

Wykład 3. Protokoły z Wiedzą Zerową

Wykładowca: Stefan Dziembowski

Skryba: Łukasz Chmielewski i Tomasz Okniński

Streszczenie. Na początku wykładu podany jest przykład dowodu z wiedzą zerową (dowód izomorfizmu grafów). Następnie podane zostaną definicje obliczeniowej / statystycznej / idealnej nierozróżnialności. Korzystając z tych definicji i definicji dowodu interakcyjnego są przytoczone definicje dowodów z wiedzą zerową (następnie zostają one trochę rozszerzone).

Jako kolejne podane są dowody twierdzeń:

- protokół dowodu izomorfizmu grafów jest dowodem z idealną wiedzą zerową.
- główne twierdzenie: klasa języków NP należy do klasy języków mających dowód z obliczeniową wiedzą zerową.
Jest również zdefiniowane pojęcie zobowiązania bitowego.

1 Wstęp

(notatki są po części oparte na [1])

Na obecnym wykładzie zajmiemy się *Dowodami z wiedzą zerową* (ang.: *Zero-knowledge proof*). Jest to szczególny rodzaj dowodów interakcyjnych.

Ich cechą jest to, że Weryfikator (nawet jeśli zachowuje się niezgodnie z protokołem) nie dowiadyje się niczego poza faktem, który jest dowodzony.

Np.: w przypadku izomorfizmu grafów Weryfikator dowie się, że 2 grafy są izomorficzne, ale nie pozna tego izomorfizmu. Inny przykład podaje Goldreich ([1], na stronie 128).

Dowody z wiedzą zerową przydają się w konstruowaniu innych bardziej skomplikowanych protokołów (bo mamy narzędzie, pozwalające by dowolny uczestnik protokołu potrafił dowieść, że zachowuje się poprawnie, bez ujawniania swoich sekretów). Mają też zastosowanie w systemach identyfikacji.

Przyjmujemy oznaczenie, że P oznacza Udowadnicza (Prover), a V Weryfikatora (Verifier).

2 Przykład: Izomorfizm grafów

Zanim zostanie podana definicja pokażemy przykład dowodu z wiedzą zerową.

Jest to dowód *izomorfizmu grafów*.

Przypomnijmy: $GI = \{(G_1, G_2) : G_1 \cong G_2\}$.

Zauważmy, że dowód interaktywny tego faktu (bez wymagania o wiedzy zerowej) jest trywialny. Wystarczy, że Udowadnicz P poda izomorfizm (\cong) Weryfikatorowi V , a on po prostu go sprawdzi.

Dowód z wiedzą zerową:

1. wspólnym wejściem P i V jest para grafów: (G_1, G_2) .
2. P : Spermuj losowo G_2 . Wyślij rezultat (G') do V .
3. V : Wyślij losową wartość $\sigma \in \{1, 2\}$ do P .
4. P : Wyślij do V izomorfizm między G' , a G_σ .
5. V : Jeśli otrzymałeś poprawny izomorfizm odpowiedz true, w przeciwnym przypadku false.

Fakt 1 Powyższy dowód jest dowodem interaktywnym dla problemu izomorfizmu grafów.

Dowód:

Pełność: Jeśli $G_1 \cong G_2$, to P postępując zgodnie z powyższym protokołem spowoduje, że V zawsze zaakceptuje wysłany izomorfizm (bo grafy G_1 , G_2 i G' są ze sobą izomorficzne).

Poprawność: Jeśli $G_1 \not\cong G_2$, to dowolny P zostanie przyłapany z prawdopodobieństwem 0.5. Jeśli $\sigma = 2$ to P nie zostanie przyłapany (gdyż G' to spermowane G_2 i wysłany izomorfizm to właśnie ta permutacja). W przeciwnym przypadku V zorientuje się, że $G_1 \not\cong G'$, gdyż skoro $G' \cong G_2$ i $G_1 \not\cong G_2$ to $G_1 \not\cong G'$. Zatem prawdopodobieństwo wynosi 0.5, bo wybór σ dokonywany jest raz.

