Oczywiście, posługuje się alfabetem $\Sigma = \{0,1\}$. Wykorzystuje pięć stanów ze zbioru $Q = \{q_0, q_1, q_2, q_3, q_4\}$. Stanem początkowym będzie q_0, podobnie q_0 jest jedynym stanem akceptującym naszego automatu. Funkcja przejścia $\delta : Q \times \Sigma \to Q$ jest zdefiniowana wzorem

$$\delta(q_i, a) = q_{3(i-a) \bmod 5}.$$

Analogiczne obliczenia wykonuje algorytm podany w kolejnym podrozdziale.

1.4 Dowód poprawności

Dowody poprawnego działania zdefiniowanego automatu i podanego niżej algorytmu są bardzo podobne. W przypadku automatu zwykle ustalamy, kiedy znajdzie

się on w poszczególnych stanach i sprawdzamy, że funkcja przejścia zachowuje te ustalenia. W przypadku algorytmu wskazujemy odpowiedni niezmiennik.

Przyjmijmy, że mamy dany ciąg cyfr W o wyrazach $W[0], W[1], \ldots, W[n-1]$ i napis $W[i..0]$ dla $i > 0$ oznacza liczbę o przedstawieniu dwójkowym $W[i-1]W[i-2]\ldots W[0]$. Tak więc po przeczytaniu i liter danego słowa automat zapoznał się z cyframi $W[0], W[1], \ldots, W[i-1]$, czyli z liczbą o przedstawieniu $W[i-1]\ldots W[1]W[0]$ oznaczaną dalej napisem $W[i..0]$. Dobrze jest przyjąć, że $W[0..0] = 0$.

Będziemy rozważać następujący algorytm:

dane: tablica W liczb 0 i 1 indeksowana od 0 do $n-1$,

stan = 0; i = n;

while (i < n) {

stan = 3(stan - W[i]) mod **5;**

i++; }

Zauważmy, że stwierdzenie

$$\text{liczba } \ 2^i \cdot \mathbf{stan} + W[i..0] \ \text{ dzieli się przez } \ 5$$

jest niezmiennikiem podanego algorytmu.

Aby się o tym przekonać, przekształćmy trochę podaną liczbą (symbole bez primów to wartości zmiennych na początku pętli, z primami – po zakończeniu wykonywania pętli):

$$2^i \cdot \mathbf{stan} + W[i..0] = 2^i \cdot (\mathbf{stan} - W[i]) + 2^i W[i] + W[i..0] = 2^i \cdot (\mathbf{stan} - W[i]) + W[i+1..0].$$

Liczba ta jest podzielna przez 5 wtedy i tylko wtedy, gdy podzielna przez 5 jest liczba

$$2^i \cdot (\mathbf{stan} - W[i]) + 5 \cdot 2^i \cdot (\mathbf{stan} - W[i]) + W[i+1..0] = 6 \cdot 2^i \cdot (\mathbf{stan} - W[i]) + W[i+1..0]$$
$$= 2^{i+1} \cdot 3 \cdot (\mathbf{stan} - W[i]) + W[i+1..0] = 2^{i'} \cdot \mathbf{stan}' + W[i'..0].$$

Oznacza to, że podane stwierdzenie jest niezmiennikiem pętli znajdującej się w algorytmie. Trzeba przekonać się, że ten niezmiennik jest prawdziwy przed wykonaniem pętli. Jeżeli tak jest, to zachochodzi też po wykonaniu pętli. Wtedy dodatkowo zachodzi równość $i = n$. Tak więc po zakończeniu algorytmu

$$\text{liczba } \ 2^n \cdot \mathbf{stan} + W[n..0] \ \text{ dzieli się przez } \ 5$$

Jeżeli po zakończeniu algorytmu zmienna **stan** ma wartość 0, to oczywiście liczba $W[n..0]$ (czyli dana liczba) jest podzielna przez 5. W przeciwnym razie liczba $2^n \cdot \mathbf{stan}$ nie dzieli się przez 5 i w konsekwencji liczba $W[n..0]$ też nie dzieli się przez 5.

W bardzo podobny sposób pokazujemy poprawność konstrukcji automatu skończonego. Dowód polega na wykazaniu, że zdefiniowany automat po przeczytaniu i znaków znajduje się w stanie q_k wtedy i tylko wtedy, gdy liczba $2^i \cdot k + W[i..0]$ jest podzielna przez 5.

2 Zadanie 22

2.1 Treść

Udowodnij, że jeśli dla pewnego języka L istnieje rozpoznajacy go NDFA, to istnieje również NDFA rozpoznający jezyk $L^R = \{w : w^R \in L\}$.

2.2 Idea

Zwykle automat skończony wyobrażamy sobie jako urządzenie działające zgodnie z pewnymi zasadami, ale możemy go również uważać za graf, którego wierzchołki odpowiadają stanom, krawędzie są etykietowane literami (są różnych rodzajów), mają określony kierunek i są wyznaczone przez dopuszczalne zmiany stanów. Patrząc na taki graf możemy sobie wyobrażać, że widzimy mapę połączeń drogowych między pewnymi miastami–stanami. Drogi na tej mapie są jednokierunkowe i są różnych rodzajów (mogą mieć różne etykiety). Słowo zbudowane z etykiet dróg możemy interpretować jako plan przejazdu. Słowo jest akceptowane, jeżeli taki plan jest możliwy do zrealizowania, a więc zgodnie z takim planem można przejechać od miasta początkowego do końcowego.

