Języki, Automaty, Obliczenia

Egzamin 4 lutego 2005

Zadanie 1. (1,5 pt.) Rozważamy automat wydający napoje, działający na następujących zasadach.

— Każdy napój kosztuje 1 złoty.

— W chwili początkowej automat nie zawiera żadnych monet.

— Automat przyjmuje monety: 1 złoty lub 1 euro; w tym ostatnim przypadku wydaje 3 złote reszty po warunkiem, oczywiście, że ma dostateczną ilość monet jednozłotowych (zakładamy, że €1 = 4 zł).

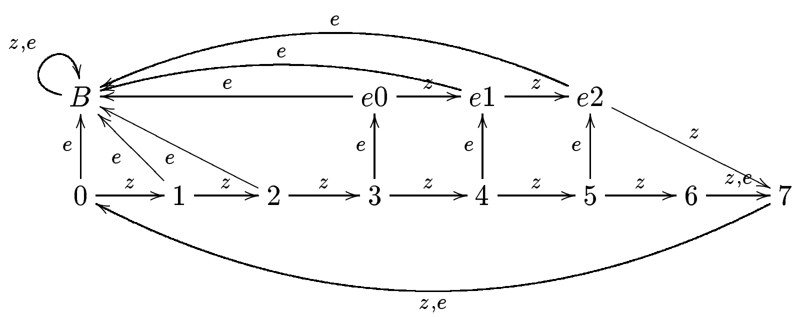
— Jeśli automat nie jest w stanie wydać reszty — sygnalizuje **błąd**.

— Jeśli po wrzuceniu monety i ewentualnym wydaniu reszty wartość wszystkich monet zgromadzonych w automacie osiągnie równowartość 8 zł, wszystkie monety są wyjmowane (*reset*).

Historią działania jest ciąg wrzuconych monet (złoty lub euro). Historia jest *udana*, jeśli w trakcie jej realizacji ani razu nie wystąpił błąd.

Skonstruować minimalny automat deterministyczny rozpoznajacy zbiór wszystkich udanych historii. Uzasadnić minimalność!

Rozwiązanie.



Wszystkie stany z wyjątkiem stanu B (błąd) są akceptujące. Stanem początkowym jest 0. Stany automatu reprezentują możliwe zawartości skarbonki automatu, z tym, że stan 7 reprezentuje również zawartość €1 + 3 zł (w obu przypadkach, po wrzuceniu którejkolwiek z monet, następuje *reset*).

Minimalność. Wykażemy, że żadnej pary stanów nie można utożsamić, ponieważ istnieje słowo akceptowane z jednego z nich, ale nie z drugiego.

Oczywiście stanu B nie można utożsamić z żadnym innym (wystarczy wziąć słowo ε). Również stan 6 łatwo odróżnia się od pozostałych, bo można z niego zaakceptować słowo ee (dwa euro wrzucone pod rząd), podczas, gdy dla każdego innego stanu $q \stackrel{ee}{\longrightarrow} B$. Wnioskujemy z tego dalej, że nie można ze sobą utożsamić żadnych stanów p i q reprezentujących dwa zbiory monet różne co do wartości. Istotnie, przypuśćmy, że p reprezentuje równowartość i złotych, a q równowartość j złotych, przy czym $0 \leq i < j < 8$. Wtedy mamy

$$q \stackrel{z^{8-j}}{\longrightarrow} 0 \stackrel{z^6}{\longrightarrow} 6$$

podczas, gdy

$$p \stackrel{z^{8-j+6}}{\longrightarrow} r \neq 6$$

a zatem słowo $z^{14-j}ee$ jest akceptowane z q, ale nie z p.

Wreszcie, nie możemy utożsamić ze sobą dwóch stanów reprezentujących tę samą wartość, tzn. stanów ei oraz $4+i$, dla $i = 0, 1, 2$, bo z $4+i$ można zaakceptować e, a z ei nie.