□

Ponadto V nie pozna niczego poza faktem, że grafy są izomorficzne, gdyż P wysyła izomorfizm między G' i którymś z grafów (G_1 i G_2). A z tego przekształcenia nie można „odczytać” izomorfizmu pomiędzy G_1 i G_2 .

Czyli powyższy dowód jest dowodem z wiedzą zerową.

3 Definicja

3.1 Intuicje

Co to znaczy, że V nie poznaje niczego poza faktem, że $x \in L$?

Jest to nietrywialne pytanie.

Nieformalna odpowiedź jest taka: Weryfikator nie jest w stanie obliczyć niczego, czego nie mógłby obliczyć wiedząc, że $x \in L$.

Trochę formalniej: będziemy wymagać, żeby dla każdego Weryfikatora V istniał *symulator* M , który zwraca (prawie) dokładnie to co V .

3.2 Formalizacja

3.2.1 Nierozróżnialność

Wprowadzamy najpierw pojęcie *obliczeniowej nierozróżnialności* (ang.: *computational indistinguishability*).

Dla dowolnego $S \subseteq \{0, 1\}^*$ przez *zespół prawdopodobieństw* (ang.: *probability ensemble*) rozumiemy ciąg

$$X = \{X_\alpha\}_{\alpha \in S},$$

gdzie każde X_α jest rozkładem prawdopodobieństwa na ciągach zerojedynkowych długości wielomianowej w $|\alpha|$.

Mówimy, że zespoły X i Y są *obliczeniowo nierozróżnialne*, jeśli dla dowolnego WAP A funkcja

$$d(n) = \max_{\alpha \in \{0,1\}^n} \{|P[A(X_\alpha, \alpha) = 1] - P[A(Y_\alpha, \alpha) = 1]|\}$$

jest zaniedbywalna.

Jeśli odrzucimy ograniczenie na moc obliczeniową A to uzyskamy definicję *statystycznej nierozróżnialności*.

Jeśli X i Y są równe (czyli $d(n) = 0$), to mówimy, że są *doskonale nierozróżnialne* (ang.: *perfectly indistinguishable*).

Powyzszą notację rozszerzamy na zmienne losowe (tzn.: A i B są obliczeniowo / statystycznie / idealnie nierozróżnialne, jeśli ich rozkłady są obliczeniowo / statystycznie / idealnie nierozróżnialne).

3.2.2 Podstawowa definicja protokołu z wiedzą zerową

Definicja 2 Niech (P, V) będzie dowodem interaktywnym dla języka L . Mówimy, że (P, V) jest dowodem z obliczeniową wiedzą zerową jeśli dla dowolnego WAP V^* istnieje WAP M^* (symulator) taki, że zespoły:

- $\{(P, V^*)(x)\}_{x \in L}$
- $\{(M^*(x))\}_{x \in L}$

są obliczeniowo nierozróżnialne.

Jeśli w powyższej definicji „obliczeniowo nierozróżnialne” zamienimy na „statystycznie nierozróżnialne”, to otrzymamy definicję dowodu ze *statystyczną wiedzą zerową*.

Aby otrzymać definicję dowodu z *idealną wiedzą zerową* zamieniamy „obliczeniowo nierozróżnialne” na „idealnie nierozróżnialne” oraz liberalizujemy definicję w następujący sposób: pozwalamy, by M z prawdopodobieństwem co najwyżej 0.5 zwrócił specjalny symbol \perp i wymagamy, by rozkłady były identyczne, *pod warunkiem, że M nie zwrócił \perp* .

Ponieważ błąd 0.5 symulatora można zmniejszać (powtarzając dowód kilkakrotnie) zachodzi zależność: jeśli dowód jest dowodem z idealną wiedzą zerową to jest dowodem ze statystyczną wiedzą zerową. Zachodzi również zależność: jeśli dowód jest dowodem ze statystyczną wiedzą zerową to jest dowodem z obliczeniową wiedzą zerową.

