

Języki, automaty i obliczenia

Wykład 11: Nieobliczalność

Sławomir Lasota

Uniwersytet Warszawski

14 maja 2025

- 1 Jak dowodzić nierozstrzygalności?
- 2 Problem odpowiedniości Posta
- 3 „Najprostszy” problem nierozstrzygalny

Problem $L \subseteq A^*$ *redukuje się (sprowadza się)* do problemu $K \subseteq B^*$ jeśli istnieje funkcja obliczalna

$$f : A^* \rightarrow B^*$$

taka, że

$$w \in L \iff f(w) \in K, \quad \text{dla każdego } w \in A^*.$$

rysunek

Ozn. $L \leq K$.

Fakt

Jeśli $L \leq K$ to $\bar{L} \leq \bar{K}$.

Fakt

Jeśli $L \leq K$ i problem K jest (częściowo) rozstrzygalny to problem L jest też (częściowo) rozstrzygalny.

Dowód:

...

rysunek

Niech $A = \{0, 1\}$.

Problem stopu

Dane: maszyna Turinga $\mathcal{M}$
i słowo w

Wynik: czy $w \in L(\mathcal{M})$?

$\leq$

Niepustość języka maszyny Turinga

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

funkcja obliczalna:

$\mathcal{M}, w \mapsto \mathcal{M}'$

poprawność:

$w \in L(\mathcal{M}) \iff L(\mathcal{M}') \neq \emptyset$

Maszyna $\mathcal{M}'$ działa następująco:

- ignoruje swoje słowo wejściowe
- pisze na taśmie słowo w
- symuluje maszynę M na słowie w
- akceptuje, gdy M akceptuje

Ta sama redukcja dowodzi:

Problem stopu

Dane: maszyna Turinga $\mathcal{M}$
i słowo w

Wynik: czy $w \in L(\mathcal{M})$?

$\leq$

Problem stopu na ε

Dane: maszyna Turinga $\mathcal{M}$

Wynik: czy $\varepsilon \in L(\mathcal{M})$?

Problem stopu

Dane: maszyna Turinga $\mathcal{M}$
i słowo w

Wynik: czy $w \in L(\mathcal{M})$?

$\leq$

Uniwersalność języka maszyny T.

Dane: maszyna Turinga $\mathcal{M}$

Wynik: czy $L(\mathcal{M}) = A^*$?

Problem stopu

Dane: maszyna Turinga $\mathcal{M}$
i słowo w

Wynik: czy $w \in L(\mathcal{M})$?



Uniwersalność języka maszyny T.

Dane: maszyna Turinga $\mathcal{M}$

Wynik: czy $L(\mathcal{M}) = A^*$?

Problem stopu

Dane: maszyna Turinga $\mathcal{M}$
i słowo w

Wynik: czy $w \in L(\mathcal{M})$?



Nieuniwersalność języka maszyny T.

Dane: maszyna Turinga $\mathcal{M}$

Wynik: czy $L(\mathcal{M}) \neq A^*$?

Niepustość języka maszyny Turinga

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

$\leq$

Nieuniwersalność języka bezkont.

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

funkcja obliczalna:	$\mathcal{M}$	$\mapsto$	$\mathcal{G}$
poprawność:	$L(\mathcal{M}) = \emptyset$	$\iff$	$L(\mathcal{G}) = A^*$

Bieg maszyny $\mathcal{M}$:
 $\rho = c_0 \rightarrow_{\mathcal{M}} c_1 \rightarrow_{\mathcal{M}} \dots \rightarrow_{\mathcal{M}} c_n$
 $w_\rho = \$c_0\$c_1\$ \dots \$c_n\$$

Niech $L(\mathcal{G}) = \overline{\{w_\rho : \rho \text{ bieg akceptujący maszyny } \mathcal{M}\}}$

Fakt

Język $\overline{\{c\$c' : c \rightarrow_{\mathcal{M}} c'\}}$ jest efektywnie bezkontekstowy.

Twierdzenie

Każdy problem częściowo rozstrzygalny redukuje się do problemu stopu.

