

Egzamin ze złożoności obliczeniowej (bioinformatyka)
30 stycznia 2025

1. Dla każdego z poniższych języków proszę ustalić, czy jest on regularny, czy jest bezkontekstowy oraz czy jest deterministycznym językiem bezkontekstowym.¹
 - (a) $\{w \in \{a, c, t, g\}^* \mid \#(a, w) \neq \#(t, w) \vee \#(c, w) \neq \#(g, w)\}$;
 - (b) $\{w \in \{a, b\}^* \mid \forall i(3i \leq |w| \rightarrow w_{3i} = a)\}$;
 - (c) $\{a^n b^{n+3} \mid n \in \mathbb{N}\} \cap \{w \mid \#(a, w) \equiv 2 \pmod{3}\}$;
2. Dla każdego z poniższych problemów proszę określić, czy jest on rozstrzygalny, częściowo obliczalny i czy należy do klasy NLOGSPACE. Czy to możliwe, że któryś z nich jest P-zupełny?
 - (a) Czy dana maszyna Turinga akceptuje słowo 30.01.2025?
 - (b) Czy dany niedeterministyczny automat skończony akceptuje słowo 30.01.2025?
3. Które z tych inkluzji są prawdziwe? Które można zastąpić znakiem $\subsetneq$, a które znakiem $=$?
 - (a) $\text{NTIME}(n^3 + 2n) \subseteq \text{NTIME}(n^3)$;
 - (b) $\text{DTIME}(n^2 \sqrt{n}) \subseteq \text{DTIME}(n^2)$;
 - (c) $\text{NSPACE}(n \log n) \subseteq \text{DSpace}(n^3)$;
 - (d) $\text{NTIME}(n) \subseteq \text{DTIME}(2^{n\sqrt{n}})$.
4. Proszę udowodnić, że jeśli $\text{DSpace}(n) \subseteq \text{NP}$, to $\text{NEXPTIME} = \text{EXPSpace}$.

Przykładowe rozwiązania

1a: To jest język bezkontekstowy, bo można go przedstawić w postaci sumy dopełnień dwóch języków deterministycznych:²

$$-\{w \in \{a, c, t, g\}^* \mid \#(a, w) = \#(t, w)\} \cup -\{w \in \{a, c, t, g\}^* \mid \#(c, w) = \#(g, w)\}.$$

Nie jest to język deterministyczny (w szczególności nie jest to język regularny), bo jego dopełnienie $D = \{w \in \{a, c, t, g\}^* \mid \#(a, w) = \#(t, w) \wedge \#(c, w) = \#(g, w)\}$ nie jest językiem bezkontekstowym. Istotnie, przypuśćmy przeciwnie i niech N będzie stałą otrzymaną dla języka D z lematu o pompowaniu. Rozpatrzmy słowo $a^N c^N t^N g^N$. Jeśli rozbijemy je, zgodnie z lematem o pompowaniu, na 5 części $a^N c^N t^N g^N = xyzuv$, gdzie $|yzu| \leq N$ oraz $yu \neq \varepsilon$, to fragment yzu składa się tylko z liter a i c albo tylko z liter c i t albo tylko z liter t i g . W pierwszym przypadku w słowie xy^2zu^2v za mało będzie liter t lub liter g , w drugim liter a lub g , w trzecim liter a lub c .

1b: To jest język regularny $((a \cup b)(a \cup b)a)^*(\varepsilon \cup a \cup b)(\varepsilon \cup a \cup b)$, w szczególności jest to deterministyczny język bezkontekstowy.

¹Symbol $|w|$ oznacza długość słowa w , symbol $\#(a, w)$ oznacza liczbę wystąpień litery a w słowie w , a symbol w_i (gdzie $i > 0$) oznacza i -tą literę słowa w .

²Początkowo w tym miejscu była umieszczona przez pomyłkę gramatyka bezkontekstowa generująca... zupełnie inny język. Można ją oczywiście przerobić, ale wtedy robi się dość skomplikowana, a niniejsze rozwiązanie jest dużo prostsze.

1c: To nie jest język regularny, bo nie spełnia lematu o pompowaniu. Istotnie, przypuśćmy, że N jest stałą z lematu. Jeśli słowo $a^{3N+2}b^{3N+5}$ rozbijemy na trzy części $a^{3N+2}b^{3N+5} = xyz$, gdzie $|xy| \leq N$ oraz $y \neq \varepsilon$, to y składa się z samych liter a i słowo xy^2z ma za mało liter b .