3.2.3 Rozszerzenia definicji

Definicję 2 zwykle rozszerza się następująco:

1. Wymagamy by pełność zachodziła tylko z jakimś znaczącym prawdopodobieństwem np.: 2/3.
2. Pozwalamy, by obie strony pobierały prywatne *pomocnicze wejście* (ang.: *auxiliary input*).

Pomijamy szczegóły (znajdują się w [1] na stronie 151).

Przez *dowód z obliczeniową/statystyczną/idealną wiedzą zerową* będziemy teraz rozumieć dowód w sensie rozszerzonej definicji.

Jeśli mówimy *dowód z wiedzą zerową* to mamy na myśli dowód z *obliczeniową* wiedzą zerową.

\mathcal{CZK} (lub po prostu \mathcal{ZK}) jest klasą języków mających dowód z obliczeniową wiedzą zerową.

Podobnie można zdefiniować \mathcal{SZK} (klasa języków ze statystyczną wiedzą zerową) i \mathcal{PK} (klasa języków z idealną wiedzą zerową).

4 Przykład dowodu

Twierdzenie 3 $GI \in \mathcal{PZK}$

Dowód:

(Zarys dowodu, szczegóły dostępne na stronie 147 [1])

Pokażemy, że protokół z Rozdziału 2 jest dowodem z idealną wiedzą zerową.

Weźmy jakiś ustalony V (dla uproszczenia zapomnijmy o pomocniczym wejściu).

Symulator M działa następująco:

1. Pobierz (G_1, G_2) . Uruchom symulowanie V (oznaczenie: V_S) : podaj mu (G_1, G_2) i zalosuj wartość losowej taśmy. Od tego momentu symuluj V .
2. Wybierz losowe $\tau \in \{1, 2\}$. Spermuj losowo G_τ . Niech $G' = \pi(G_\tau)$ będzie wynikiem tej permutacji. Wyślij G' do V_S .
3. Niech σ będzie odpowiedzią V_S .

Jeśli $\sigma \neq \tau$, to zwróć \perp .

W przeciwnym przypadku wyślij π do V_S i zwróć to co V_S zwraca.

Co by było gdybyśmy brali ustalone τ zamiast losować?

Symulator M ma działać dla dowolnego V_S , więc w szczególności ma działać również dla takiego V_S , które w odpowiedzi zawsze zwraca $\sigma = 1$. Gdyby M zawsze wybierało $\tau = 2$, otrzymywalibyśmy za każdym razem \perp .

Dlaczego to działa?

Nieformalnie:

G' , które V_S dostaje w kroku 2 ma identyczny rozkład, co G' w rzeczywistym protokole: jest po prostu losowym grafem izomorficznym z G_2 (tu korzystamy z faktu, że $G_1 \cong G_2$). Zatem zachowanie V_S jest takie samo jak V aż do początku kroku 3. Jeśli $\sigma = \tau$, to odpowiedź wysłana przez M do V_S jest taka sama, jak otrzymana przez prawdziwego V od P . Zatem ostateczny wynik działania V ma identyczny rozkład z wynikiem działania M (równy wynikowi działania V_S), *pod warunkiem, że M nie zwraca \perp .*

Pozostaje udowodnić, że \perp jest zwracane z prawdopodobieństwem co najwyżej $1/2$. Dla dowolnego G' niech $p(G')$ będzie prawdopodobieństwem, że po otrzymaniu G' , V_S odpowie 1. Wobec tego prawdopodobieństwo, że M zwróci \perp wynosi:

$$P[G' = G] \cdot \sum_G (P[V_S \text{ zwraca } 1 \text{ i } \tau = 2 \mid G' = G] + P[V_S \text{ zwraca } 2 \text{ i } \tau = 1 \mid G' = G])$$

Ponieważ to co zwraca V jest niezależne od τ jeśli znamy G' (a τ ma rozkład jednostajny), to

$$\frac{1}{n!} \sum_G \frac{1}{2} (p(G') + (1 - p(G')))$$

co jest równe $1/2$. □

5 Główne twierdzenie

Udowodnimy takie twierdzenie:

Twierdzenie 4 $NP \subseteq CZK$ (o ile funkcje jednokierunkowe istnieją)

5.1 Zobowiązania bitowe

Jako narzędzia użyjemy protokołu *zobowiązania bitowego* (ang.: *bit commitment*).

