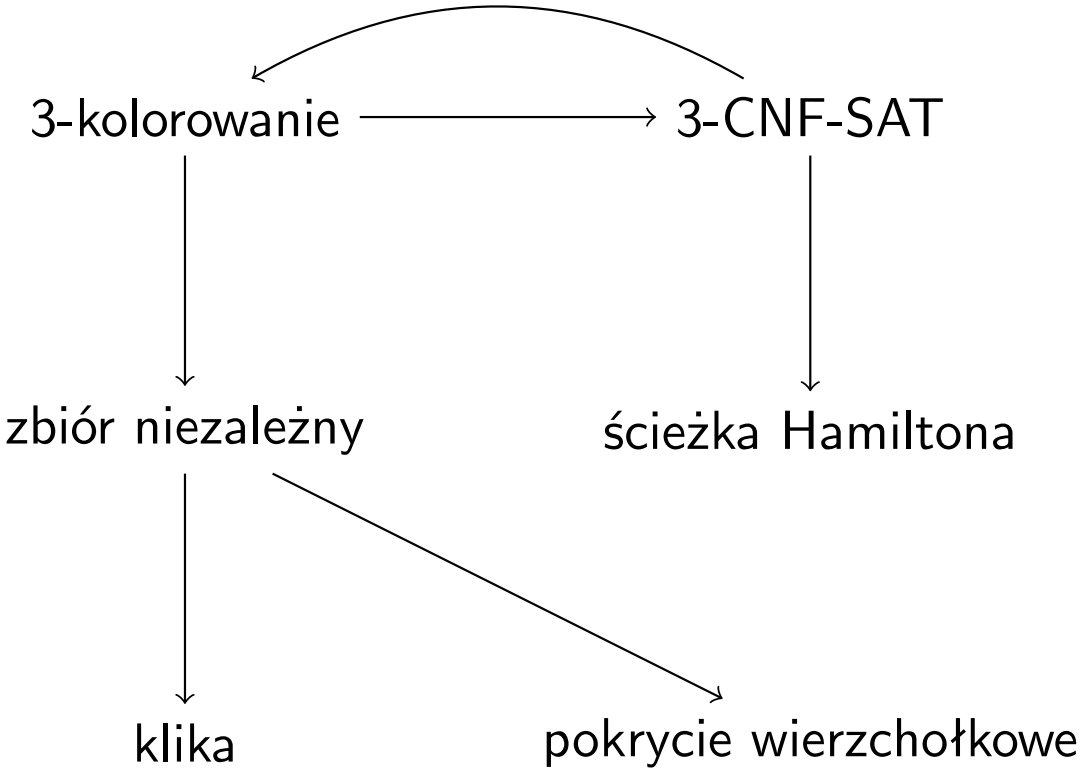
Klasa NP i NP-trudność

Marcin Pilipczuk

Algorytmika @ Uniwersytet Warszawski

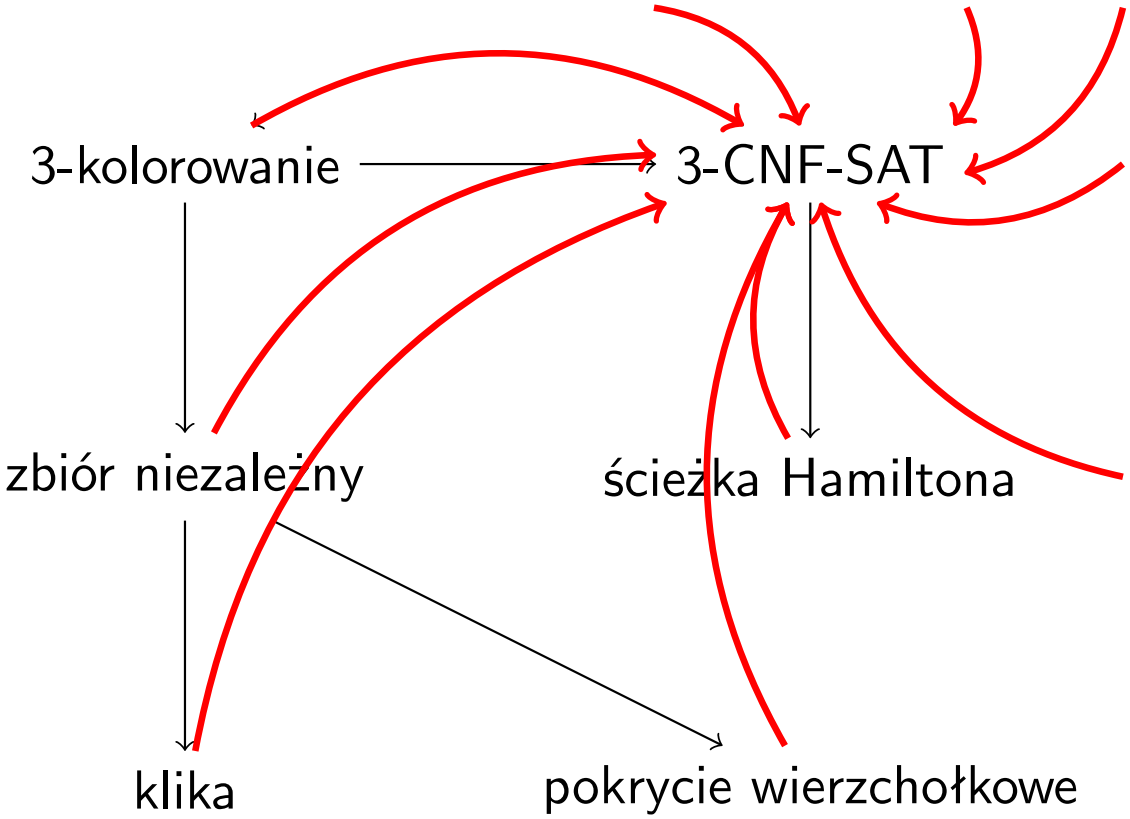
6 kwietnia 2020

W poprzednim odcinku



- Powyższe redukcje wielomianowe
- 3-kolorowanie trudne, bo tak

W poprzednim odcinku



- Powyższe redukcje wielomianowe
- 3-kolorowanie trudne, bo tak

Skąd prowadzą te strzałki?

Meta-twierdzenie

Z każdego problemu istnieje wielomianowa redukcja do spełnialności formuł 3-CNF-SAT / 3-kolorowania.

- Chcemy meta-twierdzenie o redukcji.

Skąd prowadzą te strzałki?

Meta-twierdzenie

Z każdego problemu istnieje wielomianowa redukcja do spełnialności formuł 3-CNF-SAT / 3-kolorowania.

- Chcemy meta-twierdzenie o redukcji.
- Nie da rady dla każdego problemu.

Skąd prowadzą te strzałki?

Meta-twierdzenie

Z każdego problemu typu X istnieje wielomianowa redukcja do spełnialności formuł 3-CNF-SAT / 3-kolorowania.

- Chcemy meta-twierdzenie o redukcji.
- Nie da rady dla każdego problemu.
- To jak określić jakąś naturalną klasę problemów, dla których się da?

Przykłady

3-kolorowanie

Wejście: graf nieskierowany G, $n := |V(G)|$
Wyjście: Czy istnieje 3-kolorowanie G, tj. funkcja $f : V(G) \to \{red, green, blue\}$ taka, że dla każdej $uv \in E(G)$ mamy $f(u) \neq f(v)$?

Uogólniony hex

Wejście: Graf G, dwa wierzchołki $s, t \in V(G)$ pomalowane na niebiesko.
