ZLOŻONOŚĆ OBLICZENIOWA - WYK. 13

1. **Dowody interaktywne.** Klasyczne dowodzenie twierdzeń odpowiada klasie **NP**: ktoś przedstawia dowód (świadka), który trzeba sprawdzić w czasie wielomianowym. A może jak na egzaminie ustnym: weryfikator przepytuje dowodziciela i w ten sposób szybciej sprawdza co się dzieje?

   Przykład: sprawdzenie nieizomorficzności grafów. O tym problemie nie wiadomo czy jest w **NP**, ale jest dla niego łatwy dowód interaktywny.

2. Formalnie: weźmy dwie funkcje $V, P : \{0,1\}^* \to \{0,1\}^*$. V to *verifier*, P to *prover*. Interakcja k-rundowa między nimi na słowie wejściowym w to ciąg słów:

$$\begin{aligned} q_1 &= V(w) \\ a_1 &= P(w, q_1) \\ q_2 &= V(w, q_1, a_1) \\ a_2 &= P(w, q_1, a_1, q_2) \\ &\cdots \\ a_k &= P(w, \cdots, q_k) \\ \texttt{out}(V,P)(w) &= V(w, \cdots, a_k) \end{aligned}$$

   Mówimy, że język L ma deterministyczny protokół interaktywny (klasa **dIP**) jeśli istnieje maszyna deterministyczna V działająca w czasie wielomianowym, oraz wielomianowa liczba rund, takie że

$$w \in L \iff \exists P. \texttt{out}(V,P)(w) = 1$$

   Na funkcję P nie nakładamy żadnych ograniczeń.

3. Okazuje się, że deterministyczna interakcja nie zwiększa mocy obliczeniowej: **dIP** = **NP**.

   Co z protokołem na nieizomorficzność grafów? Okazuje się, że sytuacja się zupełnie zmienia jeśli dopuścimy, by V była maszyną probabilistyczną.

4. Formalnie: weźmy dwie funkcje $V, P : \{0,1\}^* \to \{0,1\}^*$. V to *verifier*, P to *prover*. Interakcja k-rundowa między nimi na słowie wejściowym w i z dostępem do wielomianowej liczby bitów losowych r to ciąg słów:

$$\begin{aligned} q_1 &= V(w, r) \\ a_1 &= P(w, q_1) \\ q_2 &= V(w, r, q_1, a_1) \\ a_2 &= P(w, q_1, a_1, q_2) \\ &\cdots \\ a_k &= P(w, \cdots, q_k) \\ \texttt{out}(V,P)(w) &= V(w, r, \cdots, a_k) \end{aligned}$$

Całą interakcję, a także wynik $\texttt{out}(V,P)(w)$, traktujemy jako zmienną losową względem r. Zauważmy, że P nie ma dostępu do r.

Mówimy, że język $L \in \mathbf{IP}$ jeśli istnieje maszyna V działająca w czasie wielomianowym, oraz wielomianowa liczba rund, takie że

$$w \in L \Longrightarrow \exists P. Pr_r[\texttt{out}(V,P)(w) = 1] \geq \frac{3}{4}$$
$$w \notin L \Longrightarrow \forall P. Pr_r[\texttt{out}(V,P)(w) = 1] \leq \frac{1}{4}$$

Czyli: dla każdego słowa w L istnieje prover, który nam tego dowodzi z dużym prawdopodobieństwem, a dla słów spoza L każdy prover może nas oszukać tylko z małym prawdopodobieństwem.

Na funkcję P nie nakładamy żadnych ograniczeń.

5. Błąd $\frac{1}{4}$ w definicji można dowolnie zbliżać do zera.

   Dowód: Możemy zrobić taką samą aplifikację jak w przypadku **BPP**, kosztem zwiększenia liczby rund. **Ale** jest pewna subtelność: dla słów spoza L ciężko wykonać m niezależnych eksperymentów, bo może są provery, które spróbują skorzystać z zapytań zadanych w poprzednich eksperymentach? *Rozwiązanie:* To nie jest problem, bo V nie pamięta jakie pytania zadał w poprzednich eksperymentach. Gdyby więc istniał prover, który w m-tym eksperymencie uzyskuje duże prawdopodobieństwo błędu (błędnej akceptacji), to istniałby też prover, który od początku uznaje że wcześniej widział wyniki poprzednich $m-1$ eksperymentów i od razu uzyskuje duże prawdopodobieństwo błędu.

