Egzamin ze złożoności obliczeniowej – teoria. 14.6.2013

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. Udowodnimy, że $NL = P$. Niech M będzie deterministyczną maszyną Turinga pracującą w czasie wielomianowym. Dla danego słowa w, graf G_w konfiguracji osiągalnych z konfiguracji początkowej z wejściem w jest wielomianowego rozmiaru. Aby sprawdzić, czy M akceptuje w, wystarczy sprawdzić, czy w grafie G_w istnieje ścieżka z konfiguracji początkowej do pewnej konfiguracji akceptującej. Jednak problem osiągalności w grafie skierowanym jest w klasie NL. A zatem dowolny problem z P zredukowaliśmy do problemu w NL, czyli $P \subseteq NL$. Inkluzja przeciwna jest oczywista.

2. Tym razem udowodnimy, że $P = NP$. Niech M będzie niedeterministyczną maszyną Turinga pracującą w czasie wielomianowym. Podobnie jak w poprzednim przykładzie, dla słowa wejściowego w konstruujemy graf G_w konfiguracji osiągalnych z konfiguracji początkowej. Jak poprzednio, pytanie, czy maszyna M akceptuje słowo w redukuje się do pytania, czy w grafie G_w istnieje ścieżka z konfiguracji początkowej do pewnej konfiguracji akceptującej. Jednak problem osiągalności w grafie skierowanym jest rozstrzygalny w czasie wielomianowym. A zatem dowolny problem z klasy NP zredukowaliśmy do problemu w P, czyli $NP \subseteq P$. Inkluzja przeciwna jest oczywista.

3. Z kolei wykażemy ciekawą własność klasy *NP*. Niech L będzie językiem w NP. Rozważmy następujący problem: **Dany** niedeterministyczny automat skończony $\mathcal{A}$. **Rozstrzygnij**, czy $\mathcal{A}$ rozpoznaje słowo należące do języka L.

Problem ten ma dość oczywisty algorytm NP: zgadnij ścieżkę w automacie $\mathcal{A}$ ze stanu początkowego do pewnego stanu akceptującego, ścieżka ta wyznacza pewne słowo w. Następnie przy pomocy (niedeterministycznego) algorytmu dla L sprawdź, czy $w \in L$; czas całości pozostaje wielomianowy.

Ta prosta obserwacja obala pewne przesądy na temat maszyn Turinga. Ustalmy mianowicie uniwersalną maszynę Turinga U. Przypomnijmy, że U na wejściu postaci $\langle M \rangle w$ symuluje obliczenie M na w. Zbiór wszystkich obliczeń akceptujących maszyny U (traktowanych jako ciągi konfiguracji) może być przedstawiony jako język $Obl(U)$ należący do klasy NP. Natomiast dla dowolnej maszyny M możemy skonstruować automat skończony A_M, który sprawdza, czy dane słowo rozpoczyna się od konfiguracji początkowej dla słowa $\langle M \rangle$. Wtedy A_M akceptuje słowo z języka $Obl(U)$ dokładnie wtedy, gdy maszyna M akceptuje ε. Potrafimy więc rozstrzygać (i to nawet w NP), czy dana maszyna Turinga M akceptuje słowo puste, podczas gdy w różnych książkach wciąż jeszcze ściemnia się, że jest to problem „nierozstrzygalny".

4. Wątpliwe twierdzenie. Na obu wykładach podano tzw.

Twierdzenie PCP. Dla każdego problemu L w NP istnieje wielomianowy algorytm probabilistyczny (weryfikator) V, który dla wejścia o długości n używa ciągu bitów losowych $U_{r(n)}$ długości $r(n) = \mathcal{O}(\log n)$ i zadaje **stałą** liczbę M pytań o wybrane bity tzw. dowodu, przy czym

- jeśli $x \in L$, to istnieje dowód π_x, że $\Pr\big(V(x, U_{r(n)}, \pi_x) = \text{TAK}\big) = 1$,
- $x \notin L$, to dla każdego π, $\Pr\big(V(x, U_{r(n)}, \pi) = \text{TAK}\big) < \frac{1}{2}$.

To już jest wyjątkowa ściema, bo gdyby tak było to $P = NP$. Oto algorytm:

Dla wejścia x długości n przeszukujemy kolejno ciągi bitów r o długości $r(n)$ (jest ich wielomianowo wiele). Dla każdego takiego r symulujemy obliczenie V na wejściu x z ciągiem losowym r i staramy się znaleźć takie odpowiedzi na M pytań, przy których V doszedł by do akceptacji. Jeśli się udało, zapisujemy te odpowiedzi. Przy kolejnym badanym r' dbamy o to, by znalezione dlań odpowiedzi były niesprzeczne z tymi zapisanymi dla r (w przeciwnym razie nie uznajemy ich).

Jeśli istnieje dowód π_x, to powyższe obliczenie zakończy się sukcesem dla każdego r (znajdujemy odpowiedzi zgodne z dowodem). Ale również na odwrót – jeśli obliczenie zakończy się sukcesem, to dzięki niesprzeczności poszczególnych odpowiedzi otrzymamy dowód.

Tak więc $P = NP$ i zamiast tracić czas na egzaminie, powinniśmy udać się do Instytutu Claya w celu zainkasowania 1,000,000 $ (gdyż jest to Problem Milenijny).