

Języki, automaty i obliczenia

egzamin (zadania) – przykładowe rozwiązania

Zad. 1. Podaj algorytm, który dla danego wyrażenia regularnego E i liczby naturalnej n rozstrzyga, czy język definiowany przez E zawiera przynajmniej jedno słowo długości 2^n . Czas działania algorytmu powinien być wielomianowy względem $|E| + n$.

Rozwiązanie.

- Zamieniamy wyrażenie regularne na automat skończony. Bez utraty ogólności założmy, że stany automatu to $\{1 \dots m\}$.
- Obliczamy macierz binarną M rozmiaru $m \times m$, w której na pozycji (i, j) jest 1 wtw. gdy w automacie jest przejście (po jakiegokolwiek literze) ze stanu i do stanu j .
- Obliczamy macierz M^{2^n} poprzez n -krotne poniesienie do kwadratu.

Język $L(E)$ zawiera jakieś słowo długości 2^n wtw. gdy macierz M^{2^n} zawiera 1 na pozycji (i, j) , dla jakiegoś stanu początkowego i i jakiegoś stanu akceptującego j . Wszystkie kroki algorytmu są w czasie wielomianowym.

Zad. 2. Ustalmy alfabet $\{a, b\}$. Niech *ruch* oznacza wycięcie ze słowa infiksu *abba*, np. słowo *aabbabbabba* może zostać przekształcone do jednego ze słów:

$$abbabba \quad aabbbbba \quad aabbabb.$$

Niech L będzie zbiorem słów, które za pomocą pewnej liczby ruchów można przekształcić do słowa pustego.

- (a) Czy L jest regularny? Jeśli tak, to zdefiniuj automat.
- (b) Czy L jest bezkontekstowy? Jeśli tak, to zdefiniuj automat ze stosem.

Rozwiązanie.

- (a) L nie jest regularny. Dowód nie wprost. Przypuśćmy, że L jest regularny. Wtedy $K = L \cap (ab)^*(ba)^*$ też jest regularny. Zauważmy, że $K = \{(ab)^n(ba)^n : n \in \mathbb{N}\}$ – faktycznie, $(ab)^m(ba)^l \in L$ wtw. gdy $m = l$. Niech $M = h^{-1}(K)$, gdzie h jest homomorfizmem wyznaczonym przez $c \mapsto ab, d \mapsto ba$. Z regularności K wynika, że M jest regularny, ale $M = \{c^n d^n : n \in \mathbb{N}\}$ nie jest regularny. Sprzeczność.

- (b) L jest bezkontekstowy. Automat ze stosem wykonuje główną pętlę, w której wczytuje kolejne litery z wejścia i odkłada je na stos. Oprócz tego, w dowolnym momencie automat może za pomocą ε -przejścia wywołać podprocedurę, która za pomocą ε -przejść zdejmuję ze stosu kolejne litery słowa $abba$ a następnie wraca do pętli głównej. Akceptacja przez pusty stos.

Zad. 3. Czy następujące problemy są rozstrzygalne? Dla danego języka bezkontekstowego L nad alfabetem $\{a, b\}$ stwierdzić, czy:

- (a) L zawiera przynajmniej 19 różnych słów zawierających literę a ;
 (b) L zawiera wszystkie słowa zawierające literę a .

Rozwiązanie. Załóżmy bez utraty ogólności że język L jest reprezentowany przez gramatykę w postaci Greibach.

- (a) Problem rozstrzygalny. Algorytm:
- Obliczamy gramatykę dla języka $K = L \cap (a+b)^*a(a+b)^*$. Teraz należy sprawdzić, czy K zawiera przynajmniej 19 różnych słów.
 - Sprawdzamy, czy K jest nieskończony, i jeśli tak, to odpowiadamy „tak”.
 - K jest skończony, więc najdłuższe słowo w K ma długość co najwyżej wykładniczą względem rozmiaru K . Możemy więc obliczyć wszystkie słowa z K a także ich liczbę.
- (b) Problem nierozstrzygalny. Redukcja problemu uniwersalności dla języków bezkontekstowych, która przekształca język $K \subseteq (a+b)^*$ na język:

$$K' = \{wav : wv \in K, v \in b^*\}.$$

Czyli język K' zawiera dowolne słowo z K do którego wstawiono jedną literę a jako *ostatnią* a w tym słowie. Należy pokazać, że (L) $K = (a+b)^*$ wtw. gdy (P) K' zawiera wszystkie słowa zawierające a . Jeśli (L), to każde słowo zawierające a jest w K' , ponieważ po usunięciu ostatniego a słowo to należy do K . Jeśli (P), to każde słowo $w \in (a+b)^*$ należy do K , ponieważ słowo wa należy do K' .

Inne możliwe rozwiązanie to dowód nie wprost: pokażemy, że gdyby problem był rozstrzygalny, to rozstrzygalny byłby też problem uniwersalności dla języków bezkontekstowych (dla danego języka bezkontekstowego L nad $\{a, b\}$ sprawdzić, czy $K = (a+b)^*$?).

Faktycznie, oto algorytm dla uniwersalności.

- Sprawdź, czy K zawiera wszystkie słowa zawierające literę a . Teraz pozostaje nam tylko sprawdzić, czy $b^* \subseteq K$.
- Skonstruuj gramatykę dla języka $K' = K \cap b^*$, poprzez usunięcie z gramatyki wszystkich produkcji zawierających literę a ; zauważmy, że K' , będąc językiem bezkontekstowym nad alfabetem jednoliterowym $\{b\}$, jest językiem regularnym.
- Skonstruuj automat skończony dla K' .
- Sprawdź, czy $K' = b^*$ (co jest równoważne $b^* \subseteq K$).

Zad. 4. Czy dla każdego częściowo rozstrzygalnego języka L , język $\text{Cycle}(L) = \{wv : vw \in L\}$ jest częściowo rozstrzygalny?

Rozwiązanie. Tak. Niech $\mathcal{M}$ będzie maszyną rozpoznającą język L . Intuicyjnie mówiąc, maszyna dla języka $\text{Cycle}(L)$ na słowie wejściowym w uruchamia $n = |w|$ kopii maszyny $\mathcal{M}$ na wszystkich słowach otrzymanych przez przesunięcie cykliczne słowa w . Akceptacja następuje gdy którakolwiek z kopii zaakceptuje.

Organizacja takiego obliczenia nie jest trudna, ale jest żmudna. Na taśmie musimy przechowywać n konfiguracji maszyny $\mathcal{M}$:

$$\#c_1\#c_2\#\dots\#c_n\#;$$

w każdym kroku wykonujemy jeden krok (wybrany niedeterministycznie) każdej z kopii. Aby zapewnić wystarczającą ilość miejsca na taśmie do przechowywania n konfiguracji maszyny $\mathcal{M}$, przed każdym krokiem wykonujemy „rozsunięcie” konfiguracji $c_1, \dots, c_n$ na taśmie, tzn. dopisujemy symbol „blank” po lewej i po prawej stronie:

$$\#Bc_1B\#Bc_2B\#\dots\#Bc_nB\#.$$

Opisana maszyna akceptuje słowo w wtw. gdy $w \in \text{Cycle}(L)$.