Uwaga. Minimalność automatu można wykazać na wiele innych sposobów, na przykład wskazując 12 słów nierównoważnych w sensie relacji $\sim_L$ z Twierdzenia Myhilla–Nerode'a lub wykazując, że algorytm minimalizacji Moore'a produkuje ten sam automat (w istocie zatrzymuje się przy pierwszej iteracji). Przy tej ostatniej metodzie możemy również zaczynać od innego automatu, np. automatu o 13 stanach, gdzie stany 7 i *e3* nie są jeszcze utożsamione.

Zadanie 2. (2 pt.) Niech L będzie zbiorem wszystkich słow postaci $w\,w^R\,w$, gdzie w jest pewnym słowem nad alfabetem $\Sigma = \{a, b\}$ o długości parzystej[1].

1. Dowieść, że dopelnienie L jest językiem bezkontekstowym.

2. Czy język L jest bezkontekstowy ?

Rozwiązanie.

Ad (a). Zajmiemy się słowami o długości podzielnej przez 6. Takie słowo $v = xyz$, gdzie $|x| = |y| = |z|$, *nie* jest w języku L wtedy i tylko wtedy gdy, $y \neq x^R$ lub $z \neq y^R$. Rozważymy najpierw ten pierwszy przypadek. Wtedy, dla pewnego i, i-ta litera x jest różna od i-tej od końca litery y, a zatem (dla $j = |x| - i$)

$$v \in \Sigma^{i-1}a\Sigma^j\,\Sigma^j b\Sigma^{i-1}\Sigma^{i+j} \;\cup\; \Sigma^{i-1}b\Sigma^j\,\Sigma^j a\Sigma^{i-1}\Sigma^{i+j}$$

Jeśli pierwszy człon tej sumy przedstawimy $\Sigma^{i-1}a\Sigma^{2j}\,b\Sigma^j\,\Sigma^{1+2(i-1)}$, nietrudno jest widzieć, że generuje go gramatyka

$$\begin{array}{lcl} X & \to & S\,X\,SS \mid aYS \\ Y & \to & SSYS \mid b \\ S & \to & a \mid b \end{array}$$

Gramatyka dla drugiego członu sumy jest analogiczna. W ten sposób dowiedliśmy, że zbiór słów xyz, gdzie $|x| = |y| = |z|$ i $y \neq x^R$ jest bezkontekstowy. Dla drugiego przypadku, tj. dla zbioru słów, gdzie $z \neq y^R$, gramatykę można napisać podobnie, ale można też zauważyć, że zbiór ten jest lustrzanym odbiciem pierwszego, a zatem jest także bezkontekstowy.

A zatem $\bar{L}$ da się przedstawić jako suma tych dwóch języków oraz języka regularnego słów o długości niepodzielnej przez 6, a więc jest bezkontekstowy.

Uwaga. Można także skonstruować niedeterministyczny automat ze stosem rozpoznający język $\bar{L}$. Tu również wygodnie jest zbudować go jako sumę automatów. Opiszemy automat rozpoznający język $\Sigma^{i-1}a\Sigma^{2j}\,b\Sigma^{j+1+2(i-1)}$.

Automat w pierwszej fazie (być może o długości 0), czytając kolejne symbole słowa wejściowego, wkłada na stos po dwa symbole $\Box$:

[1] Założenie o parzystości długości w nie jest istotne dla rozwiązania.

$$
\begin{aligned}
q, \sigma, Z_I &\to q, \Box\Box Z_I \\
q, \sigma, \Box &\to q, \Box\Box\Box
\end{aligned}
$$

(gdzie $\sigma \in \Sigma$ jest dowolnym symbolem, a Z_I jest początkowym symbolem stosu). W pewnym momencie, czytając symbol a, niedeterministycznie zmienia stan i zamiast $\Box\Box$, wkłada na stos a:

$$
\begin{aligned}
q, a, \Box &\to p_0, a\Box \\
q, a, Z_I &\to p_0, aZ_I
\end{aligned}
$$

Następuję teraz faza *długości parzystej*, w czasie której automat wkłada na stos już tylko po jednym symbolu $\Box$:

$$
\begin{aligned}
p_0, \sigma, a &\to p_1, \Box a \\
p_1, \sigma, \Box &\to p_0, \Box\Box
\end{aligned}
$$