Zasady: Dwóch graczy na przemian koloruje po jednym wierzchołku na niebiesko i czerwono. Niebieski gracz wygrywa, jeśli połączy s i t, a czerwony, jeśli temu przeszkodzi.
Wyjście: Kto ma strategię wygrywającą?

Halting problem

Wejście: Program komputerowy.
Wyjście: Czy ten program kończy się po skończonej liczbie kroków?

Przykłady c.d.

3-**kolorowanie**	$f : V(G) \to \{red, green, blue\}$
3-**CNF-SAT**	$f : \{x_1, \ldots, x_n\} \to \{\top, \bot\}$
zbiór niezależny	$A \subseteq V(G),\ E(G[A]) = \emptyset$
klika	$A \subseteq V(G),\ E(G[A]) = \binom{A}{2}$
pokrycie wierzchołkowe	$B \subseteq V(G),\ E(G - B) = \emptyset$
ścieżka Hamiltona	ścieżka $s = v_1, v_2, \ldots, v_n = t$

Przykłady c.d.

3-**kolorowanie**	$f : V(G) \to \{red, green, blue\}$
3-**CNF-SAT**	$f : \{x_1, \ldots, x_n\} \to \{\top, \bot\}$
zbiór niezależny	$A \subseteq V(G)$, $E(G[A]) = \emptyset$
klika	$A \subseteq V(G)$, $E(G[A]) = \binom{A}{2}$
pokrycie wierzchołkowe	$B \subseteq V(G)$, $E(G - B) = \emptyset$
ścieżka Hamiltona	ścieżka $s = v_1, v_2, \ldots, v_n = t$

Wszędzie łatwo uzasadnić, że odpowiedź jest TAK.

Przykłady c.d.

3-**kolorowanie**	$f : V(G) \to \{red, green, blue\}$
3-**CNF-SAT**	$f : \{x_1, \ldots, x_n\} \to \{\top, \bot\}$
zbiór niezależny	$A \subseteq V(G)$, $E(G[A]) = \emptyset$
klika	$A \subseteq V(G)$, $E(G[A]) = \binom{A}{2}$
pokrycie wierzchołkowe	$B \subseteq V(G)$, $E(G - B) = \emptyset$
ścieżka Hamiltona	ścieżka $s = v_1, v_2, \ldots, v_n = t$

Wszędzie łatwo uzasadnić, że odpowiedź jest TAK.

