ZŁO* — ćwiczenia 12

Randomizacja, klasy z podpowiedzią

Ćwiczenia oznaczone przez ♢ są z poprzednich ćwiczeń. Ćwiczenia oznaczone przez ♠ będziemy robić w miarę możliwości czasowych.

Zadania

Zadanie 1 (♢)**.** Wykaż, że jeśli $\mathsf{NP} \subseteq \mathsf{BPP}$, to $\mathsf{NP} = \mathsf{RP}$.

Klasa ZPP obejmuje problemy posiadające wielomianowy algorytm Las Vegas: algorytm używający randomizacji, który jest zawsze poprawny, ale którego czas działania ma wartość oczekiwaną ograniczoną wielomianowo.

Zadanie 2 (♢)**.** Wykaż, że $\mathsf{ZPP} = \mathsf{RP} \cap \mathsf{coRP}$.

Niech A będzie klasą złożoności. Klasa A/poly obejmuje języki L, dla których istnieje ciąg słów $(\alpha_n)_{n\in\mathbb{N}}$ oraz algorytm $\mathcal{A}$ działający w klasie A o następujących własnościach:

- $|\alpha_n| \leqslant p(n)$ dla jakiegoś wielomianu $p(\cdot)$;
- $\mathcal{A}$ na wejściu $(x, \alpha_{|x|})$ rozstrzyga, czy $x \in L$.

Zadanie 3. Udowodnij, że $\mathsf{RP} \subseteq \mathsf{P}/\mathrm{poly}$.

Ustalmy binarny alfabet $\Sigma = \{0, 1\}$. *OR-destylacją* języka $L \subseteq \Sigma^\star$ w język $R \subseteq \Sigma^\star$ nazwiemy algorytm, który:

- Bierze na wejściu dowolnie długi ciąg instancji $x_1, x_2, \ldots, x_t$.
- Działa w czasie wielomianowym od długości swojego wejścia.
- Zwraca jedną instancję y spełniającą $|y| \leqslant p(\max_{i=1,\ldots,t} |x_i|)$ dla pewnego wielomianu $p(\cdot)$, przy czym ma zachodzić, że $y \in R$ wtedy i tylko wtedy gdy $x_i \in L$ dla co najmniej jednego indeksu $i \in \{1, 2, \ldots, t\}$.

Inaczej mówiąc, OR-destylacja kompresuje wejściowe instancje w jedną, wielkości wielomianowej od maksimum z wielkości wejściowych instancji, zachowując logiczny OR odpowiedzi na wejściowe instancje.

Zadanie 4. Załóżmy, że mamy OR-destylację $\mathcal{A}$ z L w R; niech $p(\cdot)$ będzie ograniczeniem na wielkość wyjściowej instancji y. Rozważmy odpalanie $\mathcal{A}$ na t-krotkach słów długości co najwyżej k. Powiemy, że słowo x długości co najwyżej k jest *pokryte* przez słowo y długości co najwyżej $p(k)$, jeśli istnieje krotka $(x_1, x_2, \ldots, x_t)$ taka, że $\mathcal{A}(x_1, x_2, \ldots, x_t) = y$ oraz $x = x_i$ dla jakiegoś indeksu i. Udowodnij, że jeśli ustawimy $t \geqslant p(k) + 1$, to istnieje zawsze takie $y \in \Sigma^{\leqslant p(k)} - R$, które pokrywa co najmniej połowę słów $x \in \Sigma^{\leqslant k} - L$.

Zadanie 5. Wykaż, że przy założeniach poprzedniego zadania, w szczególności $t \geqslant p(k)+1$, istnieje podzbiór $S \subseteq \Sigma^{\leqslant p(k)} - R$ wielkości poly(k), którego elementy w sumie pokrywają wszystkie elementy $\Sigma^{\leqslant k} - L$.

Zadanie 6 (Fortnow i Santhanam, 2008)**.** Udowodnij, że jeśli język L posiada OR-destylację do jakiegokolwiek języka R (nawet nierozstrzygalnego), to $L \in \mathsf{coNP/poly}$. Wysnuj wniosek, że jeśli jakikolwiek NP-trudny język L posiada OR-destylację do jakiegokolwiek języka R, to $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

Zadanie 7 (♠)**.** Ustalmy jakiekolwiek sensowne kodowanie problemu SAT przy pomocy ciągu bitów, np. po kolei podajemy klauzule kodując binarne indeksy zmiennych w literałach. W problemie Succint SAT dany jest obwód C o n wejściach i jednym wyjściu. Przez $\varphi(C)$ oznaczamy formułę logiczną uzyskaną następująco: do C wstawiamy po kolei wszystkie ciągi bitowe długości n, w kolejności leksykograficznej, wyniki C na tych ciągach ustawiamy w słowo, i interpretujemy to słowo jako zapis bitowy formuły $\varphi(C)$. W ten sposób, $\varphi(C)$ jest zapisywana na 2^n bitach. Dla danego obwodu C, pytamy się, czy $\varphi(C)$ jest spełnialna. Udowodnij, że problem Succint SAT jest zupełny dla klasy $\mathsf{NEXPTIME}$ pod P-redukcjami. Kodowanie bitowe dla problemu SAT możesz sobie dobrać dowolnie.