W pewnym momencie, czytając symbol b, automat niedeterministycznie decyduje skończyć drugą fazę i przejść do fazy opróżniania stosu; może też w ogóle ,,przeskoczyć" drugą fazę (tz. jej długość jest 0):

$$
\begin{aligned}
p_0, b, \Box &\to r_0, \Box \\
p_0, b, a &\to r_0, a
\end{aligned}
$$

(stos się nie zmienia). Słowo przeczytane do tej pory jest w $\Sigma^{i-1}a\Sigma^{2j}b$, na stosie jest $\Box^{2j}a\Box^{2(i-1)}$ (pamiętajmy, że stos rośnie w lewo!) i pozostaje sprawdzić, że pozostała część słowa jest długości $j+1+2(i-1)$. W tym celu, do napotkania a na stosie, automat wczytując kolejne litery, ściąga ze stosu po dwa symbole $\Box$, a od kiedy napotka a, po jednym symbolu. Akceptuje przez pusty stos:

$$
\begin{aligned}
r_0, \sigma, \Box &\to r_1, \varepsilon \\
r_1, \varepsilon, \Box &\to r_0, \varepsilon \\
r_0, \sigma, a &\to r, \varepsilon \\
r, \sigma, \Box &\to r, \varepsilon \\
r, \varepsilon, Z_I &\to r, \varepsilon
\end{aligned}
$$

Ad (b). Gdyby język L był bezkontekstowy, to językiem bezkontekstowym byłoby również jego przecięcie z językiem regularnym

$$L' = L \cap ba^*bba^*bba^*b$$

Mamy $L' = \{ba^\ell bba^\ell bba^\ell b : \ell \text{ parzyste }\}$. Istotnie, inkluzja $\supseteq$ jest oczywista, a inkluzja $\subseteq$ wynika z faktu, że jeśli w słowie $ba^{\ell_1}bba^{\ell_2}bba^{\ell_3}b$ (o długości podzielnej przez 3), $\ell_1 \neq \ell_2$ lub $\ell_2 \neq \ell_3$, to przy podziale tego słowa na trzy części równej długości, któraś z części zawiera mniej niż 2 b.

Niech M będzie stałą z lematu o pompowaniu. Rozważmy słowo $w = ba^M bba^M bba^M b$. Przy dowolnej dekompozycji[2] $w = \alpha_1\gamma_1\beta\gamma_2\alpha_2$, gdyby γ_1 lub γ_2 zawierało wystąpienie b, to już po jednokrotnym napompowaniu, słowo $\alpha_1\gamma_1^2\beta\gamma_2^2\alpha_2$ miałoby więcej niż 6 b. A zatem γ_1 i γ_2 muszą zawierać się w całości w blokach a (w dwóch różnych lub w tym samym). Wtedy pompowanie powiększy jeden lub dwa bloki a nie zmieniając trzeciego bloku i słowo oczywiście ,,wypadnie" z L'.

Zadanie 3. (1,5 pt.) Skonstruować gramatykę bezkontekstową generującą język słów o długości parzystej nad alfabetem $\{0, 1\}$, w których liczba jedynek na pozycjach parzystych jest równa liczbie jedynek na pozycjach nieparzystych.