Ale niekoniecznie łatwo uzasadnić, że odpowiedź jest NIE.

Co formalizujemy?

Chcemy objąć deficją:

- Różne problemy: grafowe, tekstowe, logiczne, ...

Co formalizujemy?

Chcemy objąć deficją:

- Różne problemy: grafowe, tekstowe, logiczne, ...
- Co to n, wielkość instancji.

Co formalizujemy?

Chcemy objąć deficją:

- Różne problemy: grafowe, tekstowe, logiczne, ...
- Co to n, wielkość instancji.
- Co to jest dowód, że odpowiedź jest TAK.

Co formalizujemy?

Chcemy objąć deficją:

- Różne problemy: grafowe, tekstowe, logiczne, ...
- Co to n, wielkość instancji.
- Co to jest dowód, że odpowiedź jest TAK.
- Co to znaczy, że łatwo uzasadnić.

Formalizujemy problem

- Egzemplarz to słowo $x \in \{0,1\}^*$.

Formalizujemy problem

- Egzemplarz to słowo $x \in \{0,1\}^*$.
- Wielkość egzemplarza, $|x|$, to liczba znaków słowa x.

Formalizujemy problem

- Egzemplarz to słowo $x \in \{0,1\}^*$.
- Wielkość egzemplarza, $|x|$, to liczba znaków słowa x.
- Język to podzbiór $L \subseteq \{0,1\}^*$.
  - $x \in L \Leftrightarrow x$ ma odpowiedź TAK.
  - $x \notin L \Leftrightarrow x$ ma odpowiedź NIE.

Formalizujemy problem

- Egzemplarz to słowo $x \in \{0,1\}^*$.
- Wielkość egzemplarza, $|x|$, to liczba znaków słowa x.
- Język to podzbiór $L \subseteq \{0,1\}^*$.
  - $x \in L \Leftrightarrow x$ ma odpowiedź TAK.
  - $x \notin L \Leftrightarrow x$ ma odpowiedź NIE.
- Algorytm $\mathcal{A}$ rozstrzyga L w czasie f jeśli dla każdego $x \in \{0,1\}^*$, wykonanie $\mathcal{A}(x)$ poprawnie stwierdza czy $x \in L$ w czasie nie większym niż $f(|x|)$.

Formalizujemy problem

- Egzemplarz to słowo $x \in \{0,1\}^*$.
- Wielkość egzemplarza, $|x|$, to liczba znaków słowa x.
- Język to podzbiór $L \subseteq \{0,1\}^*$.
  - $x \in L \Leftrightarrow x$ ma odpowiedź TAK.
  - $x \notin L \Leftrightarrow x$ ma odpowiedź NIE.
- Algorytm $\mathcal{A}$ rozstrzyga L w czasie f jeśli dla każdego $x \in \{0,1\}^*$, wykonanie $\mathcal{A}(x)$ poprawnie stwierdza czy $x \in L$ w czasie nie większym niż $f(|x|)$.
- Język L jest rozstrzygalny w czasie wielomianowym jeśli istnieje algorytm $\mathcal{A}$ i wielomian f, że $\mathcal{A}$ rozstrzyga L w czasie f.

3-kolorowanie formalnie

A jak to się ma do 3-kolorowania?

- Wejście to graf. Trzeba ustalić jakieś kodowanie grafów w $\{0,1\}^*$.
  - $x \in L \Leftrightarrow x$ koduje graf, który jest 3-kolorowalny.
  - Słowa, które nie odpowiadają kodowaniom grafów, mają odpowiedź NIE. Powinno być trywialne.

3-kolorowanie formalnie

A jak to się ma do 3-kolorowania?

- Wejście to graf. Trzeba ustalić jakieś kodowanie grafów w $\{0,1\}^*$.
  - $x \in L \Leftrightarrow x$ koduje graf, który jest 3-kolorowalny.
  - Słowa, które nie odpowiadają kodowaniom grafów, mają odpowiedź NIE. Powinno być trywialne.
- Różne kodowania dają różne wielkości $|x|$!
  - Listy sąsiadów: $|x| = \mathcal{O}(n + m \log n)$.
  - Macierz sąsiedztwa: $|x| = \mathcal{O}(n^2)$.

3-kolorowanie formalnie

A jak to się ma do 3-kolorowania?

- Wejście to graf. Trzeba ustalić jakieś kodowanie grafów w $\{0,1\}^*$.
  - $x \in L \Leftrightarrow x$ koduje graf, który jest 3-kolorowalny.
  - Słowa, które nie odpowiadają kodowaniom grafów, mają odpowiedź NIE. Powinno być trywialne.
- Różne kodowania dają różne wielkości $|x|$!
  - Listy sąsiadów: $|x| = \mathcal{O}(n + m \log n)$.
  - Macierz sąsiedztwa: $|x| = \mathcal{O}(n^2)$.
- Nie ma znaczenia dla pytania, czy 3-kolorowanie jest rozwiązywalny w czasie wielomianowym.