Ale jest to deterministyczny język bezkontekstowy. Automat ze stosem odkłada na stos gwiazdki gdy czyta litery a a potem zdejmuje je, czytając litery b . Jednocześnie używa stanów wewnętrznych do kontrolowania reszty jaką liczbą dotychczas przeczytanych liter a daje przy dzieleniu przez trzy. Kiedy stos się opróżni automat sprawdza trzy ostatnie litery b i akceptuje pod warunkiem, że wspomniana reszta jest równa 2.

2a: Ten problem jest nierozstrzygalny, bo redukuje się do niego problem stopu. Dla danej deterministycznej maszyny M i słowa w można bowiem skonstruować maszynę $N_{M,w}$, która dla dowolnego wejścia v uruchamia maszynę M na wejściu w i jeśli to obliczenie się zakończy, to akceptuje. A zatem:

- Jeśli M zatrzymuje się na wejściu w , to $N_{M,w}$ akceptuje wszystkie słowa (w tym słowo 30.01.2025).
- W przeciwnym razie maszyna $N_{M,w}$ nie akceptuje niczego.

Gdyby problem akceptacji dzisiejszej daty był rozstrzygalny, to moglibyśmy też rozstrzygać problem stopu, a to niemożliwe. Oczywiście ten problem nie jest P-zupełny, bo problemy z klasy P są rozstrzygalne.

2b: Ten problem jest rozstrzygalny: wystarczy sprawdzić, czy w grafie automatu jest droga o krawędziach etykietowanych kolejnymi literami naszego słowa. To można zrobić w pamięci logarytmicznej nawet deterministycznie, bo długość słowa jest stała i można systematycznie przeglądać wszystkie drogi tej długości. Nie wiemy, czy ten problem jest P-zupełny, ale to by oznaczało równość klasy P i klasy LOGSPACE.

3a: Te klasy są równe, bo funkcja $n^3 + 2n$ jest $\mathcal{O}(n^3)$.

3b: Tu jest ostra inkluzja, ale w przeciwną stronę, bo iloraz $\frac{n^2 \log(n^2)}{n^2 \sqrt{n}} = \frac{2 \log n}{\sqrt{n}}$ dąży do zera dla $n \rightarrow \infty$.

3c: Tu jest ostra inkluzja z lewej do prawej, bo klasa $\text{NSPACE}(n \log n)$ zawiera się w $\text{DSpace}((n \log n)^2)$ (tw. Savitcha), a iloraz $\frac{(n \log n)^2}{n^3} = \frac{\log^2 n}{n}$ dąży do zera.

3d: Tu mamy: $\text{NTIME}(n) \subseteq \text{DTIME}(2^{\mathcal{O}(n)}) \subseteq \text{DTIME}(2^{n \log n}) \subsetneq \text{DTIME}(2^{n\sqrt{n}})$. Ostatnia inkluzja jest ostra, bo iloraz $\frac{2^{n \log n} \cdot \log(2^{n \log n})}{2^{n\sqrt{n}}}$ można zapisać jako $2^{n \log n + \log n + \log \log n - n\sqrt{n}}$, a wykładnik tego wyrażenia ma granicę $-\infty$.

4: Załóżmy, że $\text{DSpace}(n) \subseteq \text{NP}$ i niech $L \in \text{EXPSpace}$, tj. $L \in \text{DSpace}(2^{n^k})$ dla pewnego k . Niech $L' = \{w\$^{2^{|w|^k} - |w|} \mid w \in L\}$. Do języka L' należą słowa postaci $w\$ \dots \$$ o długości $2^{|w|^k}$, gdzie $w \in L$. Rozpoznawanie takich słów odbywa się w pamięci $2^{|w|^k}$, język L' jest więc w klasie $\text{DSpace}(n)$. Skoro z założenia $\text{DSpace}(n) \subseteq \text{NP}$, to istnieje niedeterministyczna maszyna rozpoznająca L' w czasie wielomianowym n^m . Można ją przerobić na maszynę rozpoznającą L w wykładniczym czasie $(2^{n^k})^m$. Zatem $L \in \text{NEXPTIME}$.