W protokole biorą udział dwie strony: Wysyłający S (ang.: *sender*) i Odbierający R (ang.: *receiver*). Wysyłający pobiera na wejściu bit $b \in \{0, 1\}$.

Protokół ten składa się z 2 faz:

Zobowiązanie W tej fazie R otrzymuje od S jakiś ciąg bitów, który nie daje mu (prawie) żadnej informacji na temat b (to wymaganie nazywa się *tajność*).

Można na to patrzeć w ten sposób: S wkłada b do skrzynki, zamyka ją na klucz i wysyła tą skrzynkę do R .

Otwarcie zobowiązania W wyniku tej fazy R poznaje b' , przy czym (z przeważającym prawdopodobieństwem) musi zachodzić $b = b'$ (to wymaganie nazywa się *związaniem S z b*).

W tej fazie S wysyła do R klucz do skrzynki i R będzie teraz mógł ją otworzyć i poznać b .

Nie będziemy podawać formalnej definicji (znajduje się np. w [1] na stronie 159).

Protokoły zobowiązania bitowego istnieją o ile istnieją funkcje jednokierunkowe.

5.1.1 Przykład protokołu zobowiązania bitowego

Niech $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ będzie permutacją jednokierunkową (tzn. zachowuje długość wejścia i na każdym zbiorze $\{0, 1\}^n$ jest bijekcją).

Wiadomo, że dla każdej funkcji jednokierunkowej f istnieje tzw. hard-core predicate, czyli taki wydajnie obliczalny predykat $B : \{0, 1\}^* \rightarrow \{0, 1\}$, że obliczenie wartości dla $B(x)$ (z prawdopodobieństwem znacząco lepszym niż 0.5) na podstawie $f(x)$ jest trudne obliczeniowo.

Np. najbardziej znaczący bit x dla funkcji $EXP(p, g, x) = g^x \bmod p$ (patrz Wykład 1) jest predykatem hard-core.

Więcej na ten temat znajduje się np. w [2].

Protokół zobowiązania bitowego wygląda teraz tak: (n - parametr bezpieczeństwa, b - bit wysyłającego).

Zobowiązanie S : wylosuj $x \in \{0, 1\}^n$ i wyślij do R parę $(y, a) := (f(x), B(x) \oplus b)$.

Otwarcie zobowiązania S : wyślij x do R . Jeśli $f(x) = y$ to V zaakceptuj wartość $B(x) \oplus a$ jako b .

Tajność jest zapewniona, ponieważ R nie jest w stanie odczytać b po otrzymaniu takiej pary podczas zobowiązania.

Związanie jest wypełnione, bo S może przekonać R tylko wysyłając prawdziwe x . Inaczej $f(x)$ nie będzie równe y (z przeważającym prawdopodobieństwem).

5.2 Dowód Twierdzenia 4

Wystarczy pokazać dowód z wiedzą zerową dla dowolnego problemu NP-zupełnego.

Tym problemem będzie G3C — język 3-kolorowalnych grafów.

Graf $G = (W, E)$ jest *3-kolorowalny*, jeśli jego wierzchołki można pokolorować 3 kolorami w taki sposób, że żadna krawędź nie ma obydwu wierzchołków tego samego koloru, tzn. istnieje $\phi : W \rightarrow \{1, 2, 3\}$, takie że $\forall_{(u,v) \in E} \phi(u) \neq \phi(v)$.

$$\text{G3C} := \{G : G \text{ jest 3-kolorowalny}\}.$$

Zachodzi następujący fakt:

Fakt 5 *Problem G3C jest NP-zupełny.*

Konstruujemy następujący dowód z wiedzą zerową dla tego problemu. Poniższa procedura powtarzana jest k razy (wartość k ustalimy później). Jeśli za każdym razem V zaakceptuje zobowiązanie P to cały dowód kończy się sukcesem.

