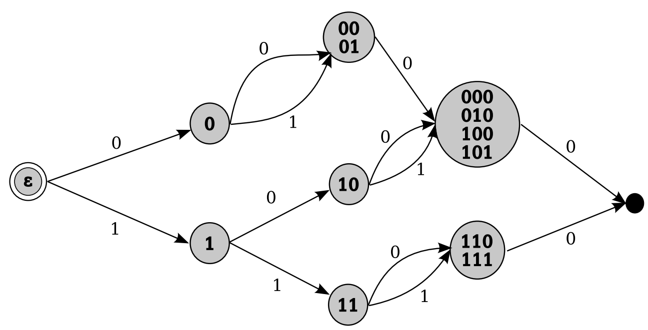
JAO: Języki, automaty i obliczenia, egzamin 18 Czerwca 2014

(każdą odpowiedź należy uzasadnić, nie dotyczy to zadania 1)

1. Narysować minimalny automat deterministyczny dla następującego języka, pomijając stan "śmietnik".

$$\{x_1x_2x_3x_4 \in \{0,1\}^4 : (x_3 \Rightarrow x_1) \wedge (x_4 \Rightarrow x_1) \wedge (x_4 \Rightarrow x_2)\}$$

Rozwiązanie Stany automatu przedstawiają klasy abstrakcji kongruencji wyznaczonej przez powyższy język.



2. Dla skończonego automatu niedeterministycznego A niech $\tilde{L}(A)$ będzie zbiorem słów, które automat akceptuje niejednoznacznie (istnieje więcej niż jedno obliczenie akceptujące). Czy $\tilde{L}(A)$ jest zawsze regularny ?

Rozwiązanie Tak, w ten sposób określony język jest zawsze regularny. Dla automatu $A = (\Sigma, Q, \delta, Q_0, F)$ pokażemy automat $\tilde{A} = (\Sigma, Q', \delta', Q'_0, F')$, rozpoznający język $\tilde{L}(A)$. Przyjmiemy

- $Q' = Q \times Q \times \{0,1\}$
- $\delta' = \{(q_1, q_2, i) \xrightarrow{a} (q'_1, q'_2, 1) \mid q_1 \xrightarrow{a} q'_1, q_2 \xrightarrow{a} q'_2, q'_1 \neq q'_2\} \cup$
  $\{(q_1, q_2, i) \xrightarrow{a} (q', q', i) \mid q_1 \xrightarrow{a} q', q_2 \xrightarrow{a} q'\}$
- $Q'_0 = Q_0 \times Q_0 \times \{0\}$
- $F' = F \times F \times \{1\}$

Jeśli automat $\tilde{A}$ akceptuje słowo w, to na pierwszych dwóch współrzędnych musi się pojawić stan końcowy A, zaś na trzeciej współrzędnej musi się pojawić wartość 1. Automat $\tilde{A}$ symuluje na pierwszych dwóch współrzędnych wykonanie A, więc uzyskanie tych stanów jest możliwe tylko w wyniku powstania dwóch symulacji przebiegu A. Pojawienie się 1 na trzeciej współrzędnej jest tylko możliwe, o ile w którymś kroku $\tilde{A}$ stany w dwóch pierwszych współrzędnych będą różne. W ten sposób uzyskujemy dwa różne przebiegi A dla słowa w. Z kolei, jeśli dwa różne przebiegi istnieją, to można je zasymulować na pierwszych dwóch współrzędnych, pierwszą różnicę zaznaczyć przejściem do 1 na trzeciej współrzędnej i w ten sposób dotrzeć do stanu akceptującego.

3. Czy następujący język jest bezkontekstowy. Jeśli tak to podać gramatykę bezkontekstową generującą ten język:

$$\{\, a^i b^{i+j} c^j \;:\; i, j \geqslant 0 \,\}$$

Rozwiązanie Powyższy język jest bezkontekstowy i jest generowany przez gramatykę

$$S \to PL, \quad P \to aPb, \quad P \to \varepsilon, \quad L \to bLc, \quad L \to \varepsilon \ .$$

Produkcja P generuje słowa postaci $a^i b^i$ dla $i \geqslant 0$. Podobnie produkcja L generuje słowa postaci $b^j c^j$ dla $j \geqslant 0$. Ich konkatenacja realizowana w produkcji S daje żądany język.

4. Czy następujący język jest bezkontekstowy. Jeśli tak to podać gramatykę bezkontekstową generującą ten język:

$$\{ a^i b^{i+j} c^j \ : \ i, j \geqslant 0 \ \& \ i \leqslant j \}$$

Rozwiązanie Ten język nie jest bezkontekstowy. Skorzystamy z lematu Ogdena. Dowód przez sprzeczność. Załóżmy, że ten język, nazwijmy go L, jest bezkontekstowy. Niech C będzie zależną od języka liczbą z lematu Ogdena. Rozważmy słowo $a^N b^{2N} c^N$, gdzie $N > C$ i umieśćmy wszystkie C znaczników w ciągu liter a. Niech $uvwxy$ będzie dekompozycją tego słowa wynikającą z lematu Ogdena. Jeśli v i x mają zawierać przynajmniej jedną zaznaczoną pozycję, to przynajmniej jedna z tych części musi zawierać litery a. Jednocześnie granica między ciągiem liter a i b nie może przebiegać ani we wnętrzu słowa v, ani we wnętrzu słowa x, bo inaczej wynik pompowania $uvvwxxy$ będzie zawierał więcej niż jedno przejście od liter a do liter b, a więc nie będzie należał do L. Analogicznie sprawa ma się dla granicy między b i c. Fragment vwx nie może cały znaleźć się w zakresie ciągu liter a, b ani c, gdyż wtedy przy pompowaniu zmieniłaby się liczba liter odpowiednio a, b i c, ale nie zmieniłaby się odpowiednio liczba liter b, a i b, więc słowo wynikowe nie mogłoby należeć do L. Pozostaje sytuacja, gdy v znajduje się wewnątrz ciągu liter a, zaś x wewnątrz ciągu liter b lub wewnątrz ciągu liter c. W tej sytuacji jednak słowo postaci $uvvwxxy$, o ile będzie postaci $a^i b^{i+j} c^j$, będzie w istocie postaci $a^i b^{i+N} c^N$ lub postaci $a^i b^{2N} c^j$. W pierwszym przypadku jednak $i > N$, co stoi w sprzeczności z założeniem, iż $i \leqslant j$, zaś w drugim suma $i + j$ nie będzie równa $2N$, zatem doszliśmy do sprzeczności.

5. Czy rozstrzygalne są następujące dwa problemy? W obu problemach dana na wejściu jest deterministyczna maszyna Turinga M i słowo wejściowe w.

(a) M wykona więcej niż 2014 zmian pojedynczych symboli.

(b) M wykona więcej niż 2014 zmian symbolu na 1-szej pozycji.

Rozwiązanie **(a)** Tak, ten problem jest rozstrzygalny. Załóżmy, że M po podaniu w na wejściu ma przebieg o co więcej niż k zmianach pojedynczych symboli. Rozważmy tą część przebiegu, która kończy się na $k+1$ zmianie. Niech $v_0, \ldots, v_k$ będą słowami reprezentującymi zawartość taśmy maszyny M przed odpowiednio pierwszą, drugą,..., $k+1$-szą modyfikacją. (Przez zawartość rozumiemy tutaj pierwsze n komórek taśmy, gdzie n jest najdalszą pozycją na taśmie odwiedzaną do $k+1$-szej modyfikacji jej zawartości włącznie.) Możemy teraz w każdym z tych słów v_j z każdą pozycją i na prawo od $|w|$ związać zbiór $W_{i,j}$ stanów, jakie M przyjęła, gdy jej głowica znajdowała się w tej pozycji między j-tą a $j+1$-szą modyfikacją symbolu na taśmie. Zauważmy teraz, że jeśli istnieją pozycje $i_1 < i_2$, takie że $v_j[i_1, i_2]$ składa się z samych blanków oraz $W_{i_1,j} = W_{i_2,j}$, to istnieje obliczenie M na słowie $v_j[1, i_1 - 1]v[i_2, |v_j|]$. Niech teraz $l_1 < l_2 < \ldots < l_p$ będą wszystkimi pozycjami, na których występuje zmiana symbolu na prawo od $|w|$. Gdyby dla jakiegoś i zachodziło $l_{i+1} - l_i - 1 > 2^{C(k+1)}$, gdzie C jest rozmiarem zbioru stanów M, to istniałaby taka para pozycji i_1, i_2, że $l_i < i_1 < i_2 < l_{i+1}$ oraz $W_{i_1,j} = W_{i_2,j}$ dla wszystkich $j = 0, \ldots, k$. Można byłoby zatem całe obliczenie z $k+1$ zmianami skrócić między pozycjami i_1 i i_2. To jest jednak niemożliwe, bo M jest deterministyczna. Dzięki temu uzyskujemy ograniczenie $|w| +$

$(k+2)2^{C(k+1)}$ na wielkość taśmy potrzebnej do zrealizowania obliczenia do $k+1$ zmiany symbolu na taśmie. Liczba konfiguracji maszyny M w tak ograniczonej przestrzeni jest ograniczona. Można zatem niedeterministycznie ustawić je w ciąg i sprawdzić, czy jego początkowy fragment stanowi obliczenie z $k+1$ modyfikacjami, co daje terminującą procedurę sprawdzającą warunek z zadania.

(b) Ten problem jest nierozstrzygalny. Gdyby był rozstrzygalny moglibyśmy za jego pomocą rozwiązywać nierozstrzygalny problem stopu. Możliwe byłoby to w następujący sposób. Dla danej maszyny N i słowa v, dla których chcemy sprawdzić własność stopu, wykonalibyśmy przekształcenie sprawiające, że całe swoje obliczenie wykonywałaby ona poza pierwszą pozycją taśmy (przesunęlibyśmy obliczenie o 1 „w prawo"), następnie dodalibyśmy po przejściu do stanu końcowego N procedurę dokonującą 1025 zmian symbolu na pierwszej pozycji, zaś słowo v uzupełnilibyśmy do $\#v$, gdzie $\#$ byłby świeżym symbolem. Tak spreparowana maszyna wykona więcej niż 2014 zmian symbolu na pierwszej pozycji wtedy i tylko wtedy, gdy N zatrzyma się na v.