Aby sprawdzić, czy dane słowo po odwróceniu jest akceptowalnym planem można badać, czy zgodnie z danym planem–słowem można przemieścić się od stanu końcowego do początkowego jadąc w odwrotnym kierunku, pod prąd.

Dodatkowym problemem jest to, że może być wiele stanów końcowych i nie wiemy, od którego stanu mamy badać możliwość powrotu. Problem ten rozwiążemy dodając nowy stan, z którego możemy się wycofać do każdego stanu połączonego z pewnym stanem końcowym.

Głównym powodem poprawności tej idei jest następujący fakt: Jeżeli dwóch podróżnych dotarło do miast bezpośrednio połączonych (np. drogą z etykietą a), to równie dobrze mogą się spotkać w każdym z tych dwóch miast: wystarczy, aby pierwszy z nich przejechał drogą a w zwykłym kierunku, albo by drugi przejechał tą drogą „pod prąd".

2.3 Rozwiązanie

Przypuśćmy, że mamy dany język L rozpoznawny przez niedeterministyczny automat skończony $\mathcal{A}$ nad alfabetem Σ, ze zbiorem stanów Q, stanem początkowym q_0, zbiorem stanów akceptujących F i funkcją przejścia $\delta : Q \times \Sigma \to \mathcal{P}(Q)$.

Język L^R będzie rozpoznawać automat $\mathcal{A}^{\leftarrow}$ posługujący się tym samym alfabetem Σ, ze stanem początkowym $\bar{q}_0$, ze zbiorem stanów $Q \cup \{\bar{q}_0\}$ z jedynym stanem akceptującym q_0 i z funkcją przejścia $\delta^{\leftarrow}$. Funkcja przejścia jest zdefiniowana w następujący sposób:

$$\delta^{\leftarrow}(\bar{q}_0, a) = \{s \in Q : \ \delta(s, a) \cap F \neq \emptyset\}$$

oraz

$$\delta^{\leftarrow}(q, a) = \{s \in Q : \ q \in \delta(s, a)\}$$

dla $q \neq \bar{q}_0$. Teraz wystarczy dowieść, że automat $\mathcal{A}^{\leftarrow}$ rzeczywiście rozpoznaje język L^R.

2.4 Formalny opis automatu niedeterministycznego

Funkcję przejścia skończonego automatu niedeterministycznego $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ rozszerzamy do funkcji $\hat{\delta} : Q \times \Sigma^* \to \mathcal{P}(Q)$ przyjmując, że

$$\hat{\delta}(q, \varepsilon) = \{q\} \ \text{ oraz } \ \hat{\delta}(q, wa) = \bigcup_{s \in \hat{\delta}(q,w)} \delta(s, a).$$

Podobnie rozszerzamy funkcję $\delta^{\leftarrow}$.

Słowo $w \in \Sigma^*$ jest akceptowane przez automat, jeżeli wśród stanów ze zbioru $\hat{\delta}(q_0, w)$ jest stan akceptujący, czyli gdy zbiór $\hat{\delta}(q_0, w) \cap F$ jest niepusty.

2.5 Dowód poprawności

Najpierw udowodnimy sobie pomocniczy

Lemat 2.1 *Przypuśćmy, że mamy dane stany $q, q' \in Q$, słowa $u, w \in \Sigma^*$ oraz literę $a \in \Sigma$. Wtedy warunek*

$$\hat{\delta}(q, wa) \cap \hat{\delta}^{\leftarrow}(q', u) \neq \emptyset$$

jest równoważny warunkowi

$$\hat{\delta}(q, w) \cap \hat{\delta}^{\leftarrow}(q', ua) \neq \emptyset.$$

Dowód. Przypuśćmy, że mamy

1) $t \in \hat{\delta}(q, wa) \cap \hat{\delta}^{\leftarrow}(q', u)$.

2) Wtedy $t \in \delta(s, a)$ dla pewnego $s \in \hat{\delta}(q, w)$.

3) Takie s należy także do $\delta^{\leftarrow}(t, a)$ oraz do sumy $\bigcup_{t' \in \hat{\delta}^{\leftarrow}(q', u)} \delta^{\leftarrow}(t', a)$.

4) W szczególności s należy do $\hat{\delta}^{\leftarrow}(q', ua)$, a także do $\hat{\delta}(q, w)$.

5) Tak więc zbiór $\hat{\delta}(q, w) \cap \hat{\delta}^{\leftarrow}(q', ua)$ jest niepusty i kończy to dowód jednej implikacji z tezy.