6. W protokole dla nieizomorfizmu grafów, jeśli grafy są nieizomorficzne to istnieje prover, przy którym zaakceptujemy zawsze, czyli w pierwszej klauzuli uzyskujemy nawet prawdopodobieństwo 1. Udowodnimy dziś, że wstawienie tam 1 nie zmienia klasy **IP**, czyli możemy wymagać proverów, które nas *zawsze* przekonają dla słów w L.

7. Z kolei wstawienie 0 w drugiej klauzuli (jeśli słowo nie jest w L to żaden prover nigdy nas nie przekona) zmniejsza klasę **IP** do **NP**.

   Dowód (ćwiczenie): świadek dla słowa w to ciąg bitów losowych wraz z pełną interakcją; weryfikator tylko sprawdza że ta interakcja jest akceptująca, podobnie jak w dowodzie **dIP**=**NP**.

8. **Dygresja:** W klasie **IP**, prover nie widzi bitów losowanych przez V. A gdyby nie mógł przewidzieć przyszłych losowań, ale widział te już dokonane? To się nazywa protokół Artur-Merlin (**AM**). Liczby rund oznaczamy tak: $\mathbf{IP}[k]$, $\mathbf{AM}[k]$. Nasz protokół dla nieizomorfizmu grafów nie jest typu **AM**. Provery **AM** mają więcej mocy niż provery **IP**.

   Fakt: $\mathbf{AM}[k] \subseteq \mathbf{IP}[k]$. *Dowód:* provery **AM** umieją symulować provery **IP**, więc warunek dla słów spoza L jest jasny. Ale jeśli dla słowa

L istnieje dobry prover **AM**, to dlaczego istnieje dobry prover **IP**? Tutaj przerabiamy weryfikator tak, by w interakcji jawnie podawał proverowi swoje bity losowe, ułatwiając mu zadanie.

Twierdzenie (Goldwasser-Sipser'87): $\mathbf{IP}[k] \subseteq \mathbf{AM}[k+2]$. *Dowód* trudny.

Idea: istnieje protokół **AM** dla takiego problemu: niech $S \subseteq \{0,1\}^m$ będzie zbiorem danym tak, by dało się efektywnie sprawdzać należenie do niego. Dla danej liczby K, prover stara się przekonać V że $|S| \geq K$, a weryfikator powinien odrzucić jeśli $|S| \leq \frac{K}{2}$. *Pomysł:* V losuje funkcję $h : \{0,1\}^m \to \{0,1\}^k$ z ustalonej rodziny funkcji haszujących, dla $k \approx \log K$. Ponadto V losuje jakiś ciąg $v \in \{0,1\}^k$ i wysyła to wszystko do P, a P ma przedstawić takie x że $h(x) = v$.

A co jeśli P ma dostęp do wszystkich bitów losowych? Ta klasa na pewno zawiera **BPP** (brak interakcji) oraz **NP** (brak bitów losowych). Widać też że zawiera się w **IP**, ale co dalej nie wiem. Ludzie jakoś nie interesują się tą klasą.

9. Jak duża jest klasa **IP**?

   - wiemy że **dIP**=**NP**.
   - z drugiej strony wierzymy że **BPP** = **P**, czyli że randomizacja niewiele wnosi.
   - wiemy że nieizomorficzność grafów jest w **IP** a nie wiadomo czy jest w **NP**.
   - z drugiej strony podejrzewamy że nieizomorficzność jest w **P**, więc to może nie jest bardzo silny argument.
   - na początku podejrzewano, że nawet jeśli **IP** jest większe niż **NP**, to raczej nie zawiera **coNP**.

