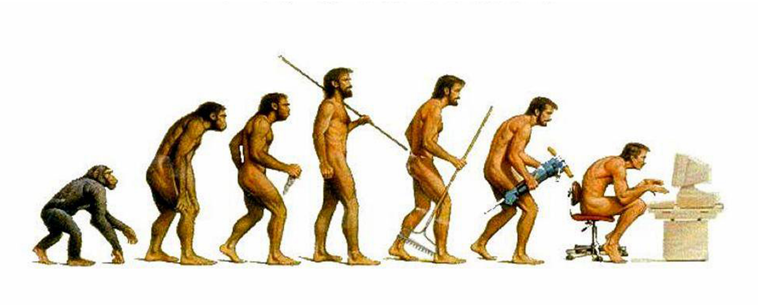


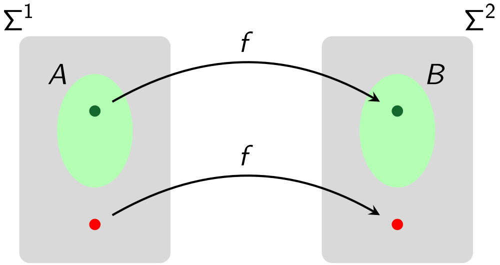
Teoria Obliczalności

Marcin Piątkowski

Wykład 11

Trudność i zupełność problemów obliczeniowych

Redukcja problemu obliczeniowego



Język $A \subseteq \Sigma_1^*$ jest **redukowalny** do języka $B \subseteq \Sigma_2^*$ jeśli istnieje totalna i obliczalna funkcja $f : \Sigma_1^* \longrightarrow \Sigma_2^*$ taka, że

$$\forall_{x \in \Sigma_1^*} \; x \in A \iff f(x) \in B$$

Efektywna redukcja

Funkcja $f : \Sigma_1^* \to \Sigma_2^*$ jest **obliczalna w czasie wielomianowym**, jeśli istnieje maszyna Turinga M działająca **w czasie wielomianowym**, która dla dowolnego słowa $w \in \Sigma_1^*$ zatrzymuje się zwracając wartość $f(w)$.

Język $A \subseteq \Sigma_1^*$ jest **wielomianowo redukowalny** do języka $B \subseteq \Sigma_2^*$, jeśli istnieje funkcja $f : \Sigma^* \to \Sigma^*$ **obliczalna w czasie wielomianowym** taka, że

$$\forall_{w \in \mathbf{\Sigma_1^*}} \; w \in A \iff f(w) \in B.$$

Efektywna redukcja

Funkcja $f : \Sigma_1^* \to \Sigma_2^*$ jest obliczalna **w pamięci logarytmicznej** jeśli istnieje maszyna Turinga M wykorzystująca co najwyżej **logarytmiczną** liczbę komórek taśm roboczych, która dla dowolnego słowa $w \in \Sigma_1^*$ zatrzymuje się zwracając wartość $f(w)$.

Język $A \subseteq \Sigma_1^*$ jest redukowalny do języka $B \subseteq \Sigma_2^*$ **w pamięci logarytmicznej**, jeśli istnieje funkcja $f : \Sigma_1^* \to \Sigma_2^*$ **obliczalna w pamięci logarytmicznej** taka, że

$$\forall_{w \in \Sigma_1^*}\ w \in A \iff f(w) \in B.$$

Efektywna redukcja

Funkcja $f : \Sigma_1^* \to \Sigma_2^*$ jest obliczalna **w pamięci logarytm**…
jeśli istnieje maszyna Turinga M wykorzystuja…
logarytmiczną liczbę komórek taśm robo…
słowa $w \in \Sigma_1^*$ zatrzymuje się zw…

!

Maszyna używająca pamięci rozmiaru $O(\log n)$ może mieć co najwyżej $c^{O(\log n)}$ różnych konfiguracji ($c \in \mathbb{N}$).

