Złożoność obliczeniowa algorytmów

Redukcje i zupełność

Kordian A. Smoliński

Wydział Fizyki i Informatyki Stosowanej

2024/2025

Redukcje i zupełność

Treść wykładu

Redukcje

Definicja

Język $L_1 \subset A_1^*$ jest redukowalny do języka $L_2 \subset A_2^*$, $L_1 \preccurlyeq L_2$, jeżeli istnieje maszyna Turinga M obliczająca funkcję $f\colon A_1^* \to A_2^*$ o złożoności pamięciowej $S(M, w) \in O(\log |w|)$ taka, że

$$\forall w \in L_1\colon f(w) \in L_2.$$

Redukcje

Definicja

Język $L_1 \subset A_1^*$ jest redukowalny do języka $L_2 \subset A_2^*$, $L_1 \preccurlyeq L_2$, jeżeli istnieje maszyna Turinga M obliczająca funkcję $f\colon A_1^* \to A_2^*$ o złożoności pamięciowej $S(M, w) \in O(\log |w|)$ taka, że

$$\forall w \in L_1\colon f(w) \in L_2.$$

Definicja

Funkcję f nazywamy redukcją L_1 do L_2.

Redukcje

Definicja

Język $L_1 \subset A_1^*$ jest redukowalny do języka $L_2 \subset A_2^*$, $L_1 \preccurlyeq L_2$, jeżeli istnieje maszyna Turinga M obliczająca funkcję $f\colon A_1^* \to A_2^*$ o złożoności pamięciowej $S(M, w) \in O(\log |w|)$ taka, że

$$\forall w \in L_1\colon f(w) \in L_2.$$

Definicja

Funkcję f nazywamy redukcją L_1 do L_2.

Interpretacja

Rozstrzyganie L_1 jest co najwyżej tak samo trudne jak rozstrzyganie L_2.

Redukcje

Fakt

Maszyna M oblicza redukcję f w czasie wielomianowym, tzn.

$$\exists k \in \mathbb{N}\ \forall w \in A_1^*\colon\ T(M, w) \in O(|w|^k).$$

Redukcje

Fakt

Maszyna M oblicza redukcję f w czasie wielomianowym, tzn.

$$\exists k \in \mathbb{N}\ \forall w \in A_1^*\colon\ T(M, w) \in O(|w|^k).$$

Dowód.

Dla słowa w M ma $O(|w|c^{\log|w|})$ możliwych konfiguracji (dla pewnego $c > 1$). Maszyna jest deterministyczna, i zatrzymuje się, więc podczas obliczenia żadna konfiguracja się nie powtarza. Zatem M może wykonać co najwyżej $O(|w|^k)$ kroków (dla pewnego $k \in \mathbb{N}$). $\square$

Redukcje

Twierdzenie

Jeżeli f_1 *jest redukcją* L_1 *do* L_2*, a* f_2 *redukcją* L_2 *do* L_3*, to* $f_2 \circ f_1$ *jest redukcją* L_1 *do* L_3*.*

Redukcje

Twierdzenie

Jeżeli f_1 jest redukcją L_1 do L_2, a f_2 redukcją L_2 do L_3, to $f_2 \circ f_1$ jest redukcją L_1 do L_3.

Dowód.

Nietrywialne jest udowodnienie, że $f_2 \circ f_1$ można obliczyć w pamięci logarytmicznej. Szczegóły w:

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ Uniwersytet Łódzki

Redukcje

Twierdzenie

Jeżeli f_1 jest redukcją L_1 do L_2, a f_2 redukcją L_2 do L_3, to $f_2 \circ f_1$ jest redukcją L_1 do L_3.

Dowód.

Nietrywialne jest udowodnienie, że $f_2 \circ f_1$ można obliczyć w pamięci logarytmicznej. Szczegóły w:

C. H. Papadimitriou,
Złożoność obliczeniowa,
Wydawnictwa Naukowo-Techniczne, Warszawa 2002.

Zupełność

Definicja

Niech $\mathcal{C}$ będzie klasą złożnoności. Język $L \in \mathcal{C}$ jest $\mathcal{C}$-zupełny, jeżeli $\forall L' \in \mathcal{C} : L' \preccurlyeq L$.

