Złożoność obliczeniowa algorytmów

Inne modele obliczeń

Kordian A. Smoliński

Wydział Fizyki i Informatyki Stosowanej

2024/2025

Część I

Maszyny rejestrowe

Maszyny rejestrowe

1. Maszyna o dostępie swobodnym
   - Złożoność obliczeniowa dla RAM
2. Maszyna z zapamiętanym programem
   - Złożoność obliczeniowa RASP
3. RAM i RASP a maszyna Turinga
   - Złożoność RAM i maszyny Turinga

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Maszyny rejestrowe

Rejestr

Komórka pamięci posiadająca:

Maszyny rejestrowe

Rejestr

Komórka pamięci posiadająca:

adres jednoznaczny identyfikator (równoważny liczbie naturalnej),

Maszyny rejestrowe

Rejestr

Komórka pamięci posiadająca:

- adres jednoznaczny identyfikator (równoważny liczbie naturalnej),
- zawartość pojedyncza liczba naturalna (dowolnej wielkości).

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Maszyny rejestrowe

Rejestr

Komórka pamięci posiadająca:

- adres jednoznaczny identyfikator (równoważny liczbie naturalnej),
- zawartość pojedyncza liczba naturalna (dowolnej wielkości).

Zbiór rejestrów maszyny nazywamy pamięcią.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Maszyny rejestrowe

Rejestr

Komórka pamięci posiadająca:

adres jednoznaczny identyfikator (równoważny liczbie naturalnej),

zawartość pojedyncza liczba naturalna (dowolnej wielkości).

Zbiór rejestrów maszyny nazywamy pamięcią.

Zapis

$[r]$ zawartość rejestru o adresie r;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Maszyny rejestrowe

Rejestr

Komórka pamięci posiadająca:

adres jednoznaczny identyfikator (równoważny liczbie naturalnej),

zawartość pojedyncza liczba naturalna (dowolnej wielkości).

Zbiór rejestrów maszyny nazywamy pamięcią.

Zapis

$[r]$ zawartość rejestru o adresie r;

$r \leftarrow v$ załadowanie do rejestru o adresie r wartości v.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Maszyna o dostępie swobodnym

Random Access Machine (RAM)

Maszyna o dostępie swobodnym (RAM)

Maszyna o dostępie swobodnym składa się z:

Maszyna o dostępie swobodnym

Random Access Machine (RAM)

Maszyna o dostępie swobodnym (RAM)

Maszyna o dostępie swobodnym składa się z:

programu — skończonego ciągu rozkazów;

Maszyna o dostępie swobodnym

Random Access Machine (RAM)

Maszyna o dostępie swobodnym (RAM)

Maszyna o dostępie swobodnym składa się z:

programu — skończonego ciągu rozkazów;

pamięci — nieograniczonej liczby rejestrów $r_0, r_1, r_2, \ldots$;

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Maszyna o dostępie swobodnym

Random Access Machine (RAM)

Maszyna o dostępie swobodnym (RAM)

Maszyna o dostępie swobodnym składa się z:

programu — skończonego ciągu rozkazów;

pamięci — nieograniczonej liczby rejestrów $r_0, r_1, r_2, \ldots$;

taśm wejścia i wyjścia z komórkami mogącymi zawierać liczby naturalne, ruch taśm jest jednostronny — po przeczytaniu/zapisaniu komórki nie można do niej wrócić;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Maszyna o dostępie swobodnym

Random Access Machine (RAM)

Maszyna o dostępie swobodnym (RAM)

Maszyna o dostępie swobodnym składa się z:

programu — skończonego ciągu rozkazów;

pamięci — nieograniczonej liczby rejestrów $r_0, r_1, r_2, \ldots$;

taśm wejścia i wyjścia z komórkami mogącymi zawierać liczby naturalne, ruch taśm jest jednostronny — po przeczytaniu/zapisaniu komórki nie można do niej wrócić;

licznika rozkazów — rejestru i przechowującego numer następnej instrukcji programu do wykonania, początkowo $[i] = 0$.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Maszyna o dostępie swobodnym