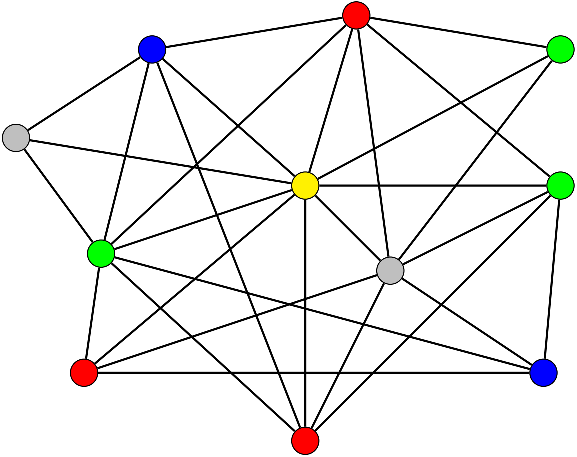
Redukcja w pamięci logarytmicznej $\Longrightarrow$ redukcja w czasie wielomianowym

… języka $B \subseteq \Sigma_2^*$ **w pamięci**
…eje funkcja $f : \Sigma_1^* \to \Sigma_2^*$ **obliczalna**
…**ytmicznej** taka, że

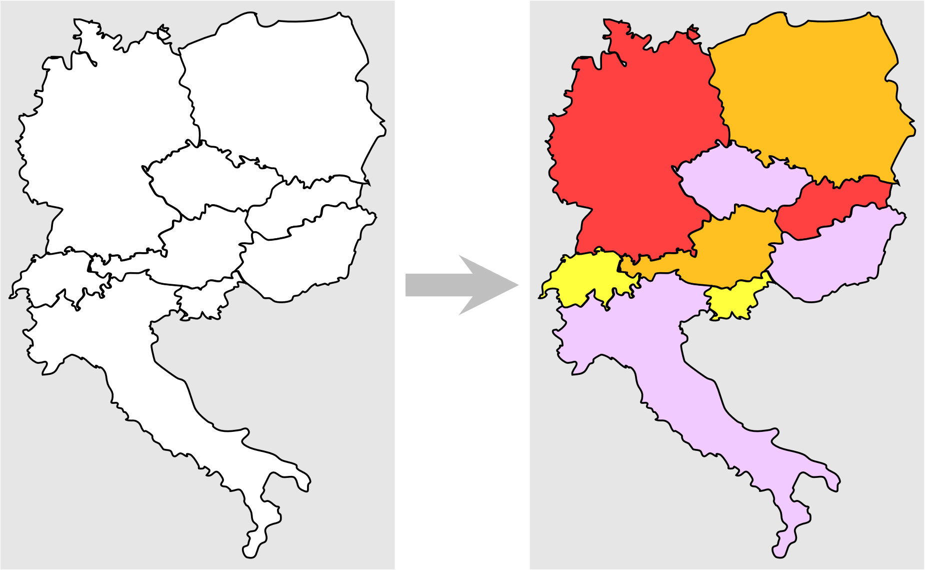
$$\forall_{w \in \Sigma_1^*}\ w \in A \iff f(w) \in B.$$

Kolorowanie grafu

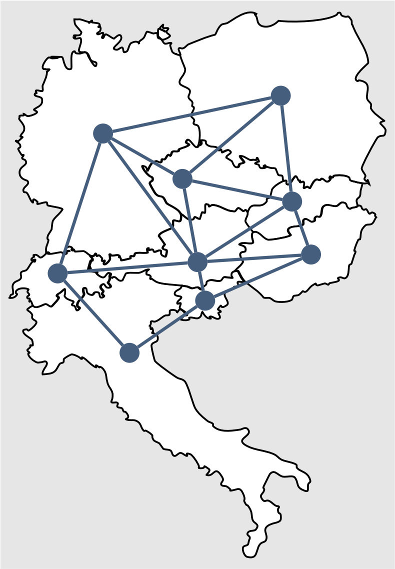
Wejście: Graf G oraz liczba $k > 0$.
Pytanie: Czy G można „pokolorować" za pomocą k kolorów?