1. P i V pobierają na wejściu graf $G = (W, E)$. Gdzie $W = \{1, \dots, n\}$. Ponadto P pobiera na wejściu 3-kolorowanie ψ grafu G .¹
2. P : wybierz losową permutację $\pi : \{1, 2, 3\} \rightarrow \{1, 2, 3\}$. Dla każdego wierzchołka x niech $\phi(x) := \pi(\psi(x))$. Dla każdego $i \in W$ zobowiąż się do $\phi(i)$ wobec V .
3. V : Wybierz losową krawędź $(u, v) \in E$ i wyślij do P .
4. P : otwórz zobowiązania do $\phi(u)$ i $\phi(v)$.
5. V : Jeśli P nie otworzył zobowiązań poprawnie, albo $\phi(u) = \phi(v)$, to odrzuć. W przeciwnym przypadku wykonujemy następny krok.

Nie będziemy pokazywać formalnego dowodu poprawności (znajduje się np. w [1]).

Nieformalnie:

Pełność Jeśli $G \in \text{G3C}$, to V zawsze akceptuje.

Poprawność Jeśli graf G nie jest 3-kolorowalny, to musi istnieć taka krawędź (u, v) , że $\phi(u) = \phi(v)$, zatem V zaakceptuje z prawdopodobieństwem $1 - \frac{1}{|E|}$.

Po k krokach to prawdopodobieństwo wynosi już

$$p(k) = \left(1 - \frac{1}{|E|}\right)^k$$

Można łatwo policzyć, że dla $k = 1 - \frac{1}{|E|}$ i dla dostatecznie dużych n otrzymamy

$$p(k) \approx e^{-1} < \frac{1}{2}.$$

Zerowa wiedza Wynika z tego, że przy każdej iteracji losowo permutujemy kolory oraz z własności tajności zobowiązania bitowego.

Powyższa analiza zakłada, że protokół zobowiązania bitowego działa z zerowym prawdopodobieństwem błędu. Takie protokoły nie istnieją, więc w formalnej analizie trzeba by to uwzględnić.

¹Dzięki temu strategia P będzie implementowana w czasie wielomianowym.

5.3 Dyskusja

Twierdzenie 4 jest bardzo silnym narzędziem, zwłaszcza, że w jego dowodzie strategia P jest wydajna (pod warunkiem, że P zna świadka, którego w przeciwnym wypadku nie da się znaleźć w czasie wielomianowym).

Umożliwia ono dowodzenie przez daną osobę, że jest tym za kogo się podaje (np. że jest autorem jakiejś wiadomości)

5.3.1 Przykład zastosowania (za [1])

1. S wysyła do odbiorców P_1, P_2 zaszyfrowane wiadomości M_1, \dots, M_t .
2. niech e_i będzie kluczem publicznym P_i
3. niech $C_i := E_{e_i}(r_i, M_i)$ (gdzie r_i jest losowe).
4. S chce udowodnić V , że $M_1 = M_2$.

Oczywiście może to zrobić ujawniając r_1, r_2, M_1, M_2 , ale okazuje się, że nie musi tego robić.

Niech

$$L := \{(C_1, C_2) : \exists_{r_1, r_2, M} C_1 = E_{e_1}(r_1, M) \wedge C_2 = E_{e_2}(r_2, M)\}.$$

Oczywiście $L \in \text{NP}$. Zatem S może udowodnić, że $M_1 = M_2$ za pomocą dowodu z wiedzą zerową!

Literatura

- [1] Oded Goldreich. Zero-knowledge proof systems. Dostępne pod adresem <http://www.wisdom.weizmann.ac.il/~oded/frag.html>.
- [2] S. Goldwasser and M. Bellare. Lecture notes in cryptography. Dostępne pod adresem <http://www.cs.ucsd.edu/users/mihir/papers/gb.html>.