10. Zaskakujące *Twierdzenie* (Lund, Fortnow, Karloff, Nisan, Shamir'90): **IP** = **PSPACE**.

    Historia tego zaskoczenia jest ładnie opisana w artykule L. Babai *E-mail and the unexpected power of interaction.*

    Ćwiczenie: Pokazać zawieranie **IP** $\subseteq$ **PSPACE**.

    Pokażemy teraz zawieranie **PSPACE** $\subseteq$ **IP**.

11. Najpierw pokażmy, że **coNP** $\subseteq$ **IP** (LFKN). Do tego wystarczy pokazać, że 3NSAT $\in$ **IP**. Pokażemy nawet coś lepszego: że do **IP** należy język

    $$\{(\phi, K) \mid \phi \text{ jest 3CNF i ma dokładnie } K \text{ wartościowań spełniających}\}$$

    Faktycznie jeśli ten problem jest w **IP** to 3NSAT też, bo wystarczy że P na początek poda liczbę $K > 0$, a potem wykona się normalna interakcja.

Potraktujmy formułę logiczną jako wielomian:

$$\begin{aligned} x \wedge y &\leftrightarrow X \cdot Y \\ \neg x &\leftrightarrow 1 - X \\ x \vee y &\leftrightarrow 1 - (1 - X)(1 - Y) \\ x \vee \neg y \vee z &\leftrightarrow 1 - (1 - X)Y(1 - Z) \end{aligned}$$

Jeśli formuła ϕ jest 3CNF, to dostajemy wielomian $P_\phi(x_1, \ldots, x_n)$ stopnia $3k$, gdzie k jest liczbą klauzul w ϕ, który w dowolnym ciele ewaluuje się do 1 dla wartościowań spełniających ϕ i do 0 dla niespełniających. Chcemy sprawdzić, że

$$K = \sum_{v_1 \in \{0,1\}} \sum_{v_2 \in \{0,1\}} \cdots \sum_{v_n \in \{0,1\}} P_\phi(v_1, \ldots, v_n).$$

Będzie nam zależało, żeby liczyć wszystko w jakimś ciele skończonym. Mamy gwarancję, że suma po prawej ma wartość co najwyżej 2^n, więc P i V muszą ustalić jakąś liczbę pierwszą $p > 2^n$ i dalej liczyć wszystko modulo p. Pytanie jak ją ustalić? V nie ma na to mocy obliczeniowej, ale P tak. Tak więc pierwszy krok interakcji:

- P przedstawia liczbę $2^n < p \leq 2^{2n}$ (żeby nie była za duża), a V weryfikuje (np. testem AKS, albo probabilistycznie) że jest ona pierwsza.

Mamy teraz takie zadanie: dane są wielomian $g(x_1, \ldots, x_n)$ stopnia całkowitego d, liczba K i liczba pierwsza p. Sprawdzić interaktywnie, czy

$$K = \sum_{v_1 \in \{0,1\}} \sum_{v_2 \in \{0,1\}} \cdots \sum_{v_n \in \{0,1\}} g(v_1, \ldots, v_n) \tag{1}$$

modulo p. Możemy przy tym założyć, że V umie łatwo obliczać wartości g dla danych argumentów.

Protokół idzie tak:

- V prosi P o przysłanie wielomianu jednej zmiennej:

  $$h(x_1) = \sum_{v_2 \in \{0,1\}} \cdots \sum_{v_n \in \{0,1\}} g(x_1, v_2, \ldots, v_n).$$

  Ten wielomian ma stopień d, a wszystkie współczynniki są co najwyżej p, więc powinien się dać krótko zapisać.
- P przysyła jakiś wielomian $s(x_1)$ (twierdząc, że $s(x_1) = h(x_1)$).
- V sprawdza, że $s(0) + s(1) = K$. Jeśli tak nie jest, to odrzuca.

Zastanówmy się, jak do tej pory radzi sobie P w sytuacji, kiedy równanie (1) nie zachodzi: gdyby przysłał V prawdziwe h jako s, to byłoby $s(0) + s(1) \neq K$ i V by odrzucił. Tak więc P przysłał s różne od h.

