Złożoność obliczeniowa algorytmów

Klasy złożoności obliczeniowej

Kordian A. Smoliński

Wydział Fizyki i Informatyki Stosowanej

2024/2025

Klasy złożoności obliczeniowej

Treść wykładu

1. Języki formalne
   - Języki a maszyna Turinga
   - Języki rekurencyjne i rekurencyjnie przeliczalne
2. Klasy złożoności
   - Liniowe przyśpieszanie i liniowa kompresja
   - Klasy złożoności dla asymptotycznych funkcji złożoności
   - Relacje pomiędzy klasami złożoności

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Języki formalne

Definicje

Alfabet A — dowolny skończony zbiór.

Języki formalne

Definicje

Alfabet A — dowolny skończony zbiór.

Symbol σ — dowolny element alfabetu, $\sigma \in A$.

Języki formalne

Definicje

Alfabet A — dowolny skończony zbiór.

Symbol σ — dowolny element alfabetu, $\sigma \in A$.

A^n — zbiór n-elementowych ciągów $\sigma_1\sigma_2 \ldots \sigma_n$ alfabetu A.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Języki formalne

Definicje

Alfabet A — dowolny skończony zbiór.

Symbol σ — dowolny element alfabetu, $\sigma \in A$.

A^n — zbiór n-elementowych ciągów $\sigma_1\sigma_2 \ldots \sigma_n$ alfabetu A.

Słowo w — dowolny skończony ciąg symboli alfabetu A.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Języki formalne

Definicje

Alfabet A — dowolny skończony zbiór.

Symbol σ — dowolny element alfabetu, $\sigma \in A$.

A^n — zbiór n-elementowych ciągów $\sigma_1\sigma_2 \ldots \sigma_n$ alfabetu A.

Słowo w — dowolny skończony ciąg symboli alfabetu A.

Słowo puste ϵ — 0-elementowy ciąg symboli dowolnego alfabetu ($\{\epsilon\} = A^0$).

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Języki formalne

Definicje

Alfabet A — dowolny skończony zbiór.

Symbol σ — dowolny element alfabetu, $\sigma \in A$.

A^n — zbiór n-elementowych ciągów $\sigma_1\sigma_2 \ldots \sigma_n$ alfabetu A.

Słowo w — dowolny skończony ciąg symboli alfabetu A.

Słowo puste ϵ — 0-elementowy ciąg symboli dowolnego alfabetu ($\{\epsilon\} = A^0$).

A^* — zbiór skończonych ciągów symboli alfabetu A,

$$A^* = \{\epsilon\} \cup A \cup A^2 \cup \cdots \cup A^n \cup \cdots$$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Języki formalne

Definicje

Alfabet A — dowolny skończony zbiór.

Symbol σ — dowolny element alfabetu, $\sigma \in A$.

A^n — zbiór n-elementowych ciągów $\sigma_1\sigma_2 \ldots \sigma_n$ alfabetu A.

Słowo w — dowolny skończony ciąg symboli alfabetu A.

Słowo puste ϵ — 0-elementowy ciąg symboli dowolnego alfabetu ($\{\epsilon\} = A^0$).

A^* — zbiór skończonych ciągów symboli alfabetu A,

$$A^* = \{\epsilon\} \cup A \cup A^2 \cup \cdots \cup A^n \cup \cdots$$

Uwaga

Słowem nie jest nieskończony ciąg symboli.

Języki formalne

Definicja

Język $L \subset A^*$ to dowolny zbiór skończonych ciągów symboli z alfabetu A.

Języki formalne

Definicja

Język $L \subset A^*$ to dowolny zbiór skończonych ciągów symboli z alfabetu A.

Przykłady

- zbiór słów z liter polskiego alfabetu i występujących w pewnym słowniku, np. oko $\in L$, krw $\notin L$;

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Języki formalne

Definicja

Język $L \subset A^*$ to dowolny zbiór skończonych ciągów symboli z alfabetu A.

Przykłady

- zbiór słów z liter polskiego alfabetu i występujących w pewnym słowniku, np. oko $\in L$, krw $\notin L$;
- zbiór słów złożonych z cyfr od 0 do 9 i reprezentujących liczbę pierwszą, np. $43 \in L$, $51 \notin L$;

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Języki formalne

Definicja

Język $L \subset A^*$ to dowolny zbiór skończonych ciągów symboli z alfabetu A.

