Złożoność obliczeniowa (bioinformatyka)
egzamin, 7 lutego 2024

1. Które z następujących języków są regularne? Które są bezkontekstowe?
   (a) $X = \{a^n b^m c^k \mid n + k = m\}$;
   (b) $Y = \{a^n b^m c^k \mid n \cdot k = m\}$;
   (c) $Z = \{a^n b^m c^k \mid n \cdot k = 0\}$.

2. Rozpatrzmy następujące problemy decyzyjne:
   (a) Czy dana gramatyka bezkontekstowa generuje jakieś słowo złożone z samych liter a?
   (b) Czy dana maszyna Turinga akceptuje jakieś słowo złożone z samych liter a?

   Który z tych problemów jest rozstrzygalny? Który jest częściowo rozstrzygalny? Czy to możliwe, że któryś z nich jest NP-zupełny?

3. Udowodnić, że jeśli CSL $\subseteq$ NP, to NP = PSPACE.

4. Które z następujących inkluzji zachodzą? Które zawierania są właściwe?
   (a) $\mathrm{NTIME}(n \log n) \subseteq \mathrm{DSPACE}(n\sqrt{n})$;
   (b) co-$\mathrm{NSPACE}(n\sqrt{n}) \subseteq \mathrm{DSPACE}(n^4)$;
   (c) $\mathrm{NSPACE}(3n^2 + 2) \subseteq co$-$\mathrm{NSPACE}(n^2)$;
   (d) $\mathrm{DTIME}(3^{n^2}) \subseteq \mathrm{NTIME}(n\sqrt{n})$.

Przykładowe rozwiązania

1a: Język X jest bezkontekstowy. Generuje go na przykład taka gramatyka:

$$\xi_0 ::= \xi_1\xi_2, \quad \xi_1 ::= \varepsilon \mid a\xi_1 b, \quad \xi_2 ::= \varepsilon \mid b\xi_1 c.$$

Ale nie jest regularny, bo ma nieskończenie wiele różnych ilorazów, np. $X \setminus b^i = \{b^x c^{x+i} \mid x \in \mathbb{N}\}$.

1b: Język Y nie jest bezkontekstowy, bo nie spełnia lematu o pompowaniu. W przeciwnym razie niech $w = a^N b^{N^2} c^N$, gdzie N jest stałą z lematu. Jeśli $w = xyzuv$, gdzie $|yzu| \leq N$, to obie części pompowane y i u zawierają się albo w prefiksie $a^N b^{N^2}$ albo w sufiksie $b^{N^2} c^N$. Przypuśćmy, że zachodzi pierwszy przypadek (drugi jest analogiczny) i załóżmy, że w słowach y i u jest łącznie p liter a oraz łącznie q liter b. Ponieważ słowo xy^2zu^2v powinno należeć do Y, więc mamy równość $(N + p)N = N^2 + q$, skąd $pN = q$. To niemożliwe, bo $p + q \leq N$.

1c: Język Z jest regularny, bo $Z = a^*b^* \cup b^*c^*$.

2a: Jeśli L jest językiem bezkontekstowym, to $L \cap a^*$ też jest językiem bezkontekstowym. Co więcej, daną gramatykę dla języka L można łatwo (w pamieci logarytmicznej) przerobić na gramatykę dla języka $L \cap a^*$. Wystarczy każdą literę terminalną inną niż a zastąpić przez nowy symbol nieterminalny $\perp$, dla którego nie ma żadnej produkcji. Nasze zadanie sprowadza się więc do problemu niepustości dla języków bezkontekstowych, o którym wiadomo, że jest rozstrzygalny.

Problem niepustości języka bezkontekstowego należy do klasy P. A zatem możliwe, że ten problem jest NP-zupełny, ale wtedy P = NP. Wielomianowy algorytm dla niepustości można opisać tak: dla danej gramatyki $G = \langle \mathcal{A}, \mathcal{N}, P, \xi_0 \rangle$ konstruujemy zbiór $D = \{\xi \in \mathcal{N} \mid \exists k \in \mathbb{N}(\xi \twoheadrightarrow a^k)\}$, jako sumę wstępującego ciągu $D_0 \subseteq D_1 \subseteq D_2 \ldots$. Zaczynamy od $D_0 = \varnothing$, a potem ze zbioru D_i tworzymy $D_{i+1} = D_i \cup \{\xi \in \mathcal{N} \mid \exists w (w \in (D_n \cup a)^* \wedge (\xi \Rightarrow w) \in P)\}$. Każdy taki krok wymaga czasu proporcjonalnego do rozmiaru gramatyki, a liczba kroków też jest tego rzędu, bo dla pewnego $i \leq |\mathcal{N}|$ dostaniemy $D = D_i = D_{i+1}$. Na koniec sprawdzamy, czy $\xi_0 \in D$ i już.