Skoro wielomian $s(x) - h(x)$ ma stopień d, to ma co najwyżej d pierwiastków. W związku z tym, dla losowego $0 \leq a < p$, prawdopodobieństwo że $s(a) = h(a)$ jest co najwyżej $\frac{d}{p}$. Dlatego interakcja idzie dalej:

- V losuje a z wybranego ciała i dalej rozwiązuje taki sam problem jak (1), ale z mniejszą liczbą zmiennych:

$$s(a) = \sum_{v_2 \in \{0,1\}} \cdots \sum_{v_n \in \{0,1\}} g(a, v_2, \ldots, v_n)$$

- Na końcu, kiedy została już tylko jedna zmienna $n = 1$, V sprawdza po prostu czy $g(0) + g(1) = K$.

Jeśli równanie (1) zachodzi, to istnieje "prawdomówny" prover, który zawsze przekona o tym V.

Jeśli równanie nie zachodzi, to każdy prover zostanie przyłapany z prawdopodobieństwem

$$\left(1 - \frac{d}{p}\right)^n \geq \left(1 - \frac{dn}{p}\right) \geq \frac{3}{4}$$

(nierówność Bernoulliego), bo p jest wykładnicze względem n, a d jest małe, równe najwyżej $\mathcal{O}(n^3)$, bo tylko tyle klauzul może być w ϕ.

12. Teraz pokażmy właściwe twierdzenie, czyli **PSPACE** $\subseteq$ **IP** (Shamir). W tym celu pokażemy podobny protokół dla problemu QBF, też oparty na arytmetyzacji.

Dla formuły kwantyfikowanej

$$\Phi = \exists x_1 \forall x_2 \cdots \forall x_n \phi(x_1, \ldots, x_n)$$

chcemy sprawdzić, czy

$$0 < \sum_{v_1 \in \{0,1\}} \prod_{v_2 \in \{0,1\}} \cdots \prod_{v_n \in \{0,1\}} g(v_1, \ldots, v_n).$$

Niestety tutaj nie da się zastosować tego samego protokołu, bo mnożenie zwiększa stopień wielomianu. Odpowiednie wielomiany jednej zmiennej mogą mieć wykładniczy stopień i być zbyt długie, żeby dało się je przesłać w wielomianowym czasie podczas interakcji.

Aby sobie z tym poradzić, przeróbmy formułę Φ na równoważną w sensie QBF, ale taką, żeby żadna zmienna nie mogła przetrwać więcej niż 2 kwantyfikatory. Konkretnie, idziemy od prawej do lewej i przy każdym kwantyfikatorze ogólnym:

$$\forall x_i . \theta(x_1, \ldots, x_i)$$

gdzie θ może kwantyfikować po zmiennych $x_{i+1}, \ldots, x_n$, zmieniamy formułę na

$$\forall x_i . \exists y_1, \ldots, y_i . (x_1 = y_1) \wedge \cdots \wedge (x_i = y_i) \wedge \theta(y_1, \ldots, y_i).$$

W ten sposób dostajemy równoważną formułę o n^2 zmiennych i o kwadratowej długości. Można ją normalnie zarytmetyzować, przy czym $x = y$ zastępujemy przez wielomian

$$x \cdot y + (1 - x) \cdot (1 - y)$$

Wtedy okazuje się, że każdy z przesyłanych w protokole wielomianów jest stopnia $\mathcal{O}(n)$, gdzie n jest długością ϕ.

13. *Uwaga 1:* w tych protokołach losowe wybory V nie są ukrywane przed P. Pokazaliśmy więc, że **AM**[poly]=**IP**, bez użycia twierdzenia Goldwasser-Sipsera.

14. *Uwaga 2:* W obu protokołach słowa z L są dla odpowiednich proverów zawsze akceptowane. W ten sposób udowodniliśmy więc, że w pierwszym punkcie definicji **IP** można zamiast $\frac{3}{4}$ wpisać 1 i nie zmienimy klasy **IP**.