Przykłady

- zbiór słów z liter polskiego alfabetu i występujących w pewnym słowniku, np. oko $\in L$, krw $\notin L$;
- zbiór słów złożonych z cyfr od 0 do 9 i reprezentujących liczbę pierwszą, np. $43 \in L$, $51 \notin L$;
- zbiór słów nad alfabetem $A = \{0, \ldots, 9, +, =\}$ reprezentujących prawidłowo zapisane działanie dodawania liczb naturalnych, np. $13 + 5 = 18 \in L$, $= 12 = 7+ \notin L$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Maszyna Turinga M o $n+1$ kolorach może obliczać funkcję na słowach w nad n-elementowym alfabetem A:

Maszyna Turinga M o $n+1$ kolorach może obliczać funkcję na słowach w nad n-elementowym alfabetem A:

- dowolny ciąg białych komórek taśmy reprezentuje słowo puste ϵ;

Maszyna Turinga M o $n+1$ kolorach może obliczać funkcję na słowach w nad n-elementowym alfabetem A:

- dowolny ciąg białych komórek taśmy reprezentuje słowo puste ϵ;
- każdemu symbolowi alfabetu A odpowiada któryś z pozostałych kolorów taśmy.

Maszyna Turinga M o $n+1$ kolorach może obliczać funkcję na słowach w nad n-elementowym alfabetem A:

- dowolny ciąg białych komórek taśmy reprezentuje słowo puste ϵ;
- każdemu symbolowi alfabetu A odpowiada któryś z pozostałych kolorów taśmy.

Definicja

$M(w)$ to słowo reprezentowane na taśmie, po tym jak M zatrzyma się na słowie $w \in A^*$.

Maszyna Turinga M o $n+1$ kolorach może obliczać funkcję na słowach w nad n-elementowym alfabetem A:

- dowolny ciąg białych komórek taśmy reprezentuje słowo puste ϵ;
- każdemu symbolowi alfabetu A odpowiada któryś z pozostałych kolorów taśmy.

Definicja

$M(w)$ to słowo reprezentowane na taśmie, po tym jak M zatrzyma się na słowie $w \in A^*$.

Jeżeli M nie zatrzymuje się na w, to $M(w) = \nearrow$.

Definicje

M akceptuje $w \iff M(w) = \epsilon$.

Definicje

M akceptuje $w \iff M(w) = \epsilon$.

M odrzuca $w \iff M(w) \neq \epsilon$.

Definicje

M akceptuje $w \iff M(w) = \epsilon$.

M odrzuca $w \iff M(w) \neq \epsilon$.

Definicja

Maszyna M rozpoznaje język L, jeżeli $\forall w \in L : M(w) = \epsilon$.

Definicje

M akceptuje $w \iff M(w) = \epsilon$.

M odrzuca $w \iff M(w) \neq \epsilon$.

Definicja

Maszyna M rozpoznaje język L, jeżeli $\forall w \in L \colon M(w) = \epsilon$.

Definicja

Maszyna M rozstrzyga język L, jeżeli
$(\forall w \in L \colon M(w) = \epsilon) \wedge (\forall w \notin L \colon M(w) \neq \epsilon)$.

Definicje

M akceptuje $w \iff M(w) = \epsilon$.

M odrzuca $w \iff M(w) \neq \epsilon$.

Definicja

Maszyna M rozpoznaje język L, jeżeli $\forall w \in L \colon M(w) = \epsilon$.