Tabela: Przykładowa lista rozkazów

1	`LOAD =n`	$r_0 \leftarrow n$	13	`SUB =n`	$r_0 \leftarrow \max([r_0] - n, 0)$
2	`LOAD n`	$r_0 \leftarrow [r_n]$	14	`SUB n`	$r_0 \leftarrow \max([r_0] - [r_n], 0)$
3	`LOAD ↑n`	$r_0 \leftarrow [r_{[r_n]}]$	15	`SUB ↑n`	$r_0 \leftarrow \max([r_0] - [r_{[r_n]}], 0)$
4	`STORE n`	$r_n \leftarrow [r_0]$	16	`MUL =n`	$r_0 \leftarrow [r_0] \cdot n$
5	`STORE ↑n`	$r_{[r_n]} \leftarrow [r_0]$	17	`MUL n`	$r_0 \leftarrow [r_0] \cdot [r_n]$
6	`READ n`	$r_n \leftarrow [in]$	18	`MUL ↑n`	$r_0 \leftarrow [r_0] \cdot [r_{[r_n]}]$
7	`READ ↑n`	$r_{[r_n]} \leftarrow [in]$	19	`DIV =n`	$r_0 \leftarrow \lfloor [r_0]/n \rfloor$
8	`WRITE n`	$out \leftarrow [r_n]$	20	`DIV n`	$r_0 \leftarrow \lfloor [r_0]/[r_n] \rfloor$
9	`WRITE ↑n`	$out \leftarrow [r_{[r_n]}]$	21	`DIV ↑n`	$r_0 \leftarrow \lfloor [r_0]/[r_{[r_n]}] \rfloor$
10	`ADD =n`	$r_0 \leftarrow [r_0] + n$	22	`JUMP n`	$i \leftarrow n$
11	`ADD n`	$r_0 \leftarrow [r_0] + [r_n]$	23	`JZERO n`	$[r_0] = 0 \implies i \leftarrow n$
12	`ADD ↑n`	$r_0 \leftarrow [r_0] + [r_{[r_n]}]$	24	`JNZERO n`	$[r_0] \neq \implies i \leftarrow n$
			25	`HALT`	$i \leftarrow 0$

O ile nie zaznaczono inaczej, dla każdej operacji $i \leftarrow [i] + 1$.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Maszyna o dostępie swobodnym

Przykład (algorytm Euklidesa)

```
 1   READ 1
 2   READ 2
 3   LOAD 2
 4   JNZERO 18
 5   SUB 1
 6   JNZERO 11
 7   LOAD 1
 8   SUB 2
 9   STORE 1
10   JUMP 3
11   LOAD 2
12   STORE 3
13   LOAD 1
14   STORE 2
15   LOAD 3
16   STORE 1
17   JUMP 3
18   WRITE 1
19   HALT
```

Maszyna o dostępie swobodnym

Przykład (algorytm Euklidesa)

```
 1   READ 1
 2   READ 2
 3   LOAD 2
 4   JNZERO 18
 5   SUB 1
 6   JNZERO 11
 7   LOAD 1
 8   SUB 2
 9   STORE 1
10   JUMP 3
11   LOAD 2
12   STORE 3
13   LOAD 1
14   STORE 2
15   LOAD 3
16   STORE 1
17   JUMP 3
18   WRITE 1
19   HALT
```

Require: $a, b \in \mathbb{N}$
Ensure: $\mathrm{nwd}(a, b) \in \mathbb{N}$
$\mathrm{nwd}(a,\ b)$
function $\mathrm{nwd}(a,\ b)$
while $b \neq 0$ **do**
if $a \geqslant b$ **then**
$a \leftarrow a - b$
else
$c \leftarrow b$
$b \leftarrow a$
$a \leftarrow c$
end if
end while
return a
end function

Maszyna o dostępie swobodnym

Złożonosć obliczeniowa dla RAM

Kryterium kosztu jednostajnego

czas wykonania rozkazu jest ustalony.

