Zadania z gwiazdką *Termin: 8.06, g. 23:59. Rozwiązania w plikach pdf należy przesłać przez moodla (każde zadanie osobno). Dla każdego zadania są 2 punkty do podziału dla wszystkich osób, które wysłały poprawne rozwiązanie.*

Zadanie 1. *Skonstruować algorytm dla następującego problemu decyzyjnego. Dany automat niedeterministyczny $\mathcal{A}$ nad alfabetem Σ. Rozstrzygnąć, czy istnieje taka liczba $k \in \mathbb{N}$, że dla każdego słowa $w \in \Sigma^*$ automat $\mathcal{A}$ ma co najwyżej k różnych biegów akceptujących po słowie w.*

Automat z licznikiem to automat deterministyczny wyposoażony w licznik przechowujący liczby naturalne, który jest początkowo ustawiony na 0. Automat ten podejmuje swoje decyzje na podstawie aktualnie wczytywanej litery, aktualnego stanu, oraz wyniku testu na zero wykonanym na liczniku. Formalnie, automat taki ma następujące składniki:

- alfabet wejściowy A,
- zbiór stanów Q,
- stan początkowy q_0,
- stany akceptujące $F \subseteq Q$,
- funkcję przejścia $\delta \colon Q \times A \times \{0, \neq 0\} \to Q \times \mathbb{Z}$.

Ustalmy słowo wejściowe $w \in A^*$. Konfiguracja automatu to trójka (q, i, c) gdzie $q \in Q$ to stan, $0 \leqslant i \leqslant |w|$ to liczba wczytanych liter, oraz $c \in \mathbb{N}$ to wartość licznika. Początkowa konfiguracja to $(q_0, 0, 0)$, a konfiguracje końcowe to $(q, |w|, c)$, gdzie $q \in F$ oraz $c \in \mathbb{Z}$. Z konfiguracji (p, i, c), dla $i \geqslant 0$, przechodzimy do konfiguracji $(q, i+1, c+k)$ jeśli $c + k \geqslant 0$, $\delta(p, a, \neq 0) = (q, k)$ oraz $c \neq 0$ lub $\delta(p, a, 0) = (q, k)$ oraz $c = 0$.

Zadanie 2. *Pokazać, że jeśli automat z licznikiem o n stanach akceptuje jakieś słowo, to też akceptuje słowo długości co najwyżej $O(n^3)$.*

Zadanie 3. *Pokazać, że jeśli automat z licznikiem o n stanach akceptuje jakieś słowo, to też akceptuje słowo długości co najwyżej $O(n^2)$.*

Automat niedeterministyczny z licznikiem zdefiniowany jest podobnie jak powyżej, z tym że zamiast funkcji jest relacja $\delta \subseteq Q \times A \times \{0, \neq 0\} \times Q \times \mathbb{Z}$. Automat *nie ma testów na zero* jeśli trzecia współrzędna nie ma znaczenia, tzn. jeżeli jakaś piątka należy do δ, to piątka otrzymana przez zmianę trzeciej współrzędnej również.

Zadanie 4. *Czy następujący problem jest rozstrzygalny?*

- ***Dane:*** *automat niedeterministyczny z licznikiem bez testów na zero,*
- ***Rozstrzygnąć:*** *czy istnieje taka stała $C \in \mathbb{N}$, że dla każdego słowa $w \in A^*$ istnieje bieg akceptujący z końcową wartością licznika nie większą niż C?*

Zadanie 5. *Rozważamy automaty* ze stosem oraz z jednym licznikiem*, który może przechowywać liczby* całkowite *(ujemne lub nieujemne) i je zwiększać lub* *zmniejszać o jeden. Początkowo, licznik jest ustawiony na* 0*. Automat w trakcie biegu nie może sprawdzać zawartości licznika, ale może akceptować tylko wtedy, gdy licznik jest wyzerowany, obecny stan jest akceptujący, oraz automat przeczytał całe słowo wejściowe.*

Pokazać, że istnieje algorytm rozwiązujący problem pustości dla takich automatów. Dokładniej, mając dany automat ze stosem oraz jednym licznikiem, rozstrzygnąć, czy istnieje słowo akceptowane przez ten automat.

Zadanie 6. *Czy następujący problem jest rozstrzygalny: Dana gramatyka bezkontekstowa z jednym nieterminalnem i nieterminalami* Σ*; czy generuje ona* Σ^**?.*

Niech $\mathbb{F}_2$ oznacza *grupę wolną* o generatorach $\{a, b\}$. To znaczy, jest to nieskończona grupa, której elementami są klasy równoważności słów nad alfabetem $\Sigma = \{a, b, a^{-1}, b^{-1}\}$. Rozważana tu relacja równoważności to najmniejsza relacja równoważności $\sim$ taka, że $aa^{-1} \sim a^{-1}a \sim bb^{-1} \sim b^{-1}b \sim \varepsilon$ oraz dla $u, v \in \Sigma^*$, jeśli $u \sim u'$ oraz $v \sim v'$ to $uv \sim u'v'$. Przykładowo, mamy ciąg równoważności:

$$\underbrace{a^{-1}a}_{\varepsilon}baa^{-1}b^{-1} \sim b\underbrace{aa^{-1}}_{\varepsilon}b^{-1} \sim bb^{-1} \sim \varepsilon.$$

Mnożenie w $\mathbb{F}_2$ zadane jest przez konkatenację (dokładniej, $[u]_\sim \cdot [v]_\sim = [u \cdot v]_\sim$).

