Obliczalność i złożoność

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

sem. zimowy 2025/26

inf.ug.edu.pl/~amb/oblicz

Nierozstrzygalność

Kodowanie maszyn Turinga

- Kodujemy maszynę Turinga M nad alfabetem wejściowym $\{0,1\}$

$$M = (Q, \{0,1\}, \{0,1,B\}, \delta, q_1, B, \{q_2\})$$

$Q = \{q_1, q_2, \ldots, q_n\}$,
stan q_1 jest stanem początkowym,
q_2 jest jedynym stanem końcowym, akceptującym

- symbole alfabetu roboczego 0,1,B przez X_1, X_2, X_3,
- kierunki ruchów L, R, N przez D_1, D_2, D_3,
- instrukcja czyli wartość funkcji przejścia

$$\delta(q_i, X_j) = (q_k, X_\ell, D_m), \quad 1 \leq i,k \leq n,\ 1 \leq j,\ell,m \leq 3$$

kodujemy przez $0^i 1 0^j 1 0^k 1 0^\ell 1 0^m$

Kodowanie maszyn Turinga, c.d.

- Kod maszyny Turinga M

$$\langle M \rangle = 111 kod_1 11 kod_2 11 \cdots 11 kod_r 111$$

gdzie kod_i jest kodem kolejnej i-tej instrukcji maszyny M

- ciąg 0 i 1 jest kodem maszyny Turinga, jeśli
  - nie ma czterech jedynek pod rząd,
  - zaczyna się i kończy trzema jedynkami, innych wystąpień trzech nie ma
  - podwójne jedynki przedzielane są blokami, w których wystepują cztery pojedyncze jedynki,
  - pojedyncze jedynki przedzielane są blokami zer sensownej długości

Uniwersalna maszyna Turinga

- def.: **uniwersalna maszyna Turinga** M_u nad $\Sigma = \{0, 1\}$, dla wejścia

$$\langle M, w \rangle$$

  gdzie $\langle M \rangle$ jest kodem maszyny M, a $w \in \Sigma^*$,
  symuluje obliczenie M na słowie w
- maszyna M_u akceptuje parę $\langle M, w \rangle$ wtedy i tylko wtedy, gdy maszyna M akceptuje w
- def.: język uniwersalny

$$L_u = \{\langle M, w \rangle \mid M \text{ akceptuje } w\}$$

Przypomnienie: rekurencyjność a rekurencyjna przeliczalność

- język L jest rekurencyjnie przeliczalny (*recursively enumerable*) jeśli $L = L(\mathcal{A})$ dla pewnej maszyny Turinga $\mathcal{A}$
- akceptowanie: maszyna zatrzyma się na słowach, które zaakceptuje, ale ma prawo nie kończyć obliczeń dla innych
- rozstrzygalność: maszyna zatrzyma się na każdym słowie i da odpowiedź tak/nie
- język L jest rekurencyjny (*recursive*) jeśli maszyna Turinga akceptująca słowa z L ma własność stopu
- tw.: jeśli język L i jego uzupełnienie $\Sigma^* \setminus L$ są rekurencyjnie przeliczalne, to są rekurencyjne
- dw: jedna maszyna akceptuje L, druga dopełnienie, na pewno jedna się zatrzyma i wówczas mamy rozstrzygnięcie
- jeśli jeden z języków L i $\Sigma^* \setminus L$ jest rekurencyjny to jest i drugi

Język uniwersalny L_u

Język L_u

Tw.: Język L_u jest rekurencyjnie przeliczalny

- dw.: maszyna uniwersalna M_u (akceptująca L_u) posiada trzy taśmy:
  - na pierwszej taśmie przechowuje dane wejściowe $\langle M, w \rangle$
  - na drugą przepisuje słowo w, będzie tam symulowana jedyna taśma maszyny M
  - na trzeciej taśmie maszyna M_u przechowuje numer bieżącego stanu maszyny M (na początku jest to stan początkowy)
  - krok po kroku symulowane są kroki obliczenia maszyny M
- maszyna M_u akceptuje, jeżeli symulowane obliczenie zatrzyma się w stanie akceptującym
- pytanie, czy jest L_u jest rekurencyjny, czyli czy dla każdej maszyny Turinga M możemy rozstrzygnąć czy akceptuje wejście w?

Porządek kanoniczny słów nad alfabetem $\{0, 1\}$

- $$\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots$$

  słowa są ustawione według wzrastającej długości, słowa jednakowej długości występują w porządku leksykograficznym.
- po dodaniu 1 z przodu otrzymujemy zapis binarny kolejnych liczb naturalnych

Język przekątniowy

- w_i oznacza i-te słowo na liście słów w porządku kanonicznym (czyli i w zapisie binarnym z usuniętą wiodącą jedynką)
- M_j oznacza j-tą maszynę Turinga, to znaczy maszynę, której kod jest liczbą j zapisaną binarnie (jeżeli liczba j nie jest prawidłowym kodem, to koduje maszynę, która niczego nie akceptuje)
- definujemy dwuwymiarową tablicę $\mathcal{D}$

$$\mathcal{D}[i,j] = \begin{cases} 1 & \text{jeżeli } w_i \in L(M_j) \\ 0 & \text{jeżeli } w_i \notin L(M_j) \end{cases}$$

- język przekątniowy

$$L_{\mathcal{D}} = \{w_i \mid \mathcal{D}[i,i] = 0\}$$

czyli $L_{\mathcal{D}} = \{w_i \mid w_i \notin L(M_i)\}$

Język przekątniowy nie jest rekurencyjnie przeliczalny

Język $L_{\mathcal{D}}$

Tw.: język $L_{\mathcal{D}} = \{w_i \mid w_i \notin L(M_i)\}$ nie jest rekurencyjnie przeliczalny

- dw.: gdyby był, to byłby rozpoznawany przez pewną maszynę M_k (k jest kodem tej maszyny)
- sprawdzamy, czy $w_k \in L_{\mathcal{D}}$
- jeśli tak, to z definicji $L_{\mathcal{D}}$, $w_k \notin L(M_k)$, czyli M_k nie akceptuje w_k
- ale $L_{\mathcal{D}}$ ma być rozpoznawany przez M_k czyli w_k ma być akceptowane
- i na odwrót, jeśli $w_k \notin L_{\mathcal{D}}$, to w_k jest akceptowane przez M_k
- sprzeczność, nie ma takiej maszyny

Język uniwersalny nie jest rekurencyjny

Język L_u

Tw.: język $L_u = \{\langle M, w\rangle \mid M \text{ akceptuje } w\}$ nie jest rekurencyjny

- dowód nie wprost:
  - przypuśćmy, że istnieje maszyna Turinga M, która zatrzymuje się na każdym wejściu i rozpoznaje język L_u
  - wtedy istnieje maszyna M^d, która także zatrzymuje się na każdym wejściu i rozpoznaje język $\{\langle M_i, w_i\rangle \mid M_i \text{ akceptuje } w_i\}$ czyli również język $\{w_i \mid M_i \text{ akceptuje } w_i\}$ – uzupełnienie języka przekątniowego
  - wówczas język przekątniowy $L_{\mathcal{D}}$ byłyby rekurencyjny
  - ale nie jest nawet rekurencyjnie przeliczalny – sprzeczność