Zadania z gwiazdką - seria II, zadania 2 i 3
szkic rozwiązań

2. Niech $\Sigma = \{a_1, \ldots, a_n\}$. Dla $S \subseteq \Sigma$ oraz słowa $w \in \Sigma^*$ niech $\#_S(w)$ oznacza liczbę liter w, które należą do zbioru S. Niech

$$L_{S_1,T_1,S_2,T_2} = \{w \in a_1^* a_2^* \cdots a_n^* \mid \#_{S_1}(w) = \#_{T_1}(w) \wedge \#_{S_2}(w) = \#_{T_2}(w)\}.$$

Oblicz liczbę czwórek (S_1, T_1, S_2, T_2) takich, że L_{S_1,T_1,S_2,T_2} jest językiem bezkontekstowym.

Szkic rozwiązania Zauważmy najpierw, że jeśli pewna litera a_i należy do $S_1 \cap T_1$, to równie dobrze można rozważać $S_1 \setminus \{a_i\}$ oraz $T_1 \setminus \{a_i\}$. Takie rozumowanie prowadzi szybko do wniosku, że jeśli zdefiniujemy: $S'_i = S_i \setminus T_i$ oraz $T'_i = T_i \setminus S_i$ dla $i \in \{1, 2\}$, to zachodzi

$$L_{S_1,T_1,S_2,T_2} = L_{S'_1,T'_1,S'_2,T'_2}$$

oraz oczywiście $S'_i \cap T'_i = \emptyset$ dla $i \in \{1, 2\}$. W dalszej części zadania zakładamy więc, że $S_i \cap T_i = \emptyset$ dla $i \in \{1, 2\}$, a dopiero przy zliczaniu czwórek przypomnimy sobie, że niekoniecznie jest to prawda dla wejściowej gramatyki.

Na początek ustalmy S_1, T_1, S_2, T_2 i oznaczmy $L = L_{S_1,T_1,S_2,T_2}$. Oznaczmy $U_1 = S_1 \cup T_1$ oraz $U_2 = S_2 \cup T_2$. Zauważmy, że jeśli jeden ze zbiorów jest pusty (załóżmy bez straty ogólności S_1), to język L jest bezkontekstowy. Istotnie, konstruujemy automat ze stosem, który jeśli napotka literę z T_1 to odrzuca słowo. Oprócz tego liczy on litery z S_2 i T_2 (w standardowy sposób, trzymając na stosie ich różnicę) i akceptuje jeśli ta różnica wyniesie na koniec 0.

W dalszej części rozwiązania zakładamy zatem, że wszystkie zbiory S_i oraz T_i są niepuste. Pokażemy teraz, że jeśli $U_1 \cap U_2 \neq \emptyset$, to język L nie jest bezkontekstowy oprócz pewnych szczególnych sytuacji. Załóżmy bez straty ogólności, że $S_1 \cap S_2 \neq \emptyset$, pozostałe przypadki są analogiczne. Niech $a_i \in S_1 \cap S_2$ oraz niech $a_j \in T_1$ i $a_k \in T_2$. Istnienie takich a_j oraz a_k wynika z niepustości T_1 i T_2. Rozważymy na razie przypadek gdy $j \neq k$. Załóżmy bez straty ogólności (argument nie używa tego założenia), że $i < j < k$. Jeśli L jest bezkontekstowy, to również $L' = L \cap a_i^* a_j^* a_k^*$ jest bezkontekstowy. Jednak $L' = \{a_i^n a_j^n a_k^n \mid n \in \mathbb{N}\}$, więc jak łatwo pokazać używając lematu o pompowaniu nie jest bezkontekstowy. A zatem również L nie jest bezkontekstowy.

Powyżej założyliśmy jednak, że $a_j \in T_1$ i $a_k \in T_2$ dla $j \neq k$. Może się zdarzyć, że nie ma takich j, k. Wówczas musi być $T_1 \subseteq T_2$ (lub odwrotnie). Jeśli $T_1 \subseteq T_2$, to istnieje pewne $a_i \in T_1 \cap T_2$. A zatem podobnie musi być $S_1 \subseteq S_2$ lub $S_2 \subseteq S_1$. Są więc dwa przypadki szczególne:

1. $S_1 \subseteq S_2$, $T_1 \subseteq T_2$, czyli $U_1 \subseteq U_2$. Wówczas $L = L_{S_1,T_1,S_2 \setminus S_1,T_2 \setminus T_1}$ i należy rozważyć ten przypadek.

2. $S_1 \subseteq S_2$, $T_2 \subseteq T_1$. Wówczas $L = L_{S_1,T_2,(S_2 \setminus S_1) \cup (T_1 \setminus T_2),\emptyset}$ i jest bezkontekstowy.

Podobnie wygląda sytuacja dla innych podprzypadków przypadku $U_1 \cap U_2 \neq \emptyset$, np. gdy $S_1 \cap T_2 \neq \emptyset$.

Załóżmy więc, że $U_1 \cap U_2 = \emptyset$. Pokażemy teraz, że jeśli istnieją $i < j < k < \ell$ takie, że $i, k \in U_1$, $j, \ell \in U_2$, bądź odwrotnie, czyli $i, k \in U_2$, $j, \ell \in U_1$, to język L nie jest bezkontekstowy. Załóżmy bez straty ogólności, że $i, k \in U_1$, $j, \ell \in U_2$. Niech $L' = L \cap a_i^* a_j^* a_k^* a_\ell^*$. Wówczas $L' = \{a_i^p a_j^r a_k^p a_\ell^r \mid p, r \in \mathbb{N}\}$ i jak łatwo pokazać używając lematu o pompowaniu L' nie jest bezkontekstowy.

Jeśli natomiast nie ma takiej czwórki indeksów i, j, k, ℓ to L jest bezkontekstowy. Wówczas, mówiąc nieformalnie, albo nie istnieje litera z U_1 otoczona literami z U_2 albo nie istnieje litera z U_2 otoczona literami z U_1. Załóżmy pierwszą opcję, czyli nie istnieje litera z U_1 otoczona literami z U_2. Wówczas istnieją pewne dwa indeksy $p < q$ takie, że $U_2 \subseteq \{a_p, a_{p+1}, \ldots, a_q\}$ oraz $U_1 \subseteq \{a_1, \ldots, a_{p-1}\} \cup \{a_{q+1}, \ldots, a_n\}$. Możemy wówczas skonstruować automat ze stosem, który najpierw liczy różnicę między liczbą liter z S_1 a liczbą liter z T_1, a potem, po napotkaniu liter a_r dla $p \leqslant r \leqslant q$ liczy różnicę między liczbą liter z S_2 a liczbą liter z T_2. Potem gdy napotka literę a_r dla $r > q$ sprawdza, czy liczba liter z S_2 równa się liczbie liter z T_2. Jeśli nie, to odrzuca słowa, a jeśli tak, to kontynuuje liczenie różnicy między S_1 a T_1. Na końcu sprawdza, czy ta różnica wynosi 0.

A zatem pozostaje zliczyć czwórki (S_1, T_1, S_2, T_2). Przypomnijmy sobie, że S_i oraz T_i są niekoniecznie rozłączne. Mamy więc $S'_i = S_i \setminus T_i$, $T'_i = T_i \setminus S_i$ oraz $U_i = S'_i \cup T'_i$ dla $i \in \{1, 2\}$. Podsumowując język L nie jest bezkontekstowy o ile zachodzi jeden z przypadków:

1. wszystkie S'_i, T'_i niepuste oraz $U_1 \cap U_2 \neq \emptyset$ i nie zachodzi $U_1 \subseteq U_2$ ani $U_2 \subseteq U_1$;
2. wszystkie S'_i, T'_i niepuste oraz istnieją indeksy $i < j < k < \ell$ takie, że $i, k \in U_1$ oraz $j, \ell \in U_2$ (lub analogicznie dla zamienionych U_1, U_2);
3. wszystkie S'_i, T'_i niepuste, $S'_1 \cap S'_2 = \emptyset$, ale $S'_1 \cap T'_2 \neq \emptyset$ lub $S'_2 \cap T'_1 \neq \emptyset$. Wówczas należy rozważyć $L_{T'_1, S'_1, S'_2, T'_2} = L$.
4. wszystkie S'_i, T'_i niepuste, $U_1 \subseteq U_2$ i $L_{S'_1, T'_1, S'_2 \setminus U_1, T'_2 \setminus U_1} = L$ jest językiem z przypadku (1), (2) lub (3) (lub analogicznie dla zamienionych U_1, U_2).

Zliczanie powyższych czwórek jest nieco skomplikowane, ale nie jest bezpośrednio związane z JAO, więc pozostawimy je bez szkicu rozwiązania.

3. Gramatykę bezkontekstową G nazwiemy *singletonową* jeśli $L(G)$ zawiera dokładnie jedno słowo. Zaprojektuj algorytm wielomianowy, który dla dwóch danych gramatyk singletonowych G_1, G_2 rozstrzyga, czy $L(G_1) = L(G_2)$.

Szkic rozwiązania To jest trudne zadanie. Uwaga: generowane słowo może być długości wykładniczej. Rozważmy np. gramatykę o nieterminalach $A_0, \ldots, A_n$ oraz produkcjach $A_0 \longrightarrow a$ i $A_k \longrightarrow A_{k-1}A_{k-1}$ dla każdego $k \in \{1, \ldots, n\}$. Wówczas $L(A_n) = \{a^{2^n}\}$, czyli jedyne generowane słowo jest długości wykładniczej. Z tego powodu wszystkie algorytmy, które obliczają jawnie generowane słowo muszą działać przynajmniej w czasie wykładniczym i nie są poprawnymi rozwiązaniami zadania.

Pomysł polega z grubsza na tym, że sprowadzamy gramatykę do postaci Chomsky'ego, a następnie rekurencyjnie rozstrzygamy, czy słowa generowane przez nieterminal X i nieterminal Y są swoimi prefiksami. Trzeba to jednak zrobić sprytnie. Gramatyki singletonowe nazywane są SLP (straight-line programs), można poczytać (m.in. o ich równoważności) tutaj: `https://arxiv.org/abs/1111.3244`.