Definicja

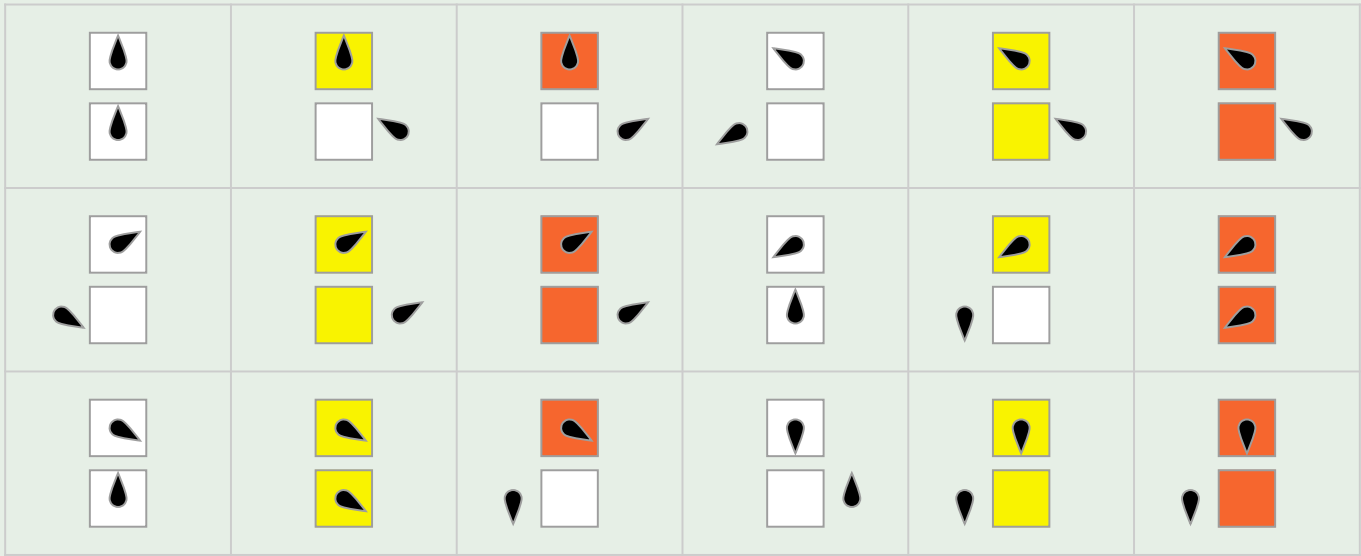
Maszyna M rozstrzyga język L, jeżeli
$(\forall w \in L \colon M(w) = \epsilon) \wedge (\forall w \notin L \colon M(w) \neq \epsilon)$.

Definicja

Niech $f : A^* \to A^*$. Maszyna M oblicza funkcję f, jeżeli
$\forall w \in A^* \colon M(w) = f(w)$.

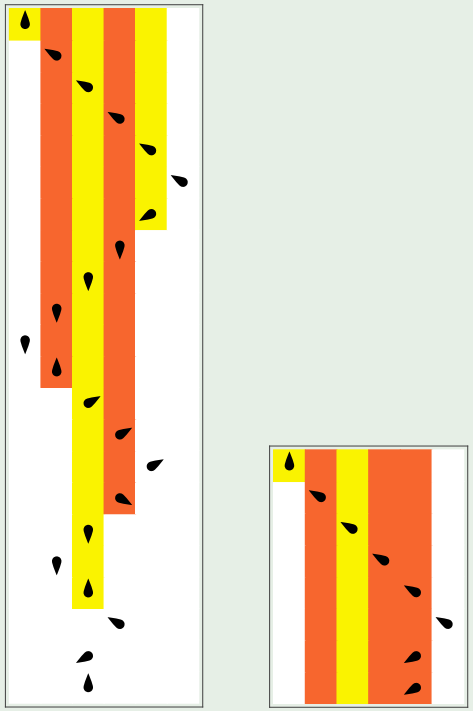
Przykład (palindromy)

$A = \{0, 1\}$, $w \in L \subset A^*$ $\iff$ w jest palindromem.
Maszyna Turinga rozstrzygająca czy w jest palindromem:



Rysunek: Maszyna Turinga dla palindromów

Przykład (palindromy)



Rysunek: Obliczenia maszyny Turinga dla palindromów

Definicja

Język L jest rekurencyjnie przeliczalny, jeżeli istnieje maszyna M, która rozpoznaje L.

Języki formalne

Języki rekurencyjne i rekurencyjnie przeliczalne

Definicja

Język L jest rekurencyjnie przeliczalny, jeżeli istnieje maszyna M, która rozpoznaje L.

Definicja

Język L jest rekurencyjny, jeżeli istnieje maszyna M, która rozstrzyga L.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Języki formalne

Języki rekurencyjne i rekurencyjnie przeliczalne

Definicja

Język L jest rekurencyjnie przeliczalny, jeżeli istnieje maszyna M, która rozpoznaje L.