3-kolorowanie formalnie

A jak to się ma do 3-kolorowania?

- Wejście to graf. Trzeba ustalić jakieś kodowanie grafów w $\{0,1\}^*$.
  - $x \in L \Leftrightarrow x$ koduje graf, który jest 3-kolorowalny.
  - Słowa, które nie odpowiadają kodowaniom grafów, mają odpowiedź NIE. Powinno być trywialne.
- Różne kodowania dają różne wielkości $|x|$!
  - Listy sąsiadów: $|x| = \mathcal{O}(n + m \log n)$.
  - Macierz sąsiedztwa: $|x| = \mathcal{O}(n^2)$.
- Nie ma znaczenia dla pytania, czy 3-kolorowanie jest rozwiązywalny w czasie wielomianowym.
- Na potrzeby trudności:
  - Grafy kodujemy jako macierze sąsiedztwa.
  - Formuły to listy klauzul, a klauzula to lista literałów.

Redukcja wielomianowa

Redukcja wielomianowa

Redukcja wielomianowa między językami L_1 i L_2 to algorytm $\mathcal{A}$ dla którego istnieje wielomian f taki, że, mając na wejściu słowo $x \in \{0,1\}^*$, oblicza słowo $\mathcal{A}(x)$ takie, że:

- $x \in L_1 \Leftrightarrow \mathcal{A}(x) \in L_2$;
- $|\mathcal{A}(x)| \leqslant f(|x|)$.
- Algorytm $\mathcal{A}(x)$ działa w czasie co najwyżej $f(|x|)$.

- L_1 to problem, który chce rozwiązać Bob, a L_2 to problem, w który umie Alicja.

Redukcja wielomianowa

Redukcja wielomianowa

Redukcja wielomianowa między językami L_1 i L_2 to algorytm $\mathcal{A}$ dla którego istnieje wielomian f taki, że, mając na wejściu słowo $x \in \{0,1\}^*$, oblicza słowo $\mathcal{A}(x)$ takie, że:

- $x \in L_1 \Leftrightarrow \mathcal{A}(x) \in L_2$;
- $|\mathcal{A}(x)| \leqslant f(|x|)$.
- Algorytm $\mathcal{A}(x)$ działa w czasie co najwyżej $f(|x|)$.

- L_1 to problem, który chce rozwiązać Bob, a L_2 to problem, w który umie Alicja.
- Powyższa definicja obejmuje jedno pytanie do Alicji.

Redukcja wielomianowa

Redukcja wielomianowa

Redukcja wielomianowa między językami L_1 i L_2 to algorytm $\mathcal{A}$ dla którego istnieje wielomian f taki, że, mając na wejściu słowo $x \in \{0,1\}^*$, oblicza słowo $\mathcal{A}(x)$ takie, że:

- $x \in L_1 \Leftrightarrow \mathcal{A}(x) \in L_2$;
- $|\mathcal{A}(x)| \leqslant f(|x|)$.
- Algorytm $\mathcal{A}(x)$ działa w czasie co najwyżej $f(|x|)$.

- L_1 to problem, który chce rozwiązać Bob, a L_2 to problem, w który umie Alicja.
- Powyższa definicja obejmuje jedno pytanie do Alicji.
- I której odpowiedź jest odpowiedzią na problem Boba.

Redukcja wielomianowa

Redukcja wielomianowa

Redukcja wielomianowa między językami L_1 i L_2 to algorytm $\mathcal{A}$ dla którego istnieje wielomian f taki, że, mając na wejściu słowo $x \in \{0,1\}^*$, oblicza słowo $\mathcal{A}(x)$ takie, że:

- $x \in L_1 \Leftrightarrow \mathcal{A}(x) \in L_2$;
- $|\mathcal{A}(x)| \leqslant f(|x|)$.
- Algorytm $\mathcal{A}(x)$ działa w czasie co najwyżej $f(|x|)$.

- L_1 to problem, który chce rozwiązać Bob, a L_2 to problem, w który umie Alicja.
- Powyższa definicja obejmuje jedno pytanie do Alicji.
- I której odpowiedź jest odpowiedzią na problem Boba.
- Dodaje do zeszłego tygodnia: trzeba rozkodować instancję Boba i zakodować instancję dla Alicji.

Świadkowie: przykłady

3-**kolorowanie**	$f : V(G) \to \{red, green, blue\}$
3-**CNF-SAT**	$f : \{x_1, \ldots, x_n\} \to \{\top, \bot\}$
zbiór niezależny	$A \subseteq V(G)$, $E(G[A]) = \emptyset$
klika	$A \subseteq V(G)$, $E(G[A]) = \binom{A}{2}$
pokrycie wierzchołkowe	$B \subseteq V(G)$, $E(G - B) = \emptyset$
ścieżka Hamiltona	ścieżka $s = v_1, v_2, \ldots, v_n = t$

Wszędzie łatwo uzasadnić, że odpowiedź jest TAK.

