Rozwiązanie zadania 40 wersja L

Grzegorz Klocek

March 2020

1 Wprowadzenie

W zadaniu rozważamy Częściowe Deterministyczne Automaty Skończone (PDFA). PDFA różni się od DFA tym, że funkcja przejścia δ może być w nim funkcją częściową, to znaczy $\delta(q, a)$ może nie być określona dla niektórych par q, a, gdzie $q \in Q$ i $a \in \Sigma$. W rezultacie, dla niektórych słów $w \in \Sigma^*$ i stanów $q \in Q$, wartość $\hat{\delta}(q, w)$ może być nieokreślona.

Dla danego PDFA $A = \langle \Sigma, Q, q_0, F, \delta \rangle$ i zbioru $S \subseteq Q$, przez $csync(S)$ („zbiór słów ostrożnie synchronizujących S") oznaczmy zbiór takich słów $w \in \Sigma^*$ że dla każdego $q \in S$ wartość $\hat{\delta}(q, w)$ jest określona, oraz dla każdych dwóch stanów q, $q_0 \in S$ zachodzi $\hat{\delta}(q, w) = \hat{\delta}(q_0, w)$. Zauważmy że, definicja nie zależy od wyboru stanów q_0 i F a tylko od zbioru stanów Q, od alfabetu Σ i od funkcji przejścia δ.

2 Zadanie

Udowodnij, że dla każdego (dostatecznie dużego) n istnieje PDFA $A = \langle \Sigma, Q, q_0, F, \delta \rangle$, taki że $csync(Q)$ jest niepusty ale nie zawiera słowa krótszego niż $p(n)$, gdzie p jest dowolnym, ustalonym wcześniej, wielomianem. Zakładamy, że $\Sigma = \{0, 1, 2\}$.

3 Rozwiązanie

Dowód pokażemy przez konstrukcje takiego częściowo deterministycznego automatu $\langle \Sigma, Q, q_0, F, \delta \rangle$ dla danego n.

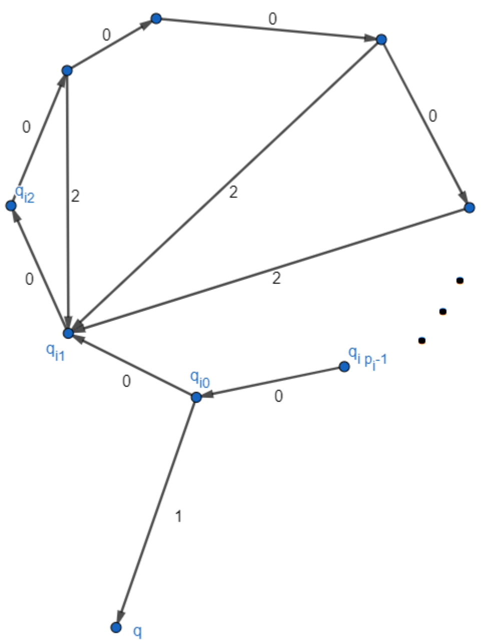
3.1 Konstrukcja automatu

Zależy nam na tym żeby najkrótsze słowo z $csync(Q)$ miało długość asymptotycznie większą niż dowolny wielomian. Nasz automat będziemy przedstawiać w postaci grafu z krawędziami skierowanymi ilustrującymi funkcje przejścia δ.

Idea naszego rozwiązania będzie następująca: graf naszego automatu będzie się składał z cykli o długościach będących różnymi liczbami pierwszymi. A konkretniej będzie to m początkowych liczb pierwszych, gdzie m jest takie że: $\sum_i^m p_i < n < \sum_i^{m+1} p_i$. Będziemy się starać, aby długość najkrótszego słowa z $csync(Q)$ była większa niż $\Pi_{i=0}^m p_i$. Wówczas na mocy nierówności zawartych we wskazówce: $n < (m+1)^2 log(m+1)$ oraz $\Pi_{i=0}^m p_i > 2^{mlogm}$ dostajemy $\Pi_{i=0}^m p_i > 2^{\sqrt{n}}$. Długość takiego słowa będzie więc większa od dowolnego wielomianu zmiennej n.