Definicja

Język L jest rekurencyjny, jeżeli istnieje maszyna M, która rozstrzyga L.

Fakt

Każdy język rekurencyjny jest rekurencyjnie przeliczalny.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Języki formalne

Języki rekurencyjne i rekurencyjnie przeliczalne

Definicja

Język L jest rekurencyjnie przeliczalny, jeżeli istnieje maszyna M, która rozpoznaje L.

Definicja

Język L jest rekurencyjny, jeżeli istnieje maszyna M, która rozstrzyga L.

Fakt

Każdy język rekurencyjny jest rekurencyjnie przeliczalny.

Definicja

Funkcja $f : A^* \to A^*$ jest funkcją rekurencyjną, jeżeli istnieje maszyna M taka, że $\forall w \in A^* : M(w) = f(w)$.

Klasy złożoności

Definicja

Język L nad alfabetem A należy do klasy złożoności **TIME**$(f(n))$ wtedy i tylko wtedy, gdy istnieje maszyna M, która dla każdego $w \in A^*$, $|w| = n$ rozstrzyga czy $w \in L$ w czasie $f(n)$, tj. $T(M, w) = f(n)$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Klasy złożoności

Definicja

Język L nad alfabetem A należy do klasy złożoności **TIME**$(f(n))$ wtedy i tylko wtedy, gdy istnieje maszyna M, która dla każdego $w \in A^*$, $|w| = n$ rozstrzyga czy $w \in L$ w czasie $f(n)$, tj. $T(M, w) = f(n)$.

Przykład

Palindromy nad $A = \{0, 1\}$ należą do **TIME**$(\frac{(n+1)(n+2)}{2})$.

Klasy złożoności

Definicja

Język L nad alfabetem A należy do klasy złożoności **TIME**$(f(n))$ wtedy i tylko wtedy, gdy istnieje maszyna M, która dla każdego $w \in A^*$, $|w| = n$ rozstrzyga czy $w \in L$ w czasie $f(n)$, tj. $T(M, w) = f(n)$.

Przykład

Palindromy nad $A = \{0, 1\}$ należą do $\mathbf{TIME}(\frac{(n+1)(n+2)}{2})$.

Definicja

Język L nad alfabetem A należy do klasy złożoności **SPACE**$(f(n))$ wtedy i tylko wtedy, gdy istnieje maszyna M, która dla każdego $w \in A^*$, $|w| = n$ rozstrzyga czy $w \in L$ w pamięci $f(n)$, tj. $S(M, w) = f(n)$.

Klasy złożoności

Definicja

Język L nad alfabetem A należy do klasy złożoności **NTIME**$(f(n))$ wtedy i tylko wtedy, gdy istnieje niedeterministyczna maszyna M, która dla każdego $w \in A^*$, $|w| = n$ rozstrzyga czy $w \in L$ w czasie $f(n)$, tj. $T(M, w) = f(n)$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Klasy złożoności

Definicja

Język L nad alfabetem A należy do klasy złożoności **NTIME**$(f(n))$ wtedy i tylko wtedy, gdy istnieje niedeterministyczna maszyna M, która dla każdego $w \in A^*$, $|w| = n$ rozstrzyga czy $w \in L$ w czasie $f(n)$, tj. $T(M, w) = f(n)$.

Definicja

Język L nad alfabetem A należy do klasy złożoności **NSPACE**$(f(n))$ wtedy i tylko wtedy, gdy istnieje niedeterministyczna maszyna M, która dla każdego $w \in A^*$, $|w| = n$ rozstrzyga czy $w \in L$ w pamięci $f(n)$, tj. $S(M, w) = f(n)$.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Twierdzenie (liniowe przyśpieszanie)

$$L \in \mathbf{TIME}(f(n)) \implies \forall \epsilon > 0 \colon L \in \mathbf{TIME}(\epsilon f(n) + n + 2).$$

Twierdzenie (liniowe przyśpieszanie)

$$L \in \mathbf{TIME}(f(n)) \implies \forall \epsilon > 0 \colon L \in \mathbf{TIME}(\epsilon f(n) + n + 2).$$

Dowód.

Twierdzenie (liniowe przyśpieszanie)

$$L \in \mathbf{TIME}(f(n)) \implies \forall \epsilon > 0 \colon L \in \mathbf{TIME}(\epsilon f(n) + n + 2).$$

Dowód.