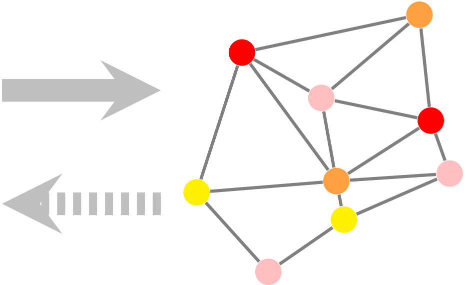


Kolorowanie mapy

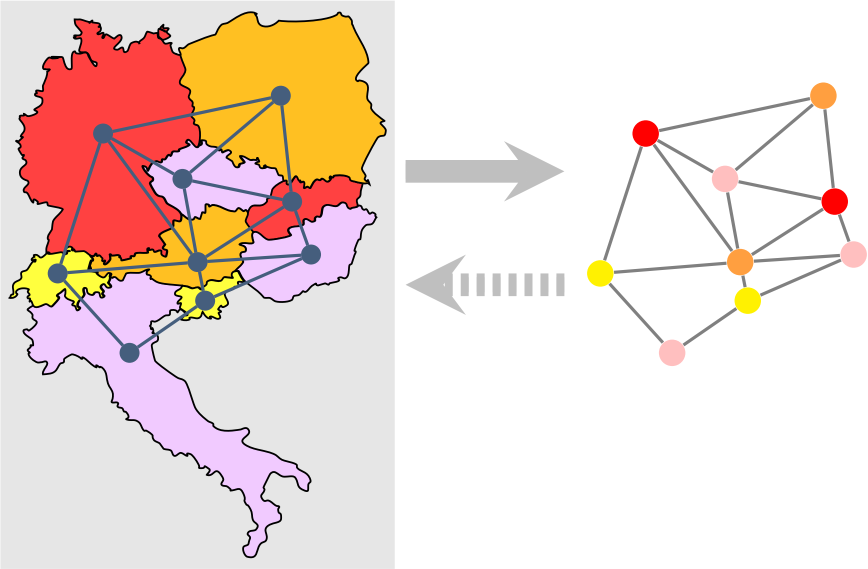


Redukcja problemów

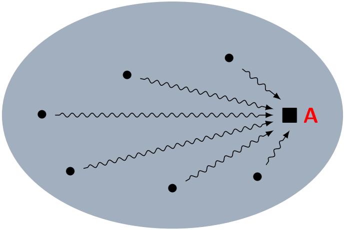




Redukcja problemów



Trudność i zupełność



Problem A nazywamy **C-trudnym** jeśli dla dowolnego problemu $B \in \mathrm{C}$ istnieje **efektywna** redukcja do problemu A

Problem A nazywamy **C-zupełnym** jeśli:

1. $A \in \mathrm{C}$
2. A jest C-trudny

Przykład

$$\mathrm{PATH} = \Big\{(G, v_1, v_2) : \text{ w grafie skierowanym } G \text{ istnieje ścieżka } v_1 \rightsquigarrow v_2\Big\}$$

Twierdzenie

Problem PATH jest NL-zupełny

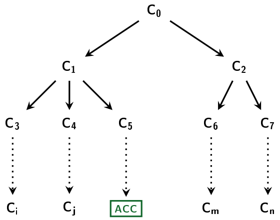
Przykład

- $\mathrm{PATH} \in \mathrm{NL}$

Przykład

- PATH $\in$ NL

- **Redukcja: A $\rightsquigarrow$ PATH**
  - $A \in NL$ – rozstrzygany przez M_A
  - Drzewo możliwych obliczeń $M_A \Longrightarrow$ graf skierowany G_A
  - M_A akceptuje wejście $\Longleftrightarrow$ G_A zawiera ścieżkę $c_0 \rightsquigarrow c_{ACC}$



Przykład

- PATH $\in$ NL

- **Redukcja: A $\rightsquigarrow$ PATH**
  - ☞ $A \in NL$ – rozstrzygany przez M_A
  - ☞ Drzewo możliwych obliczeń $M_A \Longrightarrow$ graf skierowany G_A
  - ☞ M_A akceptuje wejście $\Longleftrightarrow$ G_A zawiera ścieżkę $c_0 \rightsquigarrow c_{ACC}$

- **Złożonośc redukcji**
  - ☞ Maszyna M_G generująca kolejne konfiguracje M_A
  - ☞ Na wyjściu drukowane tylko poprawne pary $c_1 \rightarrow c_j$
  - ☞ Pojedyncza konfiguracja $M_A \Longrightarrow$ rozmiar $O(\log n)$

Klasa NP

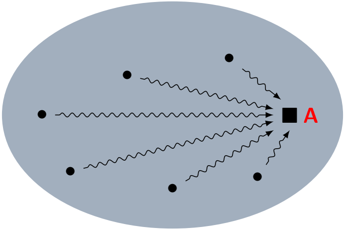
$$\text{HAMPATH} = \Big\{(G, v_1, v_2) : G \text{ zawiera ścieżkę Hamiltona } v_1 \rightsquigarrow v_2\Big\}$$