Zupełność

Definicja

Niech $\mathcal{C}$ będzie klasą złożnoności. Język $L \in \mathcal{C}$ jest $\mathcal{C}$-zupełny, jeżeli $\forall L' \in \mathcal{C} : L' \preccurlyeq L$.

Interpretacja

Język $\mathcal{C}$-zupełny jest najtrudniejszym do rozstrzygnięcia językiem w klasie złożoności $\mathcal{C}$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Zupełność

Definicja

Niech $\mathcal{C}$ będzie klasą złożnoności. Język $L \in \mathcal{C}$ jest $\mathcal{C}$-zupełny, jeżeli $\forall L' \in \mathcal{C}: L' \preccurlyeq L$.

Interpretacja

Język $\mathcal{C}$-zupełny jest najtrudniejszym do rozstrzygnięcia językiem w klasie złożoności $\mathcal{C}$.

Definicja

Klasa złożoności $\mathcal{C}$ jest zamknięta na redukcje, jeżeli $\forall L' \preccurlyeq L \in \mathcal{C}: L' \in \mathcal{C}$.

Zupełność

Fakt

Klasy **P**, **NP**, **L**, **NL**, **PSPACE**, **EXP** *są zamknięte na redukcje.*

Zupełność

Fakt

Klasy **P**, **NP**, **L**, **NL**, **PSPACE**, **EXP** *są zamknięte na redukcje.*

Twierdzenie

Jeżeli klasy C i C' są zamknięte na redukcje i istnieje język L zupełny dla obu klas, to $C = C'$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Zupełność

Fakt

Klasy **P**, **NP**, **L**, **NL**, **PSPACE**, **EXP** *są zamknięte na redukcje.*

Twierdzenie

Jeżeli klasy $\mathcal{C}$ i $\mathcal{C}'$ są zamknięte na redukcje i istnieje język L zupełny dla obu klas, to $\mathcal{C} = \mathcal{C}'$.

Dowód.

L jest zupełny w $\mathcal{C}$ więc $\forall L' \in \mathcal{C}\colon L' \preccurlyeq L \in \mathcal{C}'$. $\mathcal{C}'$ jest zamknięta na redukcje, więc $\forall L' \in \mathcal{C}\colon L' \in \mathcal{C}'$, czyli $\mathcal{C} \subseteq \mathcal{C}'$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ Uniwersytet Łódzki

Zupełność

Fakt

Klasy **P**, **NP**, **L**, **NL**, **PSPACE**, **EXP** *są zamknięte na redukcje.*

Twierdzenie

Jeżeli klasy $\mathcal{C}$ i $\mathcal{C}'$ są zamknięte na redukcje i istnieje język L zupełny dla obu klas, to $\mathcal{C} = \mathcal{C}'$.

Dowód.

L jest zupełny w $\mathcal{C}$ więc $\forall L' \in \mathcal{C}: L' \preccurlyeq L \in \mathcal{C}'$. $\mathcal{C}'$ jest zamknięta na redukcje, więc $\forall L' \in \mathcal{C}: L' \in \mathcal{C}'$, czyli $\mathcal{C} \subseteq \mathcal{C}'$. Podobnie dowodzimy, że $\mathcal{C}' \subseteq \mathcal{C}$, więc $\mathcal{C} = \mathcal{C}'$. □

Twierdzenie

$$\begin{aligned}\{0,1\}^* \supseteq L \in \mathbf{NP} \iff \\ \exists p(x) \wedge \exists L' \in \mathbf{P}\ \forall n \in \mathbb{N} \wedge \forall w \in \{0,1\}^n : \\ \left(w \in L \iff \exists v \in \{0,1\}^{p(n)} : w \| v \in L'\right).\end{aligned}$$

Twierdzenie

$$
\begin{aligned}
\{0,1\}^* \supseteq L \in \mathbf{NP} \iff& \\
&\exists p(x) \wedge \exists L' \in \mathbf{P}\ \forall n \in \mathbb{N} \wedge \forall w \in \{0,1\}^n : \\
&\qquad \left(w \in L \iff \exists v \in \{0,1\}^{p(n)} : w \| v \in L'\right).
\end{aligned}
$$

Interpretacja

Klasa **NP** składa się z języków L, dla których istnieje język L' z klasy **P** dla każdego słowa $w \in L$ o długości n istnieje dowód $w \| v \in L'$ o długości będącej wielomianem w n.