Dowód:

Niech L częściowo rozstrzygalny i niech $L = L(\mathcal{M})$.

funkcja obliczalna:	w	$\mapsto$	$\mathcal{M}, w$
poprawność:	$w \in L$	$\iff$	$w \in L(\mathcal{M})$

Wniosek

Jeśli L jest częściowo rozstrzygalny oraz $(\text{problem stopu}) \leq L$ to L też jest zupełny.

- 1 Jak dowodzić nierozstrzygalności?
- 2 Problem odpowiedniości Posta
- 3 „Najprostszy” problem nierozstrzygalny

Problem odpowiedniości Posta (PCP)

Dane: ciąg par słów $(w_1, v_1), \dots, (w_n, v_n)$.

Wynik: czy istnieje niepusty ciąg $(i_1, \dots, i_m)$ t. że $w_{i_1} \dots w_{i_m} = v_{i_1} \dots v_{i_m}$?

Odpowiedni ciąg $(i_1, \dots, i_m)$ nazywamy *rozwiązaniem*.

Inne sformułowanie

Przykład

Instancja $(b, bbb), (babbb, ba), (ba, a)$

ma rozwiązanie $(2, 1, 1, 3)$:

$$babbb \ b \ b \ ba = ba \ bbb \ bbb \ a$$

a poniższa instancja nie ma rozwiązań:

$(ba, bab), (abb, bb), (bab, abb)$

1	b	bbb
2	babbb	ba
3	ba	a

1	ba	bab
2	abb	bb
3	bab	abb

Twierdzenie

Problem odpowiedniości Posta jest nierozstrzygalny.

Ograniczony problem odpowiedniości Posta (ograniczony PCP)

Dane: ciąg par słów $(w_1, v_1), \dots, (w_n, v_n)$.

Wynik: czy istnieje niepusty ciąg $(i_1, \dots, i_m)$ t. że $w_{i_1} \dots w_{i_m} = v_{i_1} \dots v_{i_m}$
i $i_1 = 1$?

Lemat

Ograniczony PCP $\leq$ PCP.

Dowód:

1	b	bbb		0	*b*	*b*b*b
2	babbb	ba		1	b*	*b*b*b
3	ba	a	$\mapsto$	2	b*a*b*b*b*	*b*a
				3	b*a*	*a
				4	\$	*\$

$(1, i_2, \dots, i_m)$ jest rozwiązaniem $\iff (0, i_2, \dots, i_m, n+1)$ jest rozwiązaniem

Lemat

Problem stopu $\leq$ ograniczony PCP.

Dowód:

Bieg akceptujący maszyny $\mathcal{M}$ na słowie w , czyli $q_0 w \rightarrow_{\mathcal{M}} c_1 \rightarrow_{\mathcal{M}} \dots \rightarrow_{\mathcal{M}} c_n$, odpowiada rozwiązaniu PCP postaci $\$q_0 w \$c_1\$ \dots \$c_n \$d_1\$ \dots \$d_j \$q_{\text{tak}} \$\$$.

$\mathcal{M}, w$

$\mapsto$

	\$	$\$q_0 w \$$
	0	0
	1	1
	$\mathbb{B}$	$\mathbb{B}$
	\$	\$
	$q a$	$q' a'$
	$q a$	$a' q'$
	$q \$$	$a' q' \$$
	$b q a$	$q' b a'$
	$\$ q a$	$\$ q' \mathbb{B} a'$
	...	...
	$q_{\text{tak}} a$	q_{tak}
	$a q_{\text{tak}}$	q_{tak}
	$q_{\text{tak}} \$ \$$	$\$$

jeśli $(q, a, q', a', \circ) \in \delta$

jeśli $(q, a, q', a', \rightarrow) \in \delta$

jeśli $(q, \mathbb{B}, q', a', \rightarrow) \in \delta$

jeśli $(q, a, q', a', \leftarrow) \in \delta$

jeśli $(q, a, q', a', \leftarrow) \in \delta$

- 1 Jak dowodzić nierozstrzygalności?
- 2 Problem odpowiedniości Posta
- 3 „Najprostszy” problem nierozstrzygalny

- 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
- ...