Ale niekoniecznie łatwo uzasadnić, że odpowiedź jest NIE.

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,
    - istnieje $y \in \{0,1\}^*$ długości co najwyżej $f(|x|)$ taki, że

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,
    - istnieje $y \in \{0,1\}^*$ długości co najwyżej $f(|x|)$ taki, że
    - $\mathcal{B}(x, y)$ mówi TAK;

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,
    - istnieje $y \in \{0,1\}^*$ długości co najwyżej $f(|x|)$ taki, że
    - $\mathcal{B}(x, y)$ mówi TAK;
  - a dla każdego NIE-egzemplarza $x \notin L$,

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,
    - istnieje $y \in \{0,1\}^*$ długości co najwyżej $f(|x|)$ taki, że
    - $\mathcal{B}(x, y)$ mówi TAK;
  - a dla każdego NIE-egzemplarza $x \notin L$,
    - i dla każdego świadka $y \in \{0,1\}^*$,

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,
    - istnieje $y \in \{0,1\}^*$ długości co najwyżej $f(|x|)$ taki, że
    - $\mathcal{B}(x, y)$ mówi TAK;
  - a dla każdego NIE-egzemplarza $x \notin L$,
    - i dla każdego świadka $y \in \{0,1\}^*$,
    - $\mathcal{B}(x, y)$ mówi NIE.

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,
    - istnieje $y \in \{0,1\}^*$ długości co najwyżej $f(|x|)$ taki, że
    - $\mathcal{B}(x, y)$ mówi TAK;
  - a dla każdego NIE-egzemplarza $x \notin L$,
    - i dla każdego świadka $y \in \{0,1\}^*$,
    - $\mathcal{B}(x, y)$ mówi NIE.

3-kolorowanie

$f : V(G) \to \{red, green, blue\}$
$\mathcal{B}$ sprawdza, czy f to 3-kolorowanie

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,
    - istnieje $y \in \{0,1\}^*$ długości co najwyżej $f(|x|)$ taki, że
    - $\mathcal{B}(x, y)$ mówi TAK;
  - a dla każdego NIE-egzemplarza $x \notin L$,
    - i dla każdego świadka $y \in \{0,1\}^*$,
    - $\mathcal{B}(x, y)$ mówi NIE.

3-**CNF-SAT**

$f : \{x_1, \ldots, x_n\} \to \{\top, \bot\}$
$\mathcal{B}$ sprawdza spełnienie klauzul

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,
    - istnieje $y \in \{0,1\}^*$ długości co najwyżej $f(|x|)$ taki, że
    - $\mathcal{B}(x, y)$ mówi TAK;
  - a dla każdego NIE-egzemplarza $x \notin L$,
    - i dla każdego świadka $y \in \{0,1\}^*$,
    - $\mathcal{B}(x, y)$ mówi NIE.

zbiór niezależny

$A \subseteq V(G)$, $E(G[A]) = \emptyset$
$\mathcal{B}$ sprawdza, czy $G[A]$ niezależny

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,
    - istnieje $y \in \{0,1\}^*$ długości co najwyżej $f(|x|)$ taki, że
    - $\mathcal{B}(x, y)$ mówi TAK;
  - a dla każdego NIE-egzemplarza $x \notin L$,
    - i dla każdego świadka $y \in \{0,1\}^*$,
    - $\mathcal{B}(x, y)$ mówi NIE.

klika

$A \subseteq V(G)$, $E(G[A]) = \binom{A}{2}$
$\mathcal{B}$ sprawdza, czy $G[A]$ to klika

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,
    - istnieje $y \in \{0,1\}^*$ długości co najwyżej $f(|x|)$ taki, że
    - $\mathcal{B}(x, y)$ mówi TAK;
  - a dla każdego NIE-egzemplarza $x \notin L$,
    - i dla każdego świadka $y \in \{0,1\}^*$,
    - $\mathcal{B}(x, y)$ mówi NIE.

pokrycie wierzchołkowe

$B \subseteq V(G)$, $E(G - B) = \emptyset$
$\mathcal{B}$ sprawdza, czy $G - B$ niezależny

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,
    - istnieje $y \in \{0,1\}^*$ długości co najwyżej $f(|x|)$ taki, że
    - $\mathcal{B}(x, y)$ mówi TAK;
  - a dla każdego NIE-egzemplarza $x \notin L$,
    - i dla każdego świadka $y \in \{0,1\}^*$,
    - $\mathcal{B}(x, y)$ mówi NIE.

ścieżka Hamiltona

ścieżka $s = v_1, v_2, \ldots, v_n = t$
$\mathcal{B}$ sprawdza, czy to ścieżka Hamiltona

Świadkowie

- Definiujemy klasę NP języków.
  - **Intuicja**: te języki, dla których łatwo uzasadnić, że odpowiedź jest tak.
- Język L jest w NP jeśli:
  - Istnieje wielomianowy algorytm $\mathcal{B}$ i wielomian f takie, że
  - dla każdego TAK-egzemplarza $x \in L$,
    - istnieje $y \in \{0,1\}^*$ długości co najwyżej $f(|x|)$ taki, że
    - $\mathcal{B}(x, y)$ mówi TAK;
  - a dla każdego NIE-egzemplarza $x \notin L$,
    - i dla każdego świadka $y \in \{0,1\}^*$,
    - $\mathcal{B}(x, y)$ mówi NIE.

doskonałe skojarzenie

$M \subseteq E(G)$
$\mathcal{B}$ sprawdza, czy M to skojarzenie wielkości $|V(G)|/2$

Klasy złożoności P i NP

$L \in P \Leftrightarrow$ istnieje algorytm wielomianowy rozstrzygający L.

$L \in NP \Leftrightarrow$ istnieje wielomianowy algorytm weryfikujący dla L.

Klasy złożoności P i NP

$L \in P \Leftrightarrow$ istnieje algorytm wielomianowy rozstrzygający L.

$L \in NP \Leftrightarrow$ istnieje wielomianowy algorytm weryfikujący dla L.

Twierdzenie

$P \subseteq NP$

Klasy złożoności P i NP

$L \in P \Leftrightarrow$ istnieje algorytm wielomianowy rozstrzygający L.

$L \in NP \Leftrightarrow$ istnieje wielomianowy algorytm weryfikujący dla L.

Twierdzenie

$P \subseteq NP$

Dowód.
Niech $\mathcal{B}(x, y) = \mathcal{A}(x)$.
Czyli olewamy świadka, rozstrzygamy czy $x \in L$ i raportujemy.

Klasy złożoności P i NP

$L \in P \Leftrightarrow$ istnieje algorytm wielomianowy rozstrzygający L.

$L \in NP \Leftrightarrow$ istnieje wielomianowy algorytm weryfikujący dla L.

Twierdzenie

$P \subseteq NP$

Dowód.
Niech $\mathcal{B}(x, y) = \mathcal{A}(x)$.
Czyli olewamy świadka, rozstrzygamy czy $x \in L$ i raportujemy.

Dla każdego $x \in L$, $y = \epsilon$ to dobry świadek.

Klasy złożoności P i NP

$L \in P \Leftrightarrow$ istnieje algorytm wielomianowy rozstrzygający L.

$L \in NP \Leftrightarrow$ istnieje wielomianowy algorytm weryfikujący dla L.

Twierdzenie

$P \subseteq NP$

Dowód.
Niech $\mathcal{B}(x, y) = \mathcal{A}(x)$.
Czyli olewamy świadka, rozstrzygamy czy $x \in L$ i raportujemy.

Dla każdego $x \in L$, $y = \epsilon$ to dobry świadek.