Przykład

Język NONPRIME to rozwinięcia binarne liczb złożonych, tzn. $n \in \mathbb{N} \setminus \{0, 1\} \wedge n \notin \mathbb{P}$.

Przykład

Język NONPRIME to rozwinięcia binarne liczb złożonych, tzn. $n \in \mathbb{N} \setminus \{0, 1\} \wedge n \notin \mathbb{P}$. NONPRIME $\in$ **NP**.

Przykład

Język NONPRIME to rozwinięcia binarne liczb złożonych, tzn. $n \in \mathbb{N} \setminus \{0, 1\} \wedge n \notin \mathbb{P}$. NONPRIME $\in$ **NP**.
Jeżeli znamy $1 < k < n$ takie, że $k|n$, to n jest złożona.

Przykład

Język NONPRIME to rozwinięcia binarne liczb złożonych, tzn. $n \in \mathbb{N} \setminus \{0, 1\} \wedge n \notin \mathbb{P}$. NONPRIME $\in$ **NP**.

Jeżeli znamy $1 < k < n$ takie, że $k|n$, to n jest złożona.
$\lfloor \log_2 k \rfloor + 1 \leqslant \lfloor \log_2 n \rfloor + 1$, więc rozwinięcie binarne k nie jest dłuższe od rozwinięcia binarnego n, czyli możemy przyjąć $p(x) = x$, czyli konkatenacja rozwinięć binarnych n i k jest wielomianowej długości w długości rozwinięcia n. Język tych konkatenacji należy do **P**, gdyż można go rozstrzygnąć w czasie wielomianowym w długości rozwinięcia n przeprowadzając algorytm dzielenia, który działa w czasie kwadratowym w długości rozwinięcia n.

Przykład

Język NONPRIME to rozwinięcia binarne liczb złożonych, tzn. $n \in \mathbb{N} \setminus \{0, 1\} \wedge n \notin \mathbb{P}$. NONPRIME $\in$ **NP**.
Jeżeli znamy $1 < k < n$ takie, że $k|n$, to n jest złożona.
$\lfloor \log_2 k \rfloor + 1 \leqslant \lfloor \log_2 n \rfloor + 1$, więc rozwinięcie binarne k nie jest dłuższe od rozwinięcia binarnego n, czyli możemy przyjąć $p(x) = x$, czyli konkatenacja rozwinięć binarnych n i k jest wielomianowej długości w długości rozwinięcia n. Język tych konkatenacji należy do **P**, gdyż można go rozstrzygnąć w czasie wielomianowym w długości rozwinięcia n przeprowadzając algorytm dzielenia, który działa w czasie kwadratowym w długości rozwinięcia n.

Uwaga

NONPRIME jest w **NP**, ale nie jest **NP**-zupełny.

Problem (SAT)

Czy dla danej formuły rachunku zdań istnieje takie wartościowanie zmiennych zdaniowych, żeby przyjmowała dla niego wartość ***prawda****?*

Problem (SAT)

*Czy dla danej formuły rachunku zdań istnieje takie wartościowanie zmiennych zdaniowych, żeby przyjmowała dla niego wartość **prawda**?*

Fakt (Cook–Lewin)

Problem SAT *jest* **NP***-zupełny.*

Problem (SAT)

Czy dla danej formuły rachunku zdań istnieje takie wartościowanie zmiennych zdaniowych, żeby przyjmowała dla niego wartość ***prawda****?*

Fakt (Cook–Lewin)

Problem SAT *jest* **NP***-zupełny.*

Dowód.

Problem (SAT)

*Czy dla danej formuły rachunku zdań istnieje takie wartościowanie zmiennych zdaniowych, żeby przyjmowała dla niego wartość **prawda**?*

Fakt (Cook–Lewin)

Problem SAT *jest* **NP***-zupełny.*

Dowód.

C. H. Papadimitriou,
Złożoność obliczeniowa,
Wydawnictwa Naukowo-Techniczne, Warszawa 2002.