2b: Z tym gorzej: to jest problem nierozstrzygalny. Można do niego sprowadzić problem stopu. Dla danej deterministycznej maszyny $\mathcal{M}$ i słowa wejściowego w skonstruujemy bowiem maszynę $\mathcal{N}_{M,w}$ w ten sposób, że $w \in L(\mathcal{M})$ wtedy i tylko wtedy, gdy maszyna $\mathcal{N}_{M,w}$ akceptuje jakieś słowo złożone z samych liter a. Maszyna $\mathcal{N}_{M,w}$ dla dowolnego wejścia $v \in \Sigma^*$ działa tak: uruchamia $\mathcal{M}$ na w i jeśli obliczenie się zakończy, to akceptuje. A zatem:

– Jeśli $w \in L(\mathcal{M})$, to $L(\mathcal{N}_{M,w}) = \Sigma^*$ (wtedy np. $aa \in L(\mathcal{N}_{M,w})$);
– Jeśli $w \notin L(\mathcal{M})$, to $L(\mathcal{N}_{M,w}) = \varnothing$ (wtedy do $L(\mathcal{N}_{M,w})$ nie należy żadne słowo $aa \ldots a$).

Skoro problem 2b jest nierozstrzygalny, to w szczególności nie należy do klasy NP i nie może być NP-zupełny. Ale jest częściowo rozstrzygalny, bo jeśli dana maszyna akceptuje jakieś słowo postaci $aa \ldots a$, to można to wykryć, systematycznie przeglądając początkowe fragmenty wszystkich możliwych obliczeń dla takich słów wejściowych, aż nie natrafi się na obliczenie akceptujące.

3: Przypuśćmy, że CSL $\subseteq$ NP, pokażemy, że NP $=$ PSPACE. Wiadomo, że zachodzi inkluzja w prawo, więc niech $L \in$ PSPACE, powiedzmy $L \in \mathrm{DSPACE}(n^k)$. Jeśli $L' = \{w\$^{|w|^k - |w|} \mid w \in L\}$, to $L' \in \mathrm{DSPACE}(n) \subseteq$ CSL. A skoro CSL $\subseteq$ NP, to $L' \in \mathrm{NTIME}(n^\ell)$ dla pewnego ℓ. Stąd wynika, że $L \in \mathrm{NTIME}((n^k)^\ell) = \mathrm{NTIME}(n^{k \cdot \ell}) \subseteq$ NP.

4a: Wiadomo, że $\mathrm{NTIME}(n \log n) \subseteq \mathrm{DSPACE}(n \log n)$. A zatem inkluzja zachodzi, bo $n \log n \leq n\sqrt{n}$ i jest właściwa, bo $\lim\limits_{n\to\infty} \dfrac{n \log n}{n\sqrt{n}} = 0$.

4b: Wiadomo , że co-$\mathrm{NSPACE}(n\sqrt{n}) = \mathrm{NSPACE}(n\sqrt{n}) \subseteq \mathrm{DSPACE}(n^3)$. Ponieważ $n^3 \leq n^4$, a granica ilorazu jest zerem, więc inkluzja zachodzi i jest właściwa.

4c: Ta inkluzja też zachodzi, ale jest niewłaściwa, bo klasy $\mathrm{NSPACE}(3n^2 + 2)$ i $\mathrm{NSPACE}(n^2)$ są identyczne, a na dodatek zamknięte ze względu na dopełnienie.

4d: Ostatnie zawieranie nie zachodzi: faktycznie mamy właściwą inkluzję w przeciwną stronę. A to dlatego, że $\mathrm{NTIME}(n\sqrt{n}) \subseteq \mathrm{DTIME}(2^{\mathcal{O}(n\sqrt{n})}) \subseteq \mathrm{DTIME}(2^{n \log n \sqrt{n}})$. Ta ostatnia klasa jest ostro zawarta w $\mathrm{DTIME}(3^{n^2})$ bo iloraz $\dfrac{2^{n \log n \cdot \sqrt{n}} \cdot n \log n\sqrt{n}}{3^{n^2}}$ dąży do zera.