- Znalezienie rozwiązania $\Longrightarrow$ niedeterministyczny czas wielomianowy
- Znalezienie rozwiązania $\Longrightarrow$ deterministyczny czas wykładniczy
- Weryfikacja rozwiązania $\Longrightarrow$ deterministyczny czas wielomianowy

$$\overline{\text{HAMPATH}} = \Big\{(G, v_1, v_2) : G \text{ nie zawiera ścieżki Hamiltona } v_1 \rightsquigarrow v_2\Big\}$$

- Znalezienie rozwiązania $\Longrightarrow$ deterministyczny czas wykładniczy
- Weryfikacja rozwiązania $\Longrightarrow$ deterministyczny czas wykładniczy

NP-zupełność

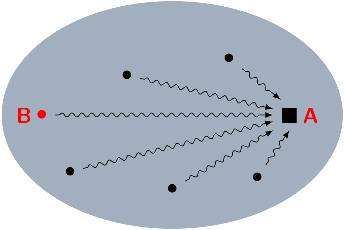


Rozwiązanie w czasie deterministycznym wielomianowym dla dowolnego problemu NP-zupełnego

⇓

Rozwiązanie w czasie deterministycznym wielomianowym dla wszystkich problemów z klasy NP

NP-zupełność



Brak rozwiązania w czasie deterministycznym wielomianowym dla dowolnego problemu NP-zupełnego

Brak rozwiązania w czasie deterministycznym wielomianowym dla wszystkich problemów NP-zupełnych

Relacje między klasami złożoności obliczeniowej

Przypadki ogólne

- $\mathbf{DTIME}(f(n)) \subseteq \mathbf{DSPACE}(f(n))$
  $\mathbf{NTIME}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$

Liczba użytych komórek taśm roboczych nie przekroczy liczby kroków obliczeń

Przypadki ogólne

- $\mathbf{DTIME}(f(n)) \subseteq \mathbf{DSPACE}(f(n))$

  $\mathbf{NTIME}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$

- $\mathbf{NTIME}(f(n)) \subseteq \mathbf{DTIME}(c^{f(n)})$

  Każda ścieżka w drzewie obliczeń maszyny niedeterministycznej ma długość co najwyżej $f(n)$. Symulacja jej działania przez maszynę deterministyczną wymaga co najwyżej $c^{f(n)}$ kroków.

Przypadki ogólne

- $\mathbf{DTIME}(f(n)) \subseteq \mathbf{DSPACE}(f(n))$

  $\mathbf{NTIME}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$

- $\mathbf{NTIME}(f(n)) \subseteq \mathbf{DTIME}(c^{f(n)})$

- $\mathbf{DSPACE}(f(n)) \subseteq \mathbf{DTIME}(c^{f(n)})$

Użycie $f(n)$ komórek taśmy, umożliwia zapisanie co najwyżej $c^{f(n)}$ różnych konfiguracji ($c \in \boldsymbol{N}$)

Przypadki ogólne

- $\mathbf{DTIME}(f(n)) \subseteq \mathbf{DSPACE}(f(n))$

  $\mathbf{NTIME}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$

- $\mathbf{NTIME}(f(n)) \subseteq \mathbf{DTIME}(c^{f(n)})$

- $\mathbf{DSPACE}(f(n)) \subseteq \mathbf{DTIME}(c^{f(n)})$

- $\mathbf{NSPACE}(f(n)) \subseteq \mathbf{DSPACE}(f^2(n))$

  Twierdzenie Savitcha

Przypadki szczególne

- $\mathbf{P} \subsetneq \mathbf{EXPTIME}$
- $\mathbf{NP} \subsetneq \mathbf{NEXPTIME}$

Problem rozstrzygalny w czasie wielomianowym jest rozstrzygalny również w czasie wykładniczym

Przypadki szczególne

- $\mathbf{P} \subsetneq \mathbf{EXPTIME}$
  $\mathbf{NP} \subsetneq \mathbf{NEXPTIME}$
- $\mathbf{PSPACE} = \mathbf{NPSPACE}$

Twierdzenie Savitcha

Przypadki szczególne

- $\mathbf{P} \subsetneq \mathbf{EXPTIME}$
  $\mathbf{NP} \subsetneq \mathbf{NEXPTIME}$
- $\mathbf{PSPACE} = \mathbf{NPSPACE}$
- $\mathbf{P} \subseteq \mathbf{PSPACE}$
  $\mathbf{NP} \subseteq \mathbf{NPSPACE}$

Liczba użytych komórek taśmy nie przekroczy liczby wykonanych kroków obliczeń

Ponadto, $\mathbf{NP} \subseteq \mathbf{PSPACE}$ (tw. Savitcha)

Przypadki szczególne

- $\mathsf{P} \subsetneq \mathsf{EXPTIME}$

  $\mathsf{NP} \subsetneq \mathsf{NEXPTIME}$

- $\mathsf{PSPACE} = \mathsf{NPSPACE}$

- $\mathsf{P} \subseteq \mathsf{PSPACE}$

  $\mathsf{NP} \subseteq \mathsf{NPSPACE}$

- $\mathsf{PSPACE} \subseteq \mathsf{EXPTIME}$

Maszyna działająca w pamięci wielomianowej może mieć co najwyżej wykładniczą liczbę konfiguracji

Przypadki szczególne

- $\mathsf{P} \subsetneq \mathsf{EXPTIME}$
  $\mathsf{NP} \subsetneq \mathsf{NEXPTIME}$
- $\mathsf{PSPACE} = \mathsf{NPSPACE}$
- $\mathsf{P} \subseteq \mathsf{PSPACE}$
  $\mathsf{NP} \subseteq \mathsf{NPSPACE}$
- $\mathsf{PSPACE} \subseteq \mathsf{EXPTIME}$
- $\mathsf{L} \subseteq \mathsf{NL}$

Problem rozstrzygalny w deterministycznej pamięci logarytmicznej jest rozstrzygalny również w niedeterministycznej pamięci logarytmicznej

Przypadki szczególne

- $\mathbf{P} \subsetneq \mathbf{EXPTIME}$
  $\mathbf{NP} \subsetneq \mathbf{NEXPTIME}$
- $\mathbf{PSPACE} = \mathbf{NPSPACE}$
- $\mathbf{P} \subseteq \mathbf{PSPACE}$
  $\mathbf{NP} \subseteq \mathbf{NPSPACE}$
- $\mathbf{PSPACE} \subseteq \mathbf{EXPTIME}$
- $\mathbf{L} \subseteq \mathbf{NL}$
- $\mathbf{P} \subseteq \mathbf{NP}$

Problem rozstrzygalny w deterministycznym czasie wielomianowym jest rozstrzygalny również w niedeterministycznym czasie wielomianowym

Przypadki szczególne

- $\mathsf{P} \subsetneq \mathsf{EXPTIME}$

  $\mathsf{NP} \subsetneq \mathsf{NEXPTIME}$

- $\mathsf{PSPACE} = \mathsf{NPSPACE}$

- $\mathsf{P} \subseteq \mathsf{PSPACE}$

  $\mathsf{NP} \subseteq \mathsf{NPSPACE}$

- $\mathsf{PSPACE} \subseteq \mathsf{EXPTIME}$

- $\mathsf{L} \subseteq \mathsf{NL}$

- $\mathsf{P} \subseteq \mathsf{NP}$

- $\mathsf{NL} \subseteq \mathsf{P}$

Maszyna niedeterministyczna działająca w pamięci logarytmicznej może mieć co najwyżej wielomianową liczbę konfiguracji

$$\mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{EXPTIME} \subseteq \mathsf{NEXPTIME}$$

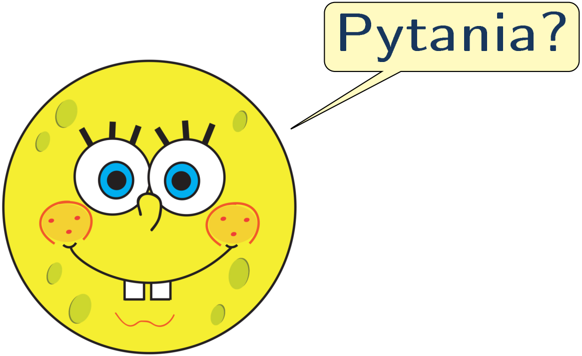
$$\mathsf{P} \overset{?}{\subseteq} \mathsf{NP} \overset{?}{\subseteq} \mathsf{EXPTIME} \subseteq \mathsf{NEXPTIME}$$

$$\mathsf{P} \neq \mathsf{EXPTIME}$$

$$\mathrm{P} \overset{?}{\subseteq} \mathrm{NP} \overset{?}{\subseteq} \mathrm{EXPTIME} \subseteq \mathrm{NEXPTIME}$$

$$\mathrm{P} \neq \mathrm{EXPTIME}$$

$$\mathrm{P} = \mathrm{NP} \longrightarrow \mathrm{EXTIME} = \mathrm{NEXPTIME}$$


Pytania?