

Języki, automaty i obliczenia

Wykład 10: Obliczalność

Sławomir Lasota

Uniwersytet Warszawski

29 kwietnia 2026

1 Teza Churcha-Turinga

2 Problem stopu

3 Rozstrzygalność

4 Obliczalność

Języki będziemy utożsamiać z *problemami decyzyjnymi*.

Cykl Hamiltona

Dane:	graf skierowany G
Wynik:	rozstrzygnąć, czy G ma cykl Hamiltona ?

Graf można opisać słowem nad $A = \{0, 1, \#\}$, np. 110#101#001

- Dane wejściowe nazywamy *instancją* problemu decyzyjnego.
- Dane wejściowe można opisać („kodować”) na różne sposoby słowem nad A .
- A co ze słowami, które nie opisują żadnej instancji?

języki rozpoznawane
przez maszyny Turinga

=

problemy decyzyjne, dla których
istnieje efektywny algorytm, przy
założeniu nieograniczonych zasobów

albo:

maszyny Turinga

≡

komputery

Wątpliwości:

- dlaczego nie maszyny Turinga z pamięcią stałą?
- nieskończoność taśmy: rozmiar pamięci rośnie wraz z rozmiarem instancji

- gramatyki typu 0
- automaty z wieloma stosami
- automaty z licznikami
- automaty z kolejką
- maszyny RAM (ang. *random access machines*) albo maszyny rejestrowe
- while-programy
- rachunek λ
- ...

1 Teza Churcha-Turinga

2 Problem stopu

3 Rozstrzygalność

4 Obliczalność

Bez u.o. możemy ograniczyć się do maszyn, w których

$$A = \{0, 1\} \qquad T = \{0, 1, \mathbb{B}\}.$$

Faktycznie, np. słowa nad alfabetem

$$T = \{a, b, c, d, \mathbb{B}\}$$

można zapisać jako słowa nad $\{0, 1, \mathbb{B}\}$,

$$a\,b\,c\,a\,\mathbb{B}\,d = 1000 \quad 0100 \quad 0010 \quad 1000 \quad \mathbb{B}\mathbb{B}\mathbb{B}\mathbb{B} \quad 0001$$

Problem stopu

Dane: maszyna Turinga $\mathcal{M}$ nad alfabetem A i słowo $w \in A^*$
Wynik: czy $w \in L(\mathcal{M})$?

Maszynę $\mathcal{M} = (A, Q, q_0, q_{\text{tak}}, T, \mathbb{B}, \delta)$ można opisać słowem $w_{\mathcal{M}} \in \{0, 1, \#\}^*$, np.

000000#001000#000001##000100#001#010000#100#001##...

albo słowem $w_{\mathcal{M}} \in \{0, 1\}^*$, np.

000000 1 001000 000001 000100 001 010000 100 001 ...

Program może stanowić dane wejściowe...

Język uniwersalny (problem stopu):

$$L_u = \{ w_{\mathcal{M}} \$ w : w \in L(\mathcal{M}) \} \subseteq \{0,1,\$ \}^*$$

$$L_u = \{ (\mathcal{M}, w) : w \in L(\mathcal{M}) \}$$

Twierdzenie

Problem stopu jest częściowo rozstrzygalny.

Dowód: Uniwersalna maszyna Turinga:

000000#001000#000001##000100#001#010000#100#001##...\$
000100\$0010101010101 $\bar{1}$ 111101011

Pytanie

Czy istnieją języki, które nie są częściowo rozstrzygalne?

W pewnym mieście żyje fryzjer, który strzyże tych mieszkańców, którzy nie strzygą się sami. Czy ten fryzjer strzyże się sam?

Twierdzenie

Język „przekątniowy”:

$$L_p = \{ w_{\mathcal{M}} : w_{\mathcal{M}} \notin L(\mathcal{M}) \}$$

$$L_p = \{ \mathcal{M} : \mathcal{M} \notin L(\mathcal{M}) \}$$

nie jest częściowo rozstrzygalny.

Dowód:

Przypuśćmy, że $L(\mathcal{M}_p) = L_p$. Wtedy

$$w_{\mathcal{M}_p} \in L(\mathcal{M}_p) \iff w_{\mathcal{M}_p} \notin L(\mathcal{M}_p)$$

$$\mathcal{M}_p \in L(\mathcal{M}_p) \iff \mathcal{M}_p \notin L(\mathcal{M}_p)$$

1 Teza Churcha-Turinga

2 Problem stopu

3 Rozstrzygalność

4 Obliczalność

Deterministyczna maszyna Turinga $\mathcal{M} = (A, Q, q_0, q_{\text{tak}}, q_{\text{nie}}, T, \mathbb{B}, \delta)$

- $q_{\text{tak}} \in Q$ – stan akceptujący
- $q_{\text{nie}} \in Q$ – stan odrzucający
- $\delta : (Q - \{q_{\text{tak}}, q_{\text{nie}}\}) \times T \rightarrow Q \times T \times \{\leftarrow, \circ, \rightarrow\}$

Konfiguracje **końcowe**:

$$T^*\{q_{\text{tak}}, q_{\text{nie}}\}T^* = \{w q w' : q \in \{q_{\text{tak}}, q_{\text{nie}}\}; w, w' \in T^*\}$$

Deterministyczna maszyna Turinga $\mathcal{M}$ **zatrzymuje się** dla słowa wejściowego w jeśli

$$q_0 w \xrightarrow{*}_{\mathcal{M}} c$$

dla jakiejś konfiguracji końcowej c .