Rozwiązanie. Jedną z możliwych gramatyk jest

$$X \to XX \mid \varepsilon \mid 1X1 \mid 0X0$$

Poprawność: Każde dwie jedynki wprowadzone przez regułę $X \to 1X1$ są rozdzielone słowem długości parzystej (bo oczywiście wszystkie słowa generowane z X mają długość parzystą), a zatem jedna jest na pozycji parzystej, a druga na nieparzystej.[3]
Pełność: Oznaczmy język określony w zadaniu przez L. Przez indukcję na długość słowa $w \in L$. Z definicji gramatyki, słowa $\varepsilon, 00$ i 11 są wyprowadzone. Jeżeli w posiada właściwy prefiks w L, tzn. $w = vu$, gdzie $v, u \neq \varepsilon$ i $v \in L$, to łatwo jest widzieć, że również $u \in L$, a zatem możemy skorzystać z założenia indukcyjnego i reguły $X \to XX$. Przypuśćmy, że $|w| \geq 4$ i w nie ma właściwego prefiksu w L. Rozważmy podział w na bloki dwuliterowe. Wtedy pierwszy i ostatni blok są wśród $01, 10$. Jednak z warunku definicyjnego języka L wynika, że liczba bloków 01 w słowie w jest taka sama jak liczba bloków 10, z czego wnioskujemy, że w zaczyna się i kończy różnymi blokami, bo w przeciwnym razie mielibyśmy właściwy prefiks w L. Zatem w zaczyna się i kończy na ten sam symbol, powiedzmy $w = SvS$, gdzie $S \in \{0, 1\}$. Zauważmy, że wtedy również $v \in L$ (wszystkie pozycje zmieniły swoją parzystość). Możemy więc zastosować regułę $X \to SXS$ i założenie indukcyjne.

Zadanie 4*. (1,5 pt.) Dla języków L_1, L_2, określamy

$$L_1 \circ L_2 = \{w : \text{ dla każdej faktoryzacji } w = xy \text{ mamy } x \in L_1 \text{ lub } y \in L_2 \}$$

Rozstrzygnąć, czy następujące klasy są zamknięte na operację $\circ$:

(a) języki regularne,

(b) języki bezkontekstowe,

(c) języki częściowo obliczalne.

Rozwiązanie.

[2] Stosując lemat o pompowaniu, wystaczy rozważać dekompozycje $\alpha_1\gamma_1\beta\gamma_2\alpha_2$, w których $|\gamma_1\beta\gamma_2| \leq M$, ale tu nie ma to znaczenia.

[3] Bardziej dokładnie, można zauważyć, że własność, iż *liczba jedynek na pozycjach parzystych jest równa liczbie jedynek na pozycjach nieparzystych* jest niezmiennikiem każdego wyprowadzenia $X \to \alpha_1 \to \alpha_2 \to \ldots$, gdzie pozycje w słowie α_i liczymy z pominięciem symboli nieterminalnych (np. w słowie $X10XX01$ jedynki są na pozycjach 1 i 4).

Ad (a). Tak. Ustalmy alfabet Σ, taki że $L_1, L_2 \subseteq \Sigma^*$ i niech $\overline{X} = \Sigma^* - X$. Wtedy dopełnienie

$$\overline{L_1 \circ L_2} \quad = \quad \overline{L_1} \quad \overline{L_2}$$

jest językiem regularnym, a zatem $L_1 \circ L_2$ jest również językiem regularnym.

Ad (b). Nie. Na przykład

$$\{a^n b^n c^n : n \in \mathbb{N}\} \quad = \quad L_1 \circ L_2$$

gdzie $L_1 = \{a^n b^n c^m : m, n \in \mathbb{N}\} \cup a^+ b^+ \cup a^+$ i $L_2 = \{a^m b^n c^n : m, n \in \mathbb{N}\} \cup b^+ c^+ \cup c^+$.

Ad (c). Tak. Niech M_1 i M_2 będą algorytmami (częściowymi) dla L_1 i L_2 odpowiednio (tzn. $M_i(x) = \mathit{true}$, o ile $x \in L_i$, ale M_i może się nie zatrzymywać gdy $x \notin L_i$). Rozważmy algorytm

$\mathit{input}\ (w)$ (* $w = w_1 \ldots w_n$ *)
$\mathit{check} := \mathit{true}$;
for $k = 0, 1, \ldots, n$
$\mathit{check} := \mathit{check} \wedge M_1(w_1 \ldots w_k) \wedge M_2(w_{k+1} \ldots w_n)$.

Łatwo jest widzieć, że ten algorytm zatrzymuje się z wynikiem $\mathit{check} = \mathit{true}$ wtedy i tylko wtedy, gdy $w \in L_1 \circ L_2$.