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

Twierdzenie (liniowe przyśpieszanie)

$$L \in \mathbf{TIME}(f(n)) \implies \forall \epsilon > 0 \colon L \in \mathbf{TIME}(\epsilon f(n) + n + 2).$$

Dowód.

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

Wniosek

Funkcja $f(n)$ w definicjach klas złożoności czasowej może być rozpatrywana z dokładnością do stałej.

Twierdzenie (liniowa kompresja)

$$L \in \mathbf{SPACE}(f(n)) \implies \forall \epsilon > 0 \colon L \in \mathbf{SPACE}(\epsilon f(n) + 2).$$

Twierdzenie (liniowa kompresja)

$$L \in \mathbf{SPACE}(f(n)) \implies \forall \epsilon > 0 \colon L \in \mathbf{SPACE}(\epsilon f(n) + 2).$$

Dowód.

Klasy złożoności

Liniowe przyśpieszanie i liniowa kompresja

Twierdzenie (liniowa kompresja)

$$L \in \mathbf{SPACE}(f(n)) \implies \forall \epsilon > 0 \colon L \in \mathbf{SPACE}(\epsilon f(n) + 2).$$

Dowód.

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

Klasy złożoności

Liniowe przyśpieszanie i liniowa kompresja

Twierdzenie (liniowa kompresja)

$$L \in \mathbf{SPACE}(f(n)) \implies \forall \epsilon > 0 \colon L \in \mathbf{SPACE}(\epsilon f(n) + 2).$$

Dowód.

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

Wniosek

Funkcja $f(n)$ w definicjach klas złożoności pamięciowej może być rozpatrywana z dokładnością do stałej.

Definicje

Określamy następujące klasy złożoności czasowej:

Definicje

Określamy następujące klasy złożoności czasowej:

$\mathbf{P} = \bigcup_{k>1} \mathbf{TIME}(n^k)$ języki rozstrzygalne w (deterministycznym) czasie wielomianowym;

Definicje

Określamy następujące klasy złożoności czasowej:

P $= \bigcup_{k>1} \mathbf{TIME}(n^k)$ języki rozstrzygalne w (deterministycznym) czasie wielomianowym;

NP $= \bigcup_{k>1} \mathbf{NTIME}(n^k)$ języki rozstrzygalne w niedeterministycznym czasie wielomianowym;

Definicje

Określamy następujące klasy złożoności czasowej:

P $= \bigcup_{k>1}$ **TIME**(n^k) języki rozstrzygalne w (deterministycznym) czasie wielomianowym;

NP $= \bigcup_{k>1}$ **NTIME**(n^k) języki rozstrzygalne w niedeterministycznym czasie wielomianowym;

EXP $= \bigcup_{k>1}$ **TIME**(2^{n^k}) języki rozstrzygalne w (deterministycznym) czasie wykładniczym;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Definicje

Określamy następujące klasy złożoności czasowej:

P $= \bigcup_{k>1} \mathbf{TIME}(n^k)$ języki rozstrzygalne w (deterministycznym) czasie wielomianowym;

NP $= \bigcup_{k>1} \mathbf{NTIME}(n^k)$ języki rozstrzygalne w niedeterministycznym czasie wielomianowym;

EXP $= \bigcup_{k>1} \mathbf{TIME}(2^{n^k})$ języki rozstrzygalne w (deterministycznym) czasie wykładniczym;

NEXP $= \bigcup_{k>1} \mathbf{NTIME}(2^{n^k})$ języki rozstrzygalne w niedeterministycznym czasie wykładniczym.

Definicje (klasy złożoności pamięciowej)

Definicje (klasy złożoności pamięciowej)

L $=$ **SPACE**$(\log n)$ języki rozstrzygalne w (deterministycznej) pamięci logarytmicznej;

Klasy złożoności

Klasy złożoności dla asymptotycznych funkcji złożoności

Definicje (klasy złożoności pamięciowej)

L $= \mathbf{SPACE}(\log n)$ języki rozstrzygalne w (deterministycznej) pamięci logarytmicznej;

NL $= \mathbf{NSPACE}(\log n)$ języki rozstrzygalne w niedeterministycznej pamięci logarytmicznej;

Definicje (klasy złożoności pamięciowej)