Maszyna $\mathcal{M}$ jest **całkowita**, jeśli zatrzymuje się dla każdego słowa wejściowego.

Przykład (całkowita maszyna deterministyczna)

• $q_{\text{tak}} = \text{ok}$

• $q_{\text{nie}} = \text{nok}$

	a	b	$\mathbb{B}$	$\#$
$\text{start}_{\rightarrow}$	$(\text{cont}_{\rightarrow}, \#, \rightarrow)$	$(\text{check}, b, \rightarrow)$	$(\text{nok}, \mathbb{B}, \circ)$	$(\text{nok}, \#, \circ)$
$\text{cont}_{\rightarrow}$	$(\text{cont}_{\rightarrow}, a, \rightarrow)$	$(\text{cont}_{\rightarrow}, b, \rightarrow)$	$(\text{start}_{\leftarrow}, \mathbb{B}, \leftarrow)$	$(\text{start}_{\leftarrow}, \#, \leftarrow)$
$\text{start}_{\leftarrow}$	$(\text{cont}_{\leftarrow}, \#, \leftarrow)$	$(\text{nok}, b, \circ)$	$(\text{nok}, \mathbb{B}, \circ)$	$(\text{nok}, \#, \circ)$
$\text{cont}_{\leftarrow}$	$(\text{cont}_{\leftarrow}, a, \leftarrow)$	$(\text{cont}_{\leftarrow}, b, \leftarrow)$	$(\text{nok}, \mathbb{B}, \circ)$	$(\text{start}_{\rightarrow}, \#, \rightarrow)$
check	$(\text{nok}, a, \circ)$	$(\text{nok}, b, \circ)$	$(\text{ok}, \mathbb{B}, \circ)$	$(\text{ok}, \#, \circ)$

$$L(\mathcal{M}) = \{ a^n b a^n : n \in \mathbb{N} \}$$

Problemy rozstrzygalne

Język (problem) $L \subseteq A^*$ nazywamy

częściowo rozstrzygalnym, albo *rekurencyjnie przeliczalnym*,

jeśli $L = L(\mathcal{M})$ dla pewnej **deterministycznej** maszyny Turinga $\mathcal{M}$.

Język (problem) $L \subseteq A^*$ nazywamy

rozstrzygalnym, albo *rekurencyjnym*,

jeśli $L = L(\mathcal{M})$ dla pewnej **całkowitej deterministycznej** maszyny Turinga $\mathcal{M}$.

Pytanie

Czy języki bezkontekstowe są rozstrzygalne? A ich dopełnienia?

Fakt

Jeśli $L \subseteq A^$ jest rozstrzygalny to $\bar{L} = A^* - L$ też.*

Język uniwersalny (problem stopu):

$$L_u = \{ w_{\mathcal{M}} \$ w : w \in L(\mathcal{M}) \} \subseteq \{0, 1, \$\}^*$$

$$L_u = \{ (\mathcal{M}, w) : w \in L(\mathcal{M}) \}$$

Twierdzenie

Problem stopu jest częściowo rozstrzygalny, ale nie jest rozstrzygalny.

Dowód:

Przypuśćmy, że

$$L(\mathcal{M}_u) = L_u = \{ (w_{\mathcal{M}} \$ w) : w \in L(\mathcal{M}) \}$$

dla całkowitej deterministycznej maszyny $\mathcal{M}_u$.

Skonstruujemy (całkowitą deterministyczną) maszynę $\mathcal{M}_p$ dla języka

$$L_p = \{ w_{\mathcal{M}} : w_{\mathcal{M}} \notin L(\mathcal{M}) \}$$

Fakt

Jeśli L i $\bar{L}$ są częściowo rozstrzygalne, to są rozstrzygalne.

Dowód:

Z dwóch deterministycznych maszyn Turinga dla języków L i $\bar{L}$ skonstruujemy *całkowitą* deterministyczną maszynę dla $L \dots$

rysunek

Wniosek

Dla każdego języka L zachodzi dokładnie jeden z warunków:

- *L jest rozstrzygalny (i $\bar{L}$ też)*
- *L jest częściowo rozstrzygalny, $\bar{L}$ nie jest częściowo rozstrzygalny*
- *$\bar{L}$ jest częściowo rozstrzygalny, L nie jest częściowo rozstrzygalny*
- *L i $\bar{L}$ nie są częściowo rozstrzygalne*

rysunek

Inne przykładowe problemy częściowo rozstrzygalne, ale nierozstrzygalne:

Niepustość języka maszyny Turinga

Dane: maszyna Turinga $\mathcal{M}$
Wynik: czy $L(\mathcal{M}) \neq \emptyset$?