Zadanie 7. *Skonstruować algorytm który dostawszy (zwykły, niedeterministyczny) automat nad alfabetem* Σ *rozstrzyga, czy akceptuje on pewne słowo równoważne* ε*.*

Zadanie 8. *Ustalmy* $g_1, g_2, \ldots, g_k \in \mathbb{F}_2$*. Pokazać, że zbiór* $\{(n_1, \ldots, n_k) \in \mathbb{N}^k \mid g_1^{n_1} \cdots g_k^{n_k} \sim \varepsilon\}$ *jest semi-liniowy, tj. jest skończoną sumą zbiorów postaci* $\{v + k_1 \cdot w_1 + k_2 \cdot w_2 + \ldots + k_l \cdot w_l \mid k_1, \ldots, k_l \in \mathbb{N}\}$*, dla* $l \in \mathbb{N}$ *oraz* $v, w_1, \ldots, w_l \in \mathbb{N}^k$*.*

Zadanie 9. *Dla liczby* $N \in \mathbb{N}$*, niech* $f(N)$ *będzie najmniejszą taką liczbą* K*, że dla dowolnych dwóch różnych liczb* $m, n < N$ *istnieje automat ze stosem o co najwyżej* K *stanach, akceptujący dokładnie jedno ze słów* 1^m*,* 1^n*. Jaka jest asymptotycznie funkcja* f*? Chodzi o odpowiedź typu* $\Theta(\sqrt{N}), \Theta(\log N)$*, itp.*

Zadanie 10. *Powiemy, że gramatyka bezkontekstowa jest* jednoznaczna*, jeżeli dla każdego danego słowa wejściowego* w *ma ona co najwyżej jedno drzewo parsowania. Pokazać, że istnieje algortym który dostawszy jednoznaczną gramatykę* $\mathcal{G}$ *nad alfabetem* A *stwierdza, czy* $L(\mathcal{G}) = A^*$*.*

Zadanie 11. *Rozważmy następujący problem decyzyjny.*

- ***Dane:*** *jednotaśmowa maszyna Turinga* $\mathcal{M}$ *oraz słowo* $v \in \{0, 1\}^*$*,*
- ***Rozstrzygnąć:*** *czy* $\mathcal{M}$ *akceptuje słowo* v *w co najwyżej* $|v|^2$ *krokach?*

1. *Pokazać, że nie istnieje maszyna Turinga rozwiązująca powyższy problem, która działa w co najwyżej* $1000n^2$ *krokach na wejściu długości* n*.*
2. *Pokazać, że istnieje maszyna Turinga rozwiązująca powyższy problemjęzyk* L*, działająca w co najwyżej* n^6 *krokach dla słów długości* n*.*

Automaty czytające grafy

Rozważmy następujący model automatów na grafach. Każdy rozważany graf G ma zbiór skończony wierzchołków $V \subseteq \mathbb{N}$, jest nieskierowany i nie ma samopętli.

Automat grafowy wczytuje ciąg wierzchołków danego grafu G (być może z powtórzeniami) i może zapisać obecnie wczytywany wierzchołek w jednym ze swoich rejestrów (których jest ustalona liczba). Decyzje o kolejnych krokach automat podejmuje na podstawie tego, które z wierzchołków w jego rejestrach są połączone krawędzią, lub są równe obecnemu wierzchołkowi.

Dokładniej, automat grafowy $\mathcal{A}$ ma:

- skończony zbiór stanów Q,
- zbiór stanów początkowych $I \subseteq Q$ oraz akceptujących $F \subseteq Q$;
- skończony zbiór rejestrów R;
- relację przejścia $\delta \subseteq Q \times P(R) \times R \times Q$.

Tranzycja postaci (q, S, r, q') jest interpretowana jako instrukcja: *jeśli jesteś w stanie q oraz obecnie wczytywany wierzchołek jest sąsiadem wierzchołków w rejestrach $S \subseteq R$, to zapisz obecny wierzchołek w rejestrze r i przejdź do stanu* q. To jest sprecyzowane poniżej.

Dla grafu G oraz ciągu jego wierzchołków $v_1, \ldots, v_k$, *konfiguracja* automatu $\mathcal{A}$ ma następujące składowe:

- pozycję $1 \leqslant i \leqslant k$,
- wartościowanie rejestrów $\rho \colon R \to \{v_1, \ldots, v_k\}$,
- stan $q \in Q$.

Niech $\rho[s \mapsto v]$ oznacza waluację otrzymaną z ρ przez zamianę wartości $\rho(s)$ na v. Będąc w konfiguracji (i, ρ, q) automat może przejść do konfiguracji $(i+1, \rho[s \mapsto v_{i+1}], q')$, gdzie $i < k$ oraz $s \in R$, jeżeli $(q, S, s, q') \in \delta$, gdzie

$$S = \{s \in R \mid \rho(s) \textit{ jest sąsiadem } v_{i+1} \textit{ w } G\}.$$

Konfiguracja początkowa to $(1, \overline{v_1}, q_0)$, gdzie $\overline{v_0} \colon R \to \{v_1, \ldots, v_k\}$ to funkcja stale równa v_1. Konfiguracja akceptująca to konfiguracja ze stanem akceptującym oraz ostatnią pozycją w ciągu.

Automat $\mathcal{A}$ akceptuje graf G jeżeli istnieje enumeracja $v_1, v_2, \ldots, v_k$ jego wierzchołków (bez powtórzeń) taka, że $\mathcal{A}$ ma bieg akceptujący po $v_1, \ldots, v_k$.

Zadanie 12. *Czy istnieje algorytm, który dostawszy automat grafowy $\mathcal{A}$ rozstrzyga, czy $\mathcal{A}$ akceptuje pewien graf G?*

Zadanie 13. *Czy istnieje algorytm, który dostawszy automat grafowy $\mathcal{A}$ rozstrzyga, czy $\mathcal{A}$ akceptuje każdy graf G?*