L $=$ **SPACE**$(\log n)$ języki rozstrzygalne w (deterministycznej) pamięci logarytmicznej;

NL $=$ **NSPACE**$(\log n)$ języki rozstrzygalne w niedeterministycznej pamięci logarytmicznej;

PSPACE $= \bigcup_{k>1}$ **SPACE**(n^k) języki rozstrzygalne w (deterministycznej) pamięci wielomianowej;

Definicje (klasy złożoności pamięciowej)

L $=$ **SPACE**$(\log n)$ języki rozstrzygalne w (deterministycznej) pamięci logarytmicznej;

NL $=$ **NSPACE**$(\log n)$ języki rozstrzygalne w niedeterministycznej pamięci logarytmicznej;

PSPACE $= \bigcup_{k>1}$ **SPACE**(n^k) języki rozstrzygalne w (deterministycznej) pamięci wielomianowej;

NPSPACE $= \bigcup_{k>1}$ **NSPACE**(n^k) języki rozstrzygalne w niedeterministycznej pamięci wielomianowej;

Klasy złożoności

Klasy złożoności dla asymptotycznych funkcji złożoności

Definicje (klasy złożoności pamięciowej)

L $=$ **SPACE**$(\log n)$ języki rozstrzygalne w (deterministycznej) pamięci logarytmicznej;

NL $=$ **NSPACE**$(\log n)$ języki rozstrzygalne w niedeterministycznej pamięci logarytmicznej;

PSPACE $= \bigcup_{k>1}$ **SPACE**(n^k) języki rozstrzygalne w (deterministycznej) pamięci wielomianowej;

NPSPACE $= \bigcup_{k>1}$ **NSPACE**(n^k) języki rozstrzygalne w niedeterministycznej pamięci wielomianowej;

EXPSPACE $= \bigcup_{k>1}$ **SPACE**(2^{n^k}) języi rozstrzygalne w (deterministycznej) pamięci wykładniczej;

Klasy złożoności

Klasy złożoności dla asymptotycznych funkcji złożoności

Definicje (klasy złożoności pamięciowej)

L $=$ **SPACE**$(\log n)$ języki rozstrzygalne w (deterministycznej) pamięci logarytmicznej;

NL $=$ **NSPACE**$(\log n)$ języki rozstrzygalne w niedeterministycznej pamięci logarytmicznej;

PSPACE $= \bigcup_{k>1}$ **SPACE**(n^k) języki rozstrzygalne w (deterministycznej) pamięci wielomianowej;

NPSPACE $= \bigcup_{k>1}$ **NSPACE**(n^k) języki rozstrzygalne w niedeterministycznej pamięci wielomianowej;

EXPSPACE $= \bigcup_{k>1}$ **SPACE**(2^{n^k}) języi rozstrzygalne w (deterministycznej) pamięci wykładniczej;

NEXPSPACE $= \bigcup_{k>1}$ **NSPACE**(2^{n^k}) języki rozstrzygalne w niedeterministycznej pamięci wykładniczej.

Klasy złożoności

Relacje pomiędzy klasami złożoności

$\mathbf{TIME}(f(n)) \subseteq \mathbf{NTIME}(f(n))$ (każda maszyna deterministyczna jest maszyną niedeterministyczną);

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Klasy złożoności

Relacje pomiędzy klasami złożoności

$\mathbf{TIME}(f(n)) \subseteq \mathbf{NTIME}(f(n))$ (każda maszyna deterministyczna jest maszyną niedeterministyczną);

$\mathbf{SPACE}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$ (j.w.);

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Klasy złożoności

Relacje pomiędzy klasami złożoności

$\mathbf{TIME}(f(n)) \subseteq \mathbf{NTIME}(f(n))$ (każda maszyna deterministyczna jest maszyną niedeterministyczną);

$\mathbf{SPACE}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$ (j.w.);

$\mathbf{TIME}(f(n)) \subseteq \mathbf{SPACE}(f(n))$ (maszyna nie może zapisać więcej komórek niż czas jej działania);

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Klasy złożoności

Relacje pomiędzy klasami złożoności

$\mathbf{TIME}(f(n)) \subseteq \mathbf{NTIME}(f(n))$ (każda maszyna deterministyczna jest maszyną niedeterministyczną);

$\mathbf{SPACE}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$ (j.w.);