Nieuniwersalność języka bezkontekstowego

Dane: język bezkontekstowy $L \subseteq A^*$
Wynik: czy $L \neq A^*$?

Język uniwersalny (problem stopu):

$$L_u = \{ w_{\mathcal{M}} \$ w : w \in L(\mathcal{M}) \} \subseteq \{0, 1, \$\}^*$$

$$L_u = \{ (\mathcal{M}, w) : w \in L(\mathcal{M}) \}$$

Twierdzenie

Problem stopu jest częściowo rozstrzygalny, ale nie jest rozstrzygalny.

Wniosek

Istnieje deterministyczna maszyna $\mathcal{M}$ i słowo wejściowe w takie, że $w \notin L(\mathcal{M})$, ale nie da się tego dowiedzieć na gruncie teorii zbiorów.^a

^aZakładając niesprzeczność aksjomatów teorii zbiorów.

1 Teza Churcha-Turinga

2 Problem stopu

3 Rozstrzygalność

4 Obliczalność

Relację $R \subseteq (A^*)^n$ możemy utożsamić z językiem

$$L_R = \{ w_1 \$ w_2 \$ \dots \$ w_n : (w_1, w_2, \dots, w_n) \in R \}$$

Definicja 1: Funkcję częściową $f : (A^*)^n \rightarrow A^*$ nazywamy *częściowo obliczalną*, jeśli język

$$L_f = \{ w_1 \$ w_2 \$ \dots \$ w_n \$ f(w_1, w_2, \dots, w_n) : (w_1, w_2, \dots, w_n) \in \text{dom}(f) \}$$

jest częściowo rozstrzygalny.

Pytanie

No dobrze, ale jak *obliczyć* wartość funkcji?

Deterministyczna maszyna Turinga $\mathcal{M} = (A, Q, q_0, q_{\text{stop}}, T, \mathbb{B}, \delta)$

- $q_{\text{stop}} \in Q$ – stan końcowy
- $\delta : (Q - \{q_{\text{stop}}\}) \times T \rightarrow Q \times T \times \{\leftarrow, \circlearrowleft, \rightarrow\}$

Funkcja częściowa

$$F(\mathcal{M}) : (A^*)^n \rightarrow A^*$$

obliczana przez maszynę $\mathcal{M}$:

$$F(\mathcal{M})(w_1, \dots, w_n) = v \iff q_0 w_1 \$ w_2 \$ \dots \$ w_n \xrightarrow{*}_{\mathcal{M}} q_{\text{stop}} v$$

Definicja 2: Funkcję częściową $f : (A^*)^n \rightarrow A^*$ nazywamy *częściowo obliczalną*, jeśli $f = F(\mathcal{M})$ dla jakiejś maszyny $\mathcal{M}$.

Funkcja częściowa $f : (A^*)^n \rightarrow A^*$.

Definicja 1: L_f jest częściowo rozstrzygalny.

$$L_f = \{ w_1 \$ w_2 \$ \dots \$ w_n \$ f(w_1, w_2, \dots, w_n) : (w_1, w_2, \dots, w_n) \in \text{dom}(f) \}$$

Definicja 2: $f = F(\mathcal{M})$ dla jakiejś maszyny $\mathcal{M}$.

$$F(\mathcal{M})(w_1, \dots, w_n) = v \iff q_0 w_1 \$ w_2 \$ \dots \$ w_n \xrightarrow{*}_{\mathcal{M}} q_{\text{stop}} v$$

Pytanie

Czy te dwie definicje są równoważne?

Definicja:

Całkowitą, częściowo obliczalną funkcję $f : (A^*)^n \rightarrow A^*$ nazywamy *obliczalną*.

	deterministyczne maszyny Turinga	całkowite deterministyczne maszyny Turinga
języki (problemy decyzyjne)	częściowo rozstrzygalne	rozstrzygalne
funkcje (problemy obliczeniowe)	częściowo obliczalne	obliczalne

funkcje obliczane przez
maszyny Turinga

=

problemy obliczeniowe, dla których
istnieje efektywny algorytm, przy
założeniu nieograniczonych zasobów

albo:

maszyny Turinga

≡

komputery

teoria obliczeń

algorytmy

język
funkcja
maszyna Turinga
słowo wejściowe
uniwersalna maszyna Turinga
częściowo rozstrzygalny / obliczalna
rozstrzygalny / obliczalna

problem decyzyjny
problem obliczeniowy
algorytm / program / system komputerowy
dane wejściowe / instancja
interpreter programów
rozwiązywalny algorytmicznie
rozwiązywalny algorytmicznie, własność stopu

W następnym odcinku:

Nieobliczalność