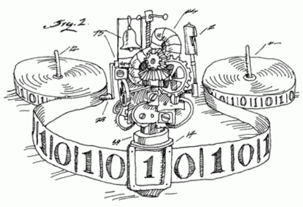
Języki, automaty i obliczenia

Wykład 9: Maszyny Turinga

Sławomir Lasota

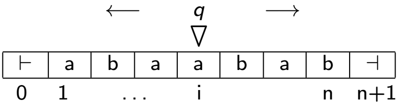
Uniwersytet Warszawski

8 maja 2024



Plan

Automaty dwukierunkowe



Automaty dwukierunkowe to. . . maszyny Turinga z taśmą tylko do odczytu.

Maszyny Turinga

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

- A – alfabet wejściowy
- Q – zbiór stanów
- $q_0 \in Q$ – stan początkowy
- $q_{tak} \in Q$ – stan akceptujący (bez u.o. jeden stan akceptujący)
- T – alfabet taśmowy, $A \subseteq T$
- $\mathbb{B} \in T - A$ – symbol pusty (ang. blank)
- $\delta \subseteq (Q - \{q_{tak}\}) \times T \times Q \times T \times \{\leftarrow, \circlearrowright, \rightarrow\}$ – relacja przejścia

Co oznacza przejście?

$(q, a, q', a', k) \in \delta$: zmień stan z q na q', czytaj a, zapisz a', zmień pozycję wg. k

Konfiguracja początkowa maszyny $\mathcal{M}$ wygląda tak:

$\longleftarrow \quad q_0 \quad \longrightarrow$

∇

...	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	a	b	a	a	b	a	b	$\mathbb{B}$	$\mathbb{B}$	...

Przykład

- $A \;=\; \{a, b\}$
- $Q \;=\; \{\mathsf{start}, \mathsf{start}_a, \mathsf{start}_b, \mathsf{ret}, \mathsf{go}, \mathsf{go}_a, \mathsf{go}_b, \mathsf{go}'_a, \mathsf{go}'_b, \mathsf{ret}', \mathsf{check}, \mathsf{ok}\}$
- $q_0 \;=\; \mathsf{start}$
- $q_{\mathsf{tak}} \;=\; \mathsf{ok}$
- $T \;=\; A \cup \{\mathbb{B}, \#\}$
- relacja przejścia (na następnym slajdzie)

$$\delta \;\subseteq\; (Q - \{\mathsf{ok}\}) \times T \times Q \times T \times \{\leftarrow, \circlearrowright, \rightarrow\}$$

$$\delta : (Q - \{\mathsf{ok}\}) \times T \;\to\; \mathcal{P}(Q \times T \times \{\leftarrow, \circlearrowright, \rightarrow\})$$

Pytanie

Jaki język rozpoznaje ta maszyna?

Przykład (relacja przejścia)

	a	b	$\mathbb{B}$	$\#$
start	$(\mathsf{start}_a, \#, \rightarrow)$	$(\mathsf{start}_b, \#, \rightarrow)$	$(\mathsf{ok}, \mathbb{B}, \circlearrowleft)$	
start_a	$(\mathsf{start}_a, a, \rightarrow)$ $(\mathsf{ret}, \#, \leftarrow)$	$(\mathsf{start}_a, b, \rightarrow)$		
start_b	$(\mathsf{start}_b, a, \rightarrow)$	$(\mathsf{start}_b, b, \rightarrow)$ $(\mathsf{ret}, \#, \leftarrow)$		
ret	$(\mathsf{ret}, a, \leftarrow)$	$(\mathsf{ret}, b, \leftarrow)$		$(\mathsf{go}, \#, \rightarrow)$
go	$(\mathsf{go}_a, \#, \rightarrow)$	$(\mathsf{go}_b, \#, \rightarrow)$		$(\mathsf{check}, \#, \circlearrowleft)$
go_a	$(\mathsf{go}_a, a, \rightarrow)$	$(\mathsf{go}_a, b, \rightarrow)$		$(\mathsf{go}'_a, \#, \rightarrow)$
go_b	$(\mathsf{go}_b, a, \rightarrow)$	$(\mathsf{go}_b, b, \rightarrow)$		$(\mathsf{go}'_b, \#, \rightarrow)$
go'_a	$(\mathsf{ret}', \#, \leftarrow)$			$(\mathsf{go}'_a, \#, \rightarrow)$
go'_b		$(\mathsf{ret}', \#, \leftarrow)$		$(\mathsf{go}'_b, \#, \rightarrow)$
ret'	$(\mathsf{ret}, a, \leftarrow)$	$(\mathsf{ret}, b, \leftarrow)$	$(\mathsf{check}, \mathbb{B}, \rightarrow)$	$(\mathsf{ret}', \#, \leftarrow)$
check			$(\mathsf{ok}, \mathbb{B}, \circlearrowleft)$	$(\mathsf{check}, \#, \rightarrow)$