doskonałe skojarzenie

świadek pusty
$\mathcal{B}$ sprawdza, czy G ma doskonałe skojarzenie

Świadkowie: przykłady

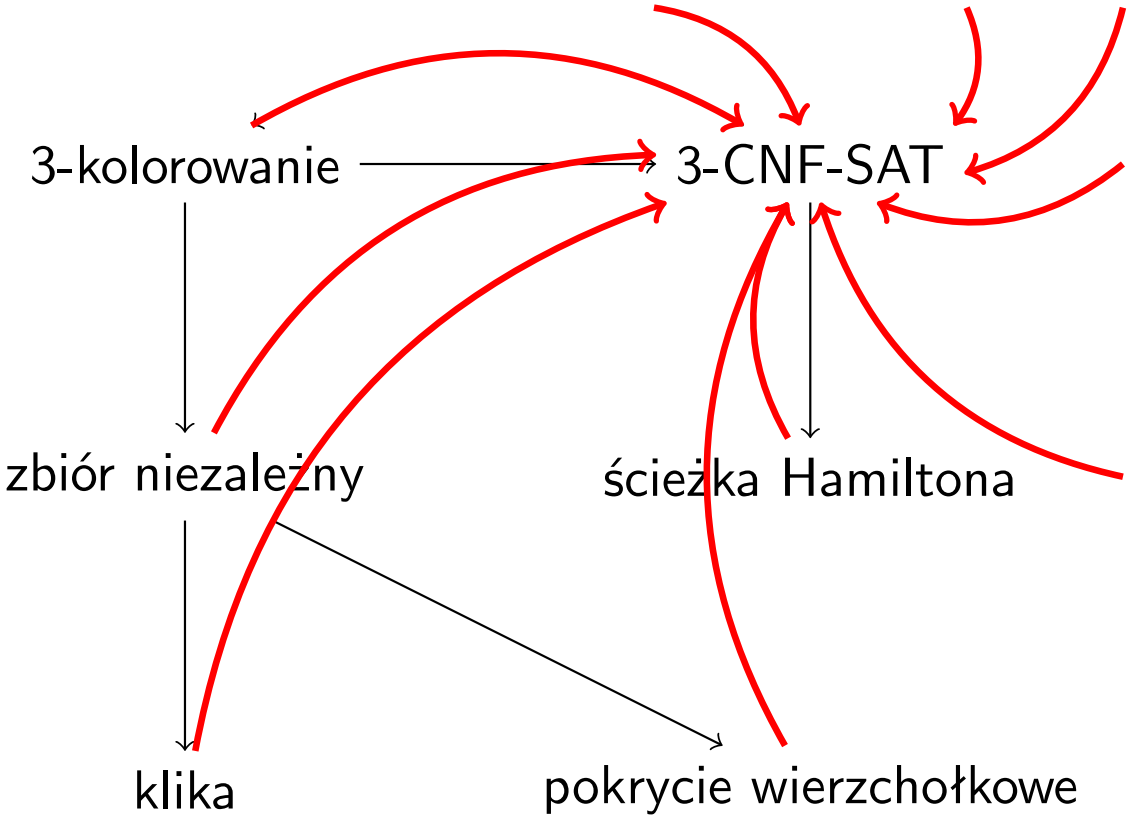
3-**kolorowanie**	$f : V(G) \to \{red, green, blue\}$ $\mathcal{B}$ sprawdza, czy f to 3-kolorowanie
3-**CNF-SAT**	$f : \{x_1, \ldots, x_n\} \to \{\top, \bot\}$ $\mathcal{B}$ sprawdza spełnienie klauzul
zbiór niezależny	$A \subseteq V(G)$, $E(G[A]) = \emptyset$ $\mathcal{B}$ sprawdza, czy $G[A]$ niezależny
klika	$A \subseteq V(G)$, $E(G[A]) = \binom{A}{2}$ $\mathcal{B}$ sprawdza, czy $G[A]$ to klika
pokrycie wierzchołkowe	$B \subseteq V(G)$, $E(G - B) = \emptyset$ $\mathcal{B}$ sprawdza, czy $G - B$ niezależny
ścieżka Hamiltona	ścieżka $s = v_1, v_2, \ldots, v_n = t$ $\mathcal{B}$ sprawdza, czy to ścieżka Hamiltona

Świadkowie: przykłady

3-**kolorowanie**	$f : V(G) \to \{red, green, blue\}$ $\mathcal{B}$ sprawdza, czy f to 3-kolorowanie
3-**CNF-SAT**	$f : \{x_1, \ldots, x_n\} \to \{\top, \bot\}$ $\mathcal{B}$ sprawdza spełnienie klauzul
zbiór niezależny	$A \subseteq V(G)$, $E(G[A]) = \emptyset$ $\mathcal{B}$ sprawdza, czy $G[A]$ niezależny
klika	$A \subseteq V(G)$, $E(G[A]) = \binom{A}{2}$ $\mathcal{B}$ sprawdza, czy $G[A]$ to klika
pokrycie wierzchołkowe	$B \subseteq V(G)$, $E(G - B) = \emptyset$ $\mathcal{B}$ sprawdza, czy $G - B$ niezależny
ścieżka Hamiltona	ścieżka $s = v_1, v_2, \ldots, v_n = t$ $\mathcal{B}$ sprawdza, czy to ścieżka Hamiltona

3-**kolorowanie**, 3-**CNF-SAT**, **zbiór niezależny**, **klika**, **pokrycie wierzchołkowe**, **ścieżka Hamiltona** $\in$ NP.

Twierdzenie Cooka-Levina



Twierdzenie (Cook-Levin, 1971-1973)

Dla każdego języka $L \in NP$, istnieje redukcja wielomianowa z L do (3-)CNF-SAT.

Twierdzenie Cooka-Levina

Twierdzenie (Cook-Levin, 1971-1973)

Dla każdego języka $L \in NP$, istnieje redukcja wielomianowa z L do 3-CNF-SAT.

Twierdzenie Cooka-Levina

Twierdzenie (Cook-Levin, 1971-1973)

Dla każdego języka $L \in NP$, istnieje redukcja wielomianowa z L do 3-CNF-SAT.

- L jest NP-trudny jeśli dla każdego $L' \in NP$ istnieje redukcja wielomianowa z L' do L.
- L jest NP-zupełny jeśli L jest NP-trudny i $L \in NP$.

Twierdzenie Cooka-Levina

Twierdzenie (Cook-Levin, 1971-1973)

Dla każdego języka $L \in NP$, istnieje redukcja wielomianowa z L do 3-CNF-SAT.

- L jest NP-trudny jeśli dla każdego $L' \in NP$ istnieje redukcja wielomianowa z L' do L.
- L jest NP-zupełny jeśli L jest NP-trudny i $L \in NP$.

Wniosek

3-**kolorowanie**, 3-**CNF-SAT**, **zbiór niezależny**, **klika**, **pokrycie wierzchołkowe**, **ścieżka Hamiltona** są NP-zupełne.