Maszyna o dostępie swobodnym

Złożonosć obliczeniowa dla RAM

Kryterium kosztu jednostajnego

czas wykonania rozkazu jest ustalony.

pamięć to jest liczba rejestrów, do których odwołał się program w trakcie działania.

Maszyna o dostępie swobodnym

Złożonosć obliczeniowa dla RAM

Kryterium kosztu jednostajnego

czas wykonania rozkazu jest ustalony.

pamięć to jest liczba rejestrów, do których odwołał się program w trakcie działania.

Kryterium kosztu logarytmicznego

czas wykonania rozkazu na liczbie n jest proporcjonalny do $\lfloor \log n \rfloor + 1$.

Maszyna o dostępie swobodnym

Złożonosć obliczeniowa dla RAM

Kryterium kosztu jednostajnego

czas wykonania rozkazu jest ustalony.

pamięć to jest liczba rejestrów, do których odwołał się program w trakcie działania.

Kryterium kosztu logarytmicznego

czas wykonania rozkazu na liczbie n jest proporcjonalny do $\lfloor \log n \rfloor + 1$.

czas dostępu do rejestru r_k w adresowaniu pośrednim jest proporcjonalny do $\lfloor \log k \rfloor + 1$.

Maszyna o dostępie swobodnym

Złożonosć obliczeniowa dla RAM

Kryterium kosztu jednostajnego

czas wykonania rozkazu jest ustalony.

pamięć to jest liczba rejestrów, do których odwołał się program w trakcie działania.

Kryterium kosztu logarytmicznego

czas wykonania rozkazu na liczbie n jest proporcjonalny do $\lfloor \log n \rfloor + 1$.

czas dostępu do rejestru r_k w adresowaniu pośrednim jest proporcjonalny do $\lfloor \log k \rfloor + 1$.

pamięć to suma logarytmów maxymalnych wartości wszystkich rejestrów, do których odwołał się program w trakcie działania.

Maszyna z zapamiętanym programem

Random-access stored-program (RASP)

Odpowiednik uniwersalnej maszyny Turinga dla maszyn o dostępie swobodnym.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Maszyna z zapamiętanym programem

Random-access stored-program (RASP)

Odpowiednik uniwersalnej maszyny Turinga dla maszyn o dostępie swobodnym.

Maszyna z zapamiętanym programem

Maszyna o dostępie swobodnym wczytująca program i dane z taśmy wejściowej do pamięci; rozkazy RAM są zakodowane w postaci liczb naturalnych.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Maszyna z zapamiętanym programem

Random-access stored-program (RASP)

Odpowiednik uniwersalnej maszyny Turinga dla maszyn o dostępie swobodnym.

Maszyna z zapamiętanym programem

Maszyna o dostępie swobodnym wczytująca program i dane z taśmy wejściowej do pamięci; rozkazy RAM są zakodowane w postaci liczb naturalnych.

Licznik rozkazów

Każda (prawie) instrukcja RASP jest zakodowana w postaci dwóch liczb naturalnych: kodu rozkazu i argumentu rozkazu, zatem dla większosci instrukcji mamy $i \leftarrow [i] + 2$.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Złożoność czasową i pamięciową dla RASP definiujemy podobnie jak dla RAM.

Twierdzenie

Jeżeli koszt instukcji jest jednostajny lub logarytmiczny, to

Złożoność czasową i pamięciową dla RASP definiujemy podobnie jak dla RAM.

Twierdzenie

Jeżeli koszt instukcji jest jednostajny lub logarytmiczny, to

1. *dla każdego programu RAM o złożoności czasowej $T_1(n)$ istnieje równoważny program RASP o złożoności czasowej $\Theta(T_1(n))$;*

Złożoność czasową i pamięciową dla RASP definiujemy podobnie jak dla RAM.

Twierdzenie

Jeżeli koszt instukcji jest jednostajny lub logarytmiczny, to

1. *dla każdego programu RAM o złożoności czasowej $T_1(n)$ istnieje równoważny program RASP o złożoności czasowej $\Theta(T_1(n))$;*
2. *dla każdego programu RASP o złożoności czasowej $T_2(n)$ istnieje równoważny program RAM o złożoności czasowej $O(T_2(n))$.*

RAM i RASP a maszyna Turinga

- RAM może symulować maszynę Turinga przechowując i-tą komórkę taśmy TM w rejestrze $i + c$, gdzie c dobieramy tak, aby zapewnić RAM „pamięć roboczą”.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

RAM i RASP a maszyna Turinga

- RAM może symulować maszynę Turinga przechowując i-tą komórkę taśmy TM w rejestrze $i + c$, gdzie c dobieramy tak, aby zapewnić RAM „pamięć roboczą".
- Dla dowolnego obliczenia RAM, którego koszt logarytmiczny jest k, można zbudować maszynę Turinga symulującą to obliczenie w $O(k^2)$ krokach.

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Definicja

Niech $f_1, f_2 : \mathbb{N} \to \mathbb{R}$. f_1 i f_2 są równoważne wielomianowo jeżeli istnieją wielomiany $p_1, p_2 : \mathbb{R} \to \mathbb{R}$ takie, że

$$\forall n \in \mathbb{N}: f_1(n) \leqslant p_1(f_2(n)) \wedge f_2(n) \leqslant p_2(f_1(n)).$$

Definicja

Niech $f_1, f_2 : \mathbb{N} \to \mathbb{R}$. f_1 i f_2 są równoważne wielomianowo jeżeli istnieją wielomiany $p_1, p_2 : \mathbb{R} \to \mathbb{R}$ takie, że

$$\forall n \in \mathbb{N}: f_1(n) \leqslant p_1(f_2(n)) \wedge f_2(n) \leqslant p_2(f_1(n)).$$

Twierdzenie

RAM i RASP z kosztem logarytmicznym oraz maszyna Turinga są równoważnymi wielomianowo modelami obliczeń.

Część II

Obwody logiczne

4. Obwody logiczne
   - Złożoność w modelu obwodów logicznych

Obwody logiczne

Obwód logiczny

Acykliczny graf skierowany, w którym każdy wierzchołek jest albo:

Obwody logiczne

Obwód logiczny

Acykliczny graf skierowany, w którym każdy wierzchołek jest albo:

- wejściem o stopniu wejściowym 0;

Obwody logiczne

Obwód logiczny

Acykliczny graf skierowany, w którym każdy wierzchołek jest albo:

- wejściem o stopniu wejściowym 0;
- wyjściem o stopniu wejściowym 1 i stopniu wyjściowym 0;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Obwody logiczne

Obwód logiczny

Acykliczny graf skierowany, w którym każdy wierzchołek jest albo:

- wejściem o stopniu wejściowym 0;
- wyjściem o stopniu wejściowym 1 i stopniu wyjściowym 0;
- koniunkcją o stopniu wejściowym 2;

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Obwody logiczne

Obwód logiczny

Acykliczny graf skierowany, w którym każdy wierzchołek jest albo:

- wejściem o stopniu wejściowym 0;
- wyjściem o stopniu wejściowym 1 i stopniu wyjściowym 0;
- koniunkcją o stopniu wejściowym 2;
- alternatywą o stopniu wejściowym 2;

WYDZIAŁ FIZYKI i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Obwody logiczne

Obwód logiczny

Acykliczny graf skierowany, w którym każdy wierzchołek jest albo:

- wejściem o stopniu wejściowym 0;
- wyjściem o stopniu wejściowym 1 i stopniu wyjściowym 0;
- koniunkcją o stopniu wejściowym 2;
- alternatywą o stopniu wejściowym 2;
- negacją o stopniu wejściowym 1.

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Obwody logiczne

Obwód logiczny

Acykliczny graf skierowany, w którym każdy wierzchołek jest albo:

- wejściem o stopniu wejściowym 0;
- wyjściem o stopniu wejściowym 1 i stopniu wyjściowym 0;
- koniunkcją o stopniu wejściowym 2;
- alternatywą o stopniu wejściowym 2;
- negacją o stopniu wejściowym 1.

Jeżeli obwód logiczny ma m wejść i n wyjść, to określa funkcję typu

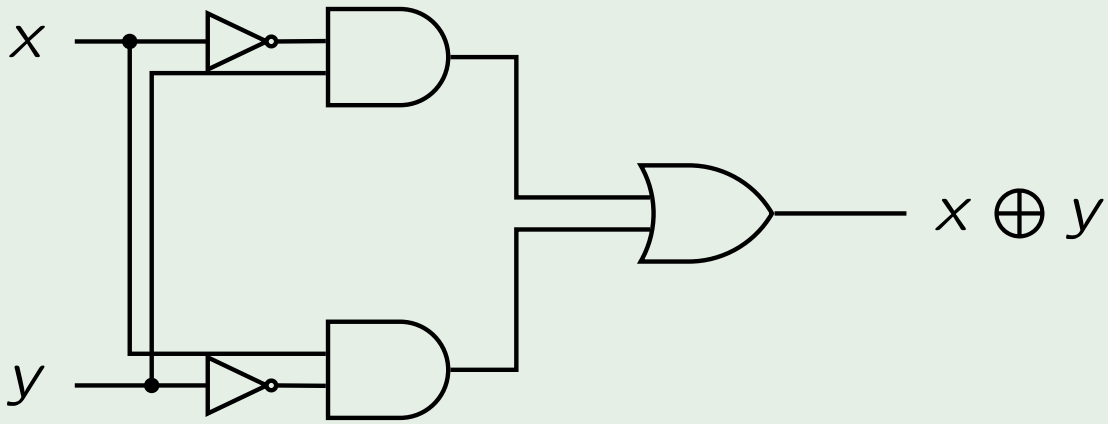
$$f : \{0,1\}^m \to \{0,1\}^n.$$

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki

Obwody logiczne

Przykład (kontrawalencja (XOR))

$$x \oplus y = [x \wedge (\neg y)] \vee [(\neg x) \wedge y]$$



Definicja

Niech $\mathcal{R}$ będzie rodziną obwodów logicznych taką, że dla każdego $n \in \mathbb{N}$ rodzina $\mathcal{R}$ zawiera dokładnie jeden obwód o n węzłach wejściowych. Funkcja $f : \mathbb{N} \to \mathbb{N}$ przyporządkowująca $n \in \mathbb{N}$ liczbę węzłów wewnętrznych (bramek) w obwodzie z rodziny $\mathcal{R}$ o n wejściach to miara złożoności ilościowej.

Definicja

Niech $\mathcal{R}$ będzie rodziną obwodów logicznych taką, że dla każdego $n \in \mathbb{N}$ rodzina $\mathcal{R}$ zawiera dokładnie jeden obwód o n węzłach wejściowych. Funkcja $f : \mathbb{N} \to \mathbb{N}$ przyporządkowująca $n \in \mathbb{N}$ liczbę węzłów wewnętrznych (bramek) w obwodzie z rodziny $\mathcal{R}$ o n wejściach to miara złożoności ilościowej.

Twierdzenie

Jeżeli M jest maszyną Turinga działającą na słowie wejściowym w o długości n w czasie $T(n)$, to istnieje rodzina obwodów logicznych $\mathcal{R}$, symulująca działanie M, o złożoności ilościowej $O(T(n)^2)$.