

Egzamin ze złożoności obliczeniowej – teoria. 14.6.2014

Imię i nazwisko:

Proszę wskazać i wyjaśnić ewentualne błędy w poniższych rozumowaniach. Odpowiedź powinna się zmieścić na tej kartce. Można zaznaczyć bezpośrednio w tekście miejsca uznane za błędne.

1. $\mathbf{NL} = \mathbf{NP}$. Pokażemy, jak rozwiązać problem 3-SAT przez złożenie obliczeń dwóch maszyn, jednej niedeterministycznej i jednej deterministycznej:

- maszyna M_1 przepisuje swoje wejście (formułę zdaniową) i dopisuje do niej zgadnięte niedeterministycznie wartościowanie zmiennych,
- maszyna M_2 sprawdza, deterministycznie, czy podane jej wartościowanie spełnia podaną formułę.

Każda z tych dwóch maszyn działa w pamięci logarytmicznej. Ich obliczenia możemy złożyć zwykłą metodą, tzn. możemy wykonywać obliczenie maszyną M_2 , a kiedy ta maszyna próbuje przeczytać k -ty element swojego wejścia, to za każdym razem uruchomić od nowa maszyną M_1 tak długo, aż wypisze k elementów swojego wyjścia.

W rezultacie otrzymujemy niedeterministyczną maszyną działającą w pamięci logarytmicznej, rozwiązującą problem 3-SAT. Ale jest to problem $\mathbf{NP}$ -zupełny. Wykazaliśmy więc, że $\mathbf{NL} = \mathbf{NP}$.

2. $\mathbf{NP} = \mathbf{co-NP}$. Korzystając z twierdzenia Immermanna-Szelepcsenyiego ($\mathbf{NL} = \mathbf{coNL}$) i posługując się techniką *paddingu*, pokażemy że $\mathbf{NP} = \mathbf{coNP}$.

Rozważmy dowolny język $L \in \mathbf{NP}$. Łatwo pokazać, że język

$$K = \{w\#0^{2^{|w|}} \mid w \in L\}$$

należy do klasy $\mathbf{NL}$, wobec czego dopełnienie K także należy do $\mathbf{NL}$. Ponieważ łatwo sprawdzić w pamięci logarytmicznej czy słowo kończy się odpowiednią liczbą zer, to także język

$$\tilde{K} = \{w\#0^{2^{|w|}} \mid w \notin L\}$$

należy do $\mathbf{NL}$. Z tego wynika, że dopełnienie języka L należy do $\mathbf{NP}$.

3. $\mathbf{NP} = \mathbf{PSPACE}$. Niech

$$\text{NEQ} = \{(M_1, M_2) : M_1, M_2 \text{ s\k{a} sko\k{n}czonymi niedeterministycznymi automatami i } L(M_1) \neq L(M_2)\}$$

Aby rozwi\k{a}za\c{c} ten problem w NP wystarczy zgadn\k{a}\c{c} s\k{ł}owo w i sprawdzi\c{c} (ju\k{z} deterministycznie), \k{z}e w nale\k{z}y do r\k{oz}nicy symetrycznej $(L(M_1) - L(M_2)) \cup (L(M_2) - L(M_1))$. Poniewa\k{z} problem NEQ jest zupe\k{ł}ny w PSPACE, otrzymujemy, \k{z}e $\mathbf{NP} = \mathbf{PSPACE}$.

4. $\mathbf{NP} \subseteq \mathbf{BPP}$. Prawdopodobie\k{n}stwo, \k{z}e losowy graf nie zawiera kliku 4-elementowej, jest dla dostatecznie du\k{z}ych graf\k{ow} bardzo ma\k{ł}e. Dok\k{ł}adniej, je\k{śli} dla grafu o wierzcho\k{ł}kach $\{1, 2, \dots, 4n + \alpha\}$ ($\alpha < 4$) interesuj\k{a} nas tylko kliku na blokach $4i + 1, 4i + 2, 4i + 3, 4i + 4$, to prawdopodobie\k{n}stwo, \k{z}e nie b\k{e}dzie takiej kliku, $(1 - \frac{1}{2^6})^m$ d\k{a}\k{z}y do 0, gdy $m \rightarrow \infty$.

Oczywi\k{st}e, graf zawieraj\k{a}cy klik\k{e} 4-elementow\k{a} nie mo\k{z}e by\c{c} 3-kolorowalny. Rozwa\k{z}my nast\k{e}puj\k{a}cy algorytm probabilistyczny dla problemu 3-kolorowalno\k{ści}. Algorytm zawsze m\k{ow}i NIE.

Czas dzia\k{ł}ania tego algorytmu jest wielomianowy (a nawet sta\k{ł}y r\k{own}y 1), a jego b\k{ł}ad, zgodnie z tym co powiedzieli\k{sm}y powy\k{z}ej, jest na pewno $\leq \frac{1}{4}$ (od pewnego miejsca, ale z tym mo\k{z}emy sobie \k{ł}atwo poradzi\c{c}, wbudowuj\k{a}c pocz\k{a}tkowe przypadki w maszyn\k{e} Turinga).

Jako \k{z}e problem 3-kolorowalno\k{ści} jest NP-zupe\k{ł}ny, wykazali\k{sm}y, \k{z}e $\mathbf{NP} \subseteq \mathbf{BPP}$, wbrew stwierdzeniu z Notatek, \k{z}e tak prawdopodobnie nie jest.