Krawędzie każdego z cykli będą oznaczone literą 0 i wszystkie będą skierowane zgodnie z ruchem wskazówek zegara. Dodatkowo nasz graf będzie posiadał jeden szczególny wierzchołek q, do którego każdy cykl będzie połączony krawędzią z literą 1. Będzie to docelowy wierzchołek, w którym znajdą się wszystkie stany po zaaplikowaniu na nich funkcji $\hat{\delta}_w$ ze słowem $w \in csync(Q)$.

Tak prezentuje się poglądowy rysunek fragmentu naszego grafu z jednym cyklem którego długość to p_i:



Każdy ze stanów $q_{i,0}, q_{i,1}, q_{i,2}...q_{i,(p_i-1)}$ jest dodatkowo połączony ze stanem $q_{i,1}$ krawędzią z literą 2. Ponadto stan q będzie połączony ze sobą krawędziami oznaczonymi literami 0,1,2 (co nie zostało zawarte na rysunku).

Łącznie mamy już $p_1 + p_2... + p_m + 1$ stanów. Dla pozostałych możemy skonstruować analogiczne cykle jednoelementowe z krawędzią 1 do q. Poza wymienionymi krawędziami, żadnych innych nasz automat nie posiada (z definicji PDFA wiemy że nie jest to wymagane). Oczywiście wybór stanu początkowego oraz stanów akceptujących nie jest istotny dla rozwiązania tego zadania.

3.2 Dowód działania automatu

Jak wcześniej wspomnieliśmy, wystarczy pokazać, że dla dowolnego słowa $w \in csync(Q)$ jego długość jest większa niż iloczyn liczb $p_1, p_2...p_m$.

rozpatrzmy najkrótsze słowo $w \in csync(Q)$. Niech $|w| = k$, oraz w_i oznacza i-tą litere w. Zdefiniujmy zbiory stanów Q_i w następujący sposób:

- $Q_0 = Q$
- $Q_i = \{\delta(q_{i-1}, w_i) : q_{i-1} \in Q_{i-1}\}$
  lub równoważnie $Q_i = \{\hat{\delta}(q_0, w_1 w_2 ... w_i) : q_0 \in Q\}$
  Oczywiście wartości $\hat{\delta}(q_0, w_1 w_2 ... w_i)$ oraz $\delta(q_j, w_i)$ są określone bo $w \in csync(Q)$

Więc Q_i to zbiór stanów, jakie otrzymalibyśmy po zaaplikowaniu funkcji przejścia δ z i-elementowym prefiksem na zbiorze Q.

Dowód wynika z następujących prostych obserwacji:

- $Q_k = \{q\}$ czyli po wczytaniu całego w dla każdego ze stanów, nasze stany synchronizują się w q. To wymusza $w_k = 1$, w przeciwnym wypadku musiało by zachodzić $Q_{k-1} = \{q\}$ co przeczy minimalności w.
- Q_{k-1} oprócz q, zawiera punkty $q_{i,0}$, czyli te elementy każdego z cykli, które są połączone z q literą 1.
- przed w_k nie może pojawić się 1. W przeciwnym wypadku przeczyło by to minimalności w, lub musielibyśmy wywołać δ_1 na stanie z którego nie wychodzi krawędź 1, co oczywiście zajść nie może.
- $w_1 = 2$ bo musimy użyć 2 w celu synchronizacji stanów na każdym cyklu oraz nie ma sensu używać wcześniej 0, ponieważ dało by to $Q_1 = Q_0$ co z kolei przeczyło by minimalności w.

- Późniejsze użycie 2 skutkowało by powrotem do Q_1, co by przeczyło minimalności w.
- Cykl o długości p_i jest zsynchronizowany w punkcie $q_{i,0}$ tylko po $p_i, 2p_i, 3p_i...$ krokach automatu (nie licząc w_1 i w_k pozostałe litery to 0).
- Wszystkie cykle będą zsynchronizowane w odpowiadającym im punktach $q_{i,0}$ (osiągniemy zbiór stanów Q_{k-1}) tylko po $NWW(p_1, p_2...p_m) = \Pi_{i=0}^{m} p_i$ krokach. Innymi słowy $k - 1 = \Pi_{i=0}^{m} p_i$.