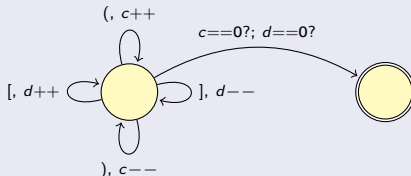
$$\mathcal{A} = (A, Q, q_0, q_{\text{stop}}, C, \delta)$$

- A – alfabet
- Q – skończony zbiór stanów
- q_0 – stan początkowy
- q_{stop} – stan akceptujący
- C – skończony zbiór liczników
- $\delta \subseteq (Q - \{q_{\text{stop}}\}) \times (A \cup \{\varepsilon\}) \times \{c++, c--, c==0? : c \in C\} \times Q$

Przykład

$$A = \{ (,), [,] \}$$

$$C = \{c, d\}$$



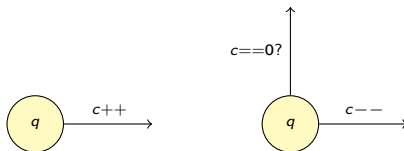
$$L(\mathcal{A}) = ?$$

$$\text{Konfiguracje} = Q \times \mathbb{N}^C$$

Automaty bez wejścia: $\delta \subseteq (Q - \{q_{\text{stop}}\}) \times \{c++, c--, c==0? : c \in C\} \times Q$.

Pytanie

Kiedy automat z licznikami jest deterministyczny?

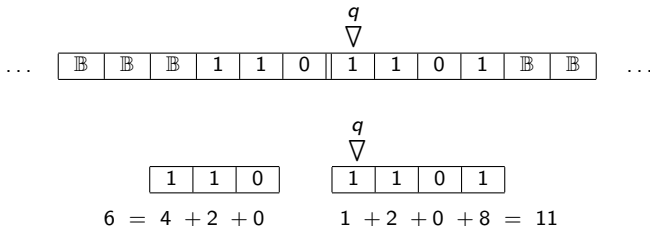


Twierdzenie

(Deterministyczne) maszyny Turinga $\equiv$ (deterministyczne) automaty z licznikami.

Dowód (idea):

Maszyna Turinga $\rightsquigarrow$ automat z 3 licznikami:

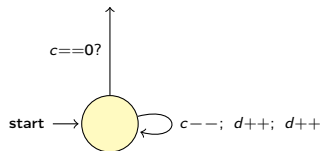
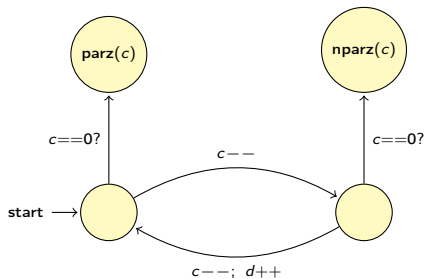


Potrzebne operacje na licznikach:

parz(c)? nparz(c)? $c = c \gg 1$ $c = c \ll 1$

Potrzebne operacje na licznikach:

$\text{parz}(c)?$ $\text{nparz}(c)?$ $c = c >> 1$ $c = c << 1$



Jeśli maszyna deterministyczna to automat też.

Twierdzenie

Automaty z 2 licznikami potrafią symulować automaty z 3 licznikami.

Dowód (idea):

3 liczniki c, d, e symulujemy za pomocą 2 liczników x, y :

$$x = 2^c \cdot 3^d \cdot 5^e$$

Pytanie

Czy automaty z 1 licznikiem potrafią symulować automaty z 2 licznikami?

„Najprostszy” problem nierozstrzygalny

Dane: $\mathcal{A}$ – automat deterministyczny z 2 licznikami c_1, c_2 bez wejścia

Wynik: czy $\mathcal{A}$ zatrzyma się, jeśli rozpocznie w konfiguracji $(q_0, c_1 = 0, c_2 = 0)$?

W następnym odcinku:

Gramatyki równoważne maszynom Turinga