$\mathbf{TIME}(f(n)) \subseteq \mathbf{SPACE}(f(n))$ (maszyna nie może zapisać więcej komórek niż czas jej działania);

$\mathbf{NTIME}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$ (j.w.);

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Klasy złożoności

Relacje pomiędzy klasami złożoności

- $\mathbf{TIME}(f(n)) \subseteq \mathbf{NTIME}(f(n))$ (każda maszyna deterministyczna jest maszyną niedeterministyczną);
- $\mathbf{SPACE}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$ (j.w.);
- $\mathbf{TIME}(f(n)) \subseteq \mathbf{SPACE}(f(n))$ (maszyna nie może zapisać więcej komórek niż czas jej działania);
- $\mathbf{NTIME}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$ (j.w.);
- $\exists c > 1 : \mathbf{NTIME}(f(n)) \subseteq \mathbf{TIME}(c^{f(n)})$ (symulacja maszyny niedeterministycznej na maszynie deterministycznej);

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Klasy złożoności

Relacje pomiędzy klasami złożoności

$\mathbf{TIME}(f(n)) \subseteq \mathbf{NTIME}(f(n))$ (każda maszyna deterministyczna jest maszyną niedeterministyczną);

$\mathbf{SPACE}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$ (j.w.);

$\mathbf{TIME}(f(n)) \subseteq \mathbf{SPACE}(f(n))$ (maszyna nie może zapisać więcej komórek niż czas jej działania);

$\mathbf{NTIME}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$ (j.w.);

$\exists c > 1 \colon \mathbf{NTIME}(f(n)) \subseteq \mathbf{TIME}(c^{f(n)})$ (symulacja maszyny niedeterministycznej na maszynie deterministycznej);

$\exists c > 1 \colon \mathbf{NSPACE}(f(n)) \subseteq \mathbf{TIME}(c^{f(n)})$ (j.w.);

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Klasy złożoności

Relacje pomiędzy klasami złożoności

- $\mathbf{TIME}(f(n)) \subseteq \mathbf{NTIME}(f(n))$ (każda maszyna deterministyczna jest maszyną niedeterministyczną);
- $\mathbf{SPACE}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$ (j.w.);
- $\mathbf{TIME}(f(n)) \subseteq \mathbf{SPACE}(f(n))$ (maszyna nie może zapisać więcej komórek niż czas jej działania);
- $\mathbf{NTIME}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$ (j.w.);
- $\exists c > 1 \colon \mathbf{NTIME}(f(n)) \subseteq \mathbf{TIME}(c^{f(n)})$ (symulacja maszyny niedeterministycznej na maszynie deterministycznej);
- $\exists c > 1 \colon \mathbf{NSPACE}(f(n)) \subseteq \mathbf{TIME}(c^{f(n)})$ (j.w.);
- $\exists c > 1 \colon \mathbf{SPACE}(f(n)) \subseteq \mathbf{TIME}(c^{f(n)})$ (liczba możliwych konfiguracji maszyny);

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Klasy złożoności

Relacje pomiędzy klasami złożoności

- $\mathbf{TIME}(f(n)) \subseteq \mathbf{NTIME}(f(n))$ (każda maszyna deterministyczna jest maszyną niedeterministyczną);
- $\mathbf{SPACE}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$ (j.w.);
- $\mathbf{TIME}(f(n)) \subseteq \mathbf{SPACE}(f(n))$ (maszyna nie może zapisać więcej komórek niż czas jej działania);
- $\mathbf{NTIME}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$ (j.w.);
- $\exists c > 1 \colon \mathbf{NTIME}(f(n)) \subseteq \mathbf{TIME}(c^{f(n)})$ (symulacja maszyny niedeterministycznej na maszynie deterministycznej);
- $\exists c > 1 \colon \mathbf{NSPACE}(f(n)) \subseteq \mathbf{TIME}(c^{f(n)})$ (j.w.);
- $\exists c > 1 \colon \mathbf{SPACE}(f(n)) \subseteq \mathbf{TIME}(c^{f(n)})$ (liczba możliwych konfiguracji maszyny);
- $\mathbf{NTIME}(f(n)) \subseteq \mathbf{SPACE}(f(n))$ (maszyna deterministyczna może symulować niedeterministyczną).

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki