§8 Złożoność

- Ile zasobów jest potrzebny do obliczenia f?
- Zasoby: zwykle czas i pamięć
- Miary złożoności zdefiniowane dla maszyn Turinga
- rozważamy problemy decyzyjne, tzn. języki $L \subseteq \Sigma^*$ i funkcje χ_L. Problemy obliczalne (optymalizujące) traktujemy jako problemy decyzyjne spytając 'Czy istnieje rozwiązanie problemu mniejsze niż k.

8.1 Definicje

Def: M maszyna Turinga

(i) złożoność czasowa maszyny M, to funkcja $T_M : \mathbb{N} \rightsquigarrow \mathbb{N}$, taka że

- $T_M(n)\uparrow$, jeżeli nie ma słowa $w \in \Sigma^*$ długości n, którego M akceptuje.
- $T_M(n)$ = najmniejsze m, takie że wszystkie słowa $w \in \Sigma^*$ długości n, które M akceptuje, są akceptowane przez obliczenie długości co najwyżej m.

Dla $h : \mathbb{N} \to \mathbb{N}$ maszyna M jest czasowo h-ograniczona jeżeli z $T_M(n)\downarrow$ wynika $h(n)\downarrow$ i $T_M(n) \leq h(n)$.

(ii) złożoność pamięciowa maszyny M, to funkcja $S_M : \mathbb{N} \rightsquigarrow \mathbb{N}$, taka że

- $S_M(n)\uparrow$, jeżeli nie ma słowa $w \in \Sigma^*$ długości n, którego M akceptuje.
- $S_M(n)$ = najmniejsze m, takie że wszystkie słowa $w \in \Sigma^*$ długości n, które M akceptuje, są akceptowane przez obliczenie używając conajwyżej m komórek na taśmie.

Dla $h : \mathbb{N} \to \mathbb{N}$ maszyna M jest pamięciowo h-ograniczona, jeżeli z $S_M(n)\downarrow$ wynika $h(n)\downarrow$ i $S_M(n) \leq h(n)$.

Przykład:

$L = \{ wcw \mid w \in \{a,b\}^* \}$

Maszyna M akceptująca L czyta pierwszą literę, znajduje c i sprawdza, czy po c jest ta sama litera. Jeżeli tak, M czyta kolejną literę i wraca na początek i ponowie porównuje litery. Jeżeli na taśmie zostaje tylko litera c, to M akceptuje, w innym przypadku M nie akceptuje.

Jeżeli $|wcw| = n$, to M wykonuje mniej niż $2(\lfloor \frac{n}{2} \rfloor + 3) \cdot \lfloor \frac{n}{2} \rfloor$ operacje, czyli M jest czasowo n^2-ograniczona.

Złożoność pamięciowa M, to $n+1$.

Jeżeli pozwolone są dwie taśmy, to istnieje maszyna M' akceptująca L, która jest czasowo n-ograniczona:

M' idąc do c kopiuje słowo w do drugiej taśmy. Po litery c M' dalej czyta wejście jednocześnie porównując z słowem na drugiej taśmie.

<u>Def:</u>

$f: \mathbb{N} \longrightarrow \mathbb{N}$

(i) $DTIME(f) :=$

$\{L \subseteq \Sigma^* \mid$ istnieje deterministyczna czasowo f-ograniczona maszyna Turinga M, która akceptuje $L\}$

(ii) $NTIME(f) :=$

$\{L \subseteq \Sigma^* \mid$ istnieje niedeterministyczna czasowo f-ograniczona maszyna Turinga M, która akceptuje $L\}$

(iii) $DSPACE(f) :=$

$\{L \subseteq \Sigma^* \mid$ istnieje deterministyczna pamięciowo f-ograniczona maszyna Turinga M, która akceptuje $L\}$

(iv) $NSPAC(f) :=$

$\{L \subseteq \Sigma^* \mid$ istnieje niedeterministyczna pamięciowo f-ograniczona maszyna Turinga M, która akceptuje $L\}$

Założenia

- każda maszyna Turinga M używa jedną komórkę, tzn. $S_M(n) \geq 1$ dla wszystkich n, więc $(c \cdot S_M)(n) := \max\{1, c \cdot \lceil S_M(n) \rceil\}$, dla $c > 0$
- każda maszyna Turinga M czyta całe wejście, tzn. $T_M(n) \geq n+1$ dla wszystkich n, więc $(c \cdot T_M)(n) := \max\{n+1, c \cdot \lceil T_M(n) \rceil\}$

Twierdzenie

$f: \mathbb{N} \longrightarrow \mathbb{N}$, $c > 0$

(i) $DSPACE(f) = DSPACE(c \cdot f)$

$NSPACE(f) = NSPACE(c \cdot f)$.

(ii) Jeżeli f rośnie mocniej niż liniowe (czyli $\forall k \exists m \forall n \geq m : f(n) \geq k \cdot n$), to

$DTIME(f) = DTIME(c \cdot f)$

$NTIME(f) = NTIME(c \cdot f)$

"Dowód": Powiększenie alfabetu

Twierdzenie

(i) Jeżeli $L \subseteq \Sigma^*$ jest akceptowana przez (nie-) deterministyczną pamięciowo f-ograniczoną maszynę Turinga M z k taśmami, to istnieje (nie-) deterministyczna pamięciowo f-ograniczona maszyna Turinga M' z jedną taśmą, który akceptuje L.

(ii) Jeżeli $L \subseteq \Sigma^*$ jest akceptowana przez (nie-) deterministyczną czasowo f-ograniczoną maszynę Turinga M z k taśmami, to istnieje (nie-) deterministyczna czasowo f^2- ograniczona maszyna Turinga M' z jedną taśmą, która akceptuje L.

"Dowód":

M' używa taśmę wielościeszką

8.2 Czas vs. Pamięć

Twierdzenie

$$DTIME(f) \subseteq NTIME(f) \subseteq DSPACE(f) \subseteq NSPACE(f)$$

"Dowód":

M niedeterministyczna czasowo f-ograniczona maszyna Turinga. Wtedy M pamięciowo f'- ograniczona, gdzie $f'(n) = f(n)+1$.

Deterministyczna maszyna M' systematycznie symuluje możliwe obliczenia maszyny M. Do tego M' potrzebuje

- pamięć, żeby zachować przegląd alternatyw maszyny M; ograniczona przez $f(n)$
- pamięć dla obliczeń maszyny M; ograniczona przez $f'(n)$

czyli M pamięciowo f-ograniczona.

Twierdzenie

$$NSPACE(f) \subseteq \bigcup_{c>0} DTIME(c^f)$$

Uwaga: Stąd też

$$NDTIME(f) \subseteq \bigcup_{c>0} DTIME(c^f)$$

Dla pamięci sytuacja jest lepsza – jeżeli f spełnia następujący warunek.

Def: $f: \mathbb{N} \to \mathbb{N}$

(i) f jest pamięciowo konstruowalna, jeżeli istnieje deterministyczna pamięciowo f-ograniczona maszyna Turinga M, taka że dla wszystkich n istnieje słowo w długości n, takie że M używa dokładnie $f(n)$ komórek.

(ii) f jest całkowicie pamięciowo konstruowalna, jeżeli M używa dokładnie $f(n)$ komórek dla wszystkich słów długości n.

Uwaga:

n^2, 2^n, $n!$, $\sqrt{n}$, $\log_2 n$ są całkowicie pamięciowo konstruowalne.

Twierdzenie

f całkowicie pamięciowo konstruowalna oraz $f(n) \geq \log_2 n$ dla wszystkich n. Wtedy

$$NSPACE(f) \subseteq DSPACE(f^2)$$

8.3 Hierarchie

Twierdzenie

Niech $f: \mathbb{N} \to \mathbb{N}$ obliczalna.
Wtedy istnieje rozstrzygalny $L \subseteq \Sigma^*$, taki że

$$L \notin DTIME(f).$$

„Dowód": Diagonalizacja

$$L := \{ x_i \mid \text{maszyna Turinga } M_i \text{ nie akceptuje } x_i \text{ w czasie } f(|x_i|) \}$$

Uwaga: Analogiczne rezultaty dostaniemy dla NTIME, DSPACE i NSPACE.

Jaka musi być „różnica" między f_1 a f_2, żeby ich klasy złożoności były różne?

<u>Def:</u> $f: \mathbb{N} \to \mathbb{N}$

(i) f jest czasowo konstrualna, jeżeli istnieje deterministyczna czasowo f-ograniczona maszyna Turinga, taka że dla wszystkich n istnieje słowo w długości n, takie że M wykonuje obliczenie długosci dokładnie $f(n)$.

(ii) f jest całkowicie czasowo konstrualna, jeżeli M wykonuje obliczenie długości dokładnie $f(n)$ dla wszystkich słów długości n.

<u>Twierdzenie</u>

(i) $f_1, f_2: \mathbb{N} \to \mathbb{N}$ całkowicie pamięciowo konstrualne, takie że $\forall k \exists m \forall n \geq m: f_2(n) \geq k \cdot f_1(n)$.

Wtedy $DSPACE(f_1) \subsetneq DSPACE(f_2)$.

(ii) $f_1, f_2: \mathbb{N} \to \mathbb{N}$ całkowicie czasowo konstrualne, takie że $\forall k \exists m \forall n \geq m: f_2(n) \geq k \cdot (f(n))^2$.

Wtedy

$$DTIME(f_1) \subsetneq DTIME(f_2).$$

"Dowod:" Diagonalizacja

Twierdzenie

(i) $f_1, f_2, g : \mathbb{N} \to \mathbb{N}$ całkowicie pamięciowo konstruowalne, takie że $f_2(n) \geq n$ i $g(n) \geq n$ dla wszystkich n. Wtedy

- jeżeli $DSPACE(f_1) \subseteq DSPACE(f_2)$, to $DSPACE(f_1 \circ g) \subseteq DSPACE(f_2 \circ g)$.
- jeżeli $NSPACE(f_1) \subseteq NSPACE(f_2)$, to $NSPACE(f_1 \circ g) \subseteq NSPACE(f_2 \circ g)$.

(ii) $f_1, f_2, g : \mathbb{N} \to \mathbb{N}$ całkowicie czasowo konstruowalne, takie że $f_1(n) \geq n$, $f_2(n) \geq n$ i $g(n) \geq n$ dla wszystkich n. Wtedy

- jeżeli $DTIME(f_1) \subseteq DTIME(f_2)$, to $DTIME(f_1 \circ g) \subseteq DTIME(f_2 \circ g)$.
- jeżeli $NTIME(f_1) \subseteq NTIME(f_2)$, to $NTIME(f_1 \circ g) \subseteq NTIME(f_2 \circ g)$.

Przykład:

założenie: $NSPACE(n^3) \subseteq NSPACE(n^2)$

wtedy $NSPACE(n^6) \subseteq NSPACE(n^4)$

i $NSPACE(n^9) \subseteq NSPACE(n^6)$

czyli $DSPACE(n^9) \subseteq NSPACE(n^9)$

$\subseteq NSPACE(n^4) \subseteq DSPACE(n^8)$

ale $\forall k \exists m \forall n \geq m : n^9 \geq k \cdot n^8$,

czyli $DSPACE(n^9) \not\supseteq DSPACE(n^8)$ ↯

8.4 Dalsze własciowości

Twierdzenie

Niech $g: \mathbb{N} \to \mathbb{N}$ obliczalna, taka że $g(n) \geq n$. Wtedy istnieje $f: \mathbb{N} \to \mathbb{N}$ obliczalna, taka że

$$DTIME(f) = DTIME(g \circ f)$$

tzn. nie ma języka L, który jest akceptowany przez $g \circ f$ - ograniczoną maszynę, ale nie jest akceptowany przez f- ograniczoną maszynę.

Uwaga: f nie czasowo konstruowalna

Twierdzenie

Istnieje $f: \mathbb{N} \to \mathbb{N}$ obliczalna, taka że

$$DTIME(f) = NTIME(f) = DSPACE(f) = NSPACE(f).$$

"Dowód":

Dla $g(x) := x^x$ istnieje $f: \mathbb{N} \to \mathbb{N}$, taka że

$$DTIME(f) = DTIME(g \circ f) = DTIME(f^f).$$

Niech $L \in NSPACE(f)$.
Wtedy $L \in DTIME(c^f)$ dla stałej c. Stąd wynika, że $L \in DTIME(f^f) = DTIME(f)$.

Twierdzenie

Niech $r: \mathbb{N} \to \mathbb{N}$ obliczalna.

Wtedy istnieje rozstrzygalny $L \subseteq \Sigma^*$, taki że dla każdego maszyna Turinga M, która akceptuje L istnieje maszyna Turinga M', która akceptuje L, i $r(T_{M'}(n)) \leq T_M(n)$ dla prawie wszystkich n,

tzn. istnieją języki (problemy), które nie mają najszybszego rozwiązania.

"Dowód": Diagonalizacja