Twierdzenie Cooka-Levina

Twierdzenie (Cook-Levin, 1971-1973)

Dla każdego języka $L \in NP$, istnieje redukcja wielomianowa z L do 3-CNF-SAT.

- L jest NP-trudny jeśli dla każdego $L' \in NP$ istnieje redukcja wielomianowa z L' do L.
- L jest NP-zupełny jeśli L jest NP-trudny i $L \in NP$.

Wniosek

3-**kolorowanie**, 3-**CNF-SAT**, **zbiór niezależny**, **klika**, **pokrycie wierzchołkowe**, **ścieżka Hamiltona** są NP-zupełne.

By udowodnić NP-trudność L, wystarczy udowodnić redukcję z L' do L dla dowolnego NP-trudnego L'.

Twierdzenie Cooka-Levina

Twierdzenie (Cook-Levin, 1971-1973)

Dla każdego języka $L \in NP$, istnieje redukcja wielomianowa z L do 3-CNF-SAT.

- L jest NP-trudny jeśli dla każdego $L' \in NP$ istnieje redukcja wielomianowa z L' do L.
- L jest NP-zupełny jeśli L jest NP-trudny i $L \in NP$.

Wniosek

3-**kolorowanie**, 3-**CNF-SAT**, **zbiór niezależny**, **klika**, **pokrycie wierzchołkowe**, **ścieżka Hamiltona** są NP-zupełne.

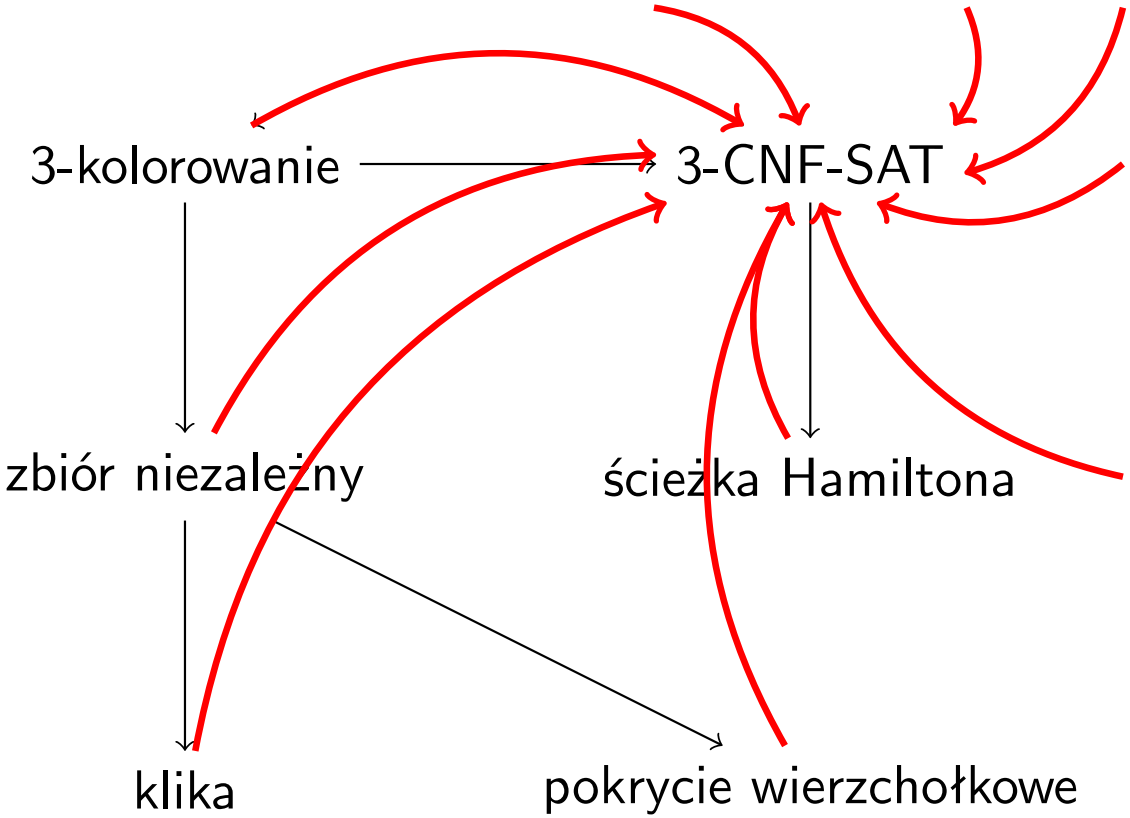
By udowodnić NP-trudność L, wystarczy udowodnić redukcję z L' do L dla dowolnego NP-trudnego L'.

Reszta wykładu: Dowód twierdzenia Cooka-Levina.

Dowód twierdzenia Cooka-Levina

na marginesie

Podsumowanie



Twierdzenie (Cook-Levin, 1971-1973)

Dla każdego języka $L \in NP$, istnieje redukcja wielomianowa z L do (3-)CNF-SAT.