Drugą z implikacji dowodzimy analogicznie. □

Wniosek 2.2 *Warunek $q' \in \hat{\delta}(q, v)$ zachodzi wtedy i tylko wtedy, gdy $q \in \hat{\delta}^{\leftarrow}(q', v^R)$.*

Dowód. Załóżmy, że $q' \in \hat{\delta}(q, v)$. Warunek ten można wyrazić także pisząc, że

$$\hat{\delta}(q, v) \cap \hat{\delta}^{\leftarrow}(q', \varepsilon) \neq \emptyset.$$

Pokażemy, że dla dowolnego przedstawienia słowa $v = wu$ zachodzi warunek

$$\hat{\delta}(q, w) \cap \hat{\delta}^{\leftarrow}(q', u^R) \neq \emptyset$$

i zrobimy to przez indukcję ze względu na długość słowa u. Dla słowa u długości 0 fakt ten wynika z założenia.

Przypuśćmy, że dla podziału $v = wu$ zachodzi powyższe stwierdzenie, oraz że $w = w'a$ dla litery a, czyli $v = (w'a)u = w'(au)$. Posługując się poprzednim lematem w przypadku $w = w'a$ otrzymujemy, że

$$\hat{\delta}(q, w') \cap \hat{\delta}^{\leftarrow}(q', u^R a) \neq \emptyset.$$

To kończy dowód indukcyjny, gdyż $u^R a = (au)^R$.

Jeżeli analizowany warunek zachodzi dla każdego podziału $v = wu$, to zachodzi także dla podziału $v = \varepsilon v$. Tak więc niepusty jest także zbiór

$$\hat{\delta}(q, \varepsilon) \cap \hat{\delta}^{\leftarrow}(q', v^R) = \{q\} \cap \hat{\delta}^{\leftarrow}(q', v^R)$$

i może do niego należeć wyłącznie q. Stąd (w dowodzie jednej z implikacji) otrzymujemy tezę, a drugą implikację dowodzimy tak samo. □

Teraz możemy już podjąć próbę wykazania, że skonstruowany automat rozpoznaje „odwrócenie" języka L. Po pierwsze, niekoniecznie dobrze działa on dla słowa pustego i języków, w których jest słowo puste. Ten problem pozostaje do rozwiązania dla zainteresowanych Czytelników.

Przypuśćmy, że słowo v należy do L. Słowo v jest więc akceptowane przez automat $\mathcal{A}$, czyli w zbiorze $\hat{\delta}(q_0, v)$ jest pewien stan akceptujący, powiedzmy stan q_a. Z wyżej sformułowanego wniosku wynika, że stan q_0 (czyli stan akceptujący automatu $\mathcal{A}^{\leftarrow}$) należy do zbioru $\hat{\delta}^{\leftarrow}(q_a, v^R)$. Gdyby stan q_a był stanem początkowym automatu $\mathcal{A}^{\leftarrow}$, to oznaczałoby to, że ten automat akceptuje słowo v^R.

Stan q_a nie może być stanem początkowym automatu $\mathcal{A}^{\leftarrow}$, gdyż może być tylko jeden stan początkowy, ale wiele stanów końcowych. Konieczne jest więc jakieś „połączenie" stanów końcowych, a jedno z możliwych rozwiązań zostało wbudowane w definicję automatu $\mathcal{A}^{\leftarrow}$. Dzięki temu można wykazać, że jeżeli $q_0 \in \hat{\delta}^{\leftarrow}(q_a, v^R)$, to także $q_0 \in \hat{\delta}^{\leftarrow}(\bar{q}_0, v^R)$.

Niepuste słowo v jest postaci wa dla pewnej litery a. Wobec tego[1]

$$q_0 \in \hat{\delta}^{\leftarrow}(q_a, v^R) = \hat{\delta}^{\leftarrow}(q_a, aw^R) = \bigcup_{s \in \delta^{\leftarrow}(q_a, a)} \hat{\delta}^{\leftarrow}(s, w^R).$$

Jest więc w zbiorze $\delta^{\leftarrow}(q_a, a)$ taki stan s, że $q_0 \in \hat{\delta}^{\leftarrow}(s, w^R)$. Nietrudno zauważyć, że na mocy definicji funkcji przejścia stan s należy także do $\delta^{\leftarrow}(\bar{q}_0, a)$. Stąd

$$q_0 \in \bigcup_{s \in \delta^{\leftarrow}(\bar{q}_0, a)} \hat{\delta}^{\leftarrow}(s, w^R) = \hat{\delta}^{\leftarrow}(\bar{q}_0, aw^R) = \hat{\delta}^{\leftarrow}(\bar{q}_0, v^R),$$

co oznacza, że automat $\mathcal{A}^{\leftarrow}$ akceptuje słowo v^R.

W ten sposób dowiedliśmy, że automat $\mathcal{A}^{\leftarrow}$ akceptuje wszystkie słowa należące do języka L^R. Rzecz jasna, należy jeszcze dowieść, że ten automat nie akceptuje słów spoza języka L^R i robi się to tymi samymi metodami.

[1] Ostatnia równość wynika stąd, że tego typu wzór jest słuszny dla każdego podziału słowa v^R, także dla podziału na pierwszą literę i resztę, i jest to konsekwencja użycia w definicji $\hat{\delta}^{\leftarrow}$ podziału na najdłuższy właściwy prefiks i ostatnią literę.