Przykład (język)

$$L(\mathcal{M}) \;=\; \{\, w\,w \;:\; w \in A^* \,\}$$

Pytanie

Jak zmodyfikować tę maszynę, aby *obliczała* funkcję:

$$w \;\mapsto\; w\,w?$$

Przykład (funkcja)

$F(\mathcal{M}) \ : \ w \ \mapsto \ w\,w$ $\qquad T = \{a, b, \hat{a}, \hat{b}, \underline{a}, \underline{b}, \mathbb{B}\}$

Maszyna $\mathcal{M}$

powtarzaj, w zależności od symbolu $x \in \{a, b\}$ pod głowicą:

- $x \in \{a, b\}$
  - zapamiętaj x w stanie i zaznacz ją na taśmie $\hat{x}$
  - idź w prawo do pierwszego symbolu $\mathbb{B}$
  - zapisz tam podkreśloną zapamiętaną literę $\underline{x}$
  - idź w lewo do pierwszej zaznaczonej litery,
  - usuń zaznaczenie, przesuń się o jedną pozycję w prawo
- $x \in \{\underline{a}, \underline{b}\}$
  - idź w prawo, usuwając podkreślenia, do pierwszego symbolu $\mathbb{B}$, i zakończ
- $x = \mathbb{B}$
  - zakończ

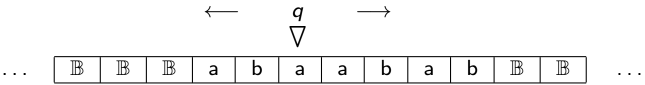
Taśma

...	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	a	b	a	a	b	a	b	$\mathbb{B}$	$\mathbb{B}$	...

- Taśma jest nieskończona w obydwie strony.
- Zawartość taśmy reprezentuje zawartość pamięci maszyny: nie nieskończoną, ale dowolnie dużą skończoną.
- Bez u.o. możemy założyć, że maszyna nigdy nie zapisuje symbolu $\mathbb{B}$; wtedy $\mathbb{B}$ oznacza „nieużywane” pozycje taśmy.
- Poza obszarem używanym przez maszynę, na wszystkich pozycjach taśmy jest symbol $\mathbb{B}$.
- Nieistotne, na której pozycji taśmy zaczyna się obszar używany przez maszynę (przesunięcie tego obszaru w prawo lub w lewo nic istotnego nie zmienia).

Konfiguracje

Zapisując konfiguracje, pomijamy nieskończenie wiele symboli $\mathbb{B}$ poza obszarem odwiedzonym przez maszynę, i poza słowem wejściowym:



$$a\,b\,(q,a)\,a\,b\,a\,b \qquad \text{albo} \qquad a\,b\,q\,a\,a\,b\,a\,b$$

Formalnie, *konfiguracja* maszyny $\mathcal{M}$ to

$$c \;=\; w\,q\,w' \in T^*\,Q\,T^*$$

Umowa

$a\,b\,q\,a\,a\,b\,a\,b \;=\; \mathbb{B}\,a\,b\,q\,a\,a\,b\,a\,b \;=\; a\,b\,q\,a\,a\,b\,a\,b\,\mathbb{B} \;=\; \mathbb{B}\mathbb{B}\,a\,b\,q\,a\,a\,b\,a\,b\,\mathbb{B} \;=\; \ldots$

Konfiguracje początkowa: $c_0 \;=\; q_0\,w$ ($w \in A^*$ to słowo wejściowe)

Konfiguracje akceptujące: $T^*\{q_{\mathrm{tak}}\}\,T^* \;=\; \{\,w\,q_{\mathrm{tak}}\,w' \;:\; w, w' \in T^*\,\}$

Język rozpoznawany przez maszynę

Przejścia pomiędzy konfiguracjami $c \longrightarrow_{\mathcal{M}} c'$: $(w, v \in T^*)$

- jeśli $(q, a, q', a', \circlearrowleft) \in \delta$ to

$$w\, q\, a\, v \longrightarrow_{\mathcal{M}} w\, q'\, a'\, v$$

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

$$w\, q\, a\, v \longrightarrow_{\mathcal{M}} w\, a'\, q'\, v$$

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

$$w\, b\, q\, a\, v \longrightarrow_{\mathcal{M}} w\, q'\, b\, a'\, v$$

Umowa

$$a\,b\,q\,a\,a\,b\,a\,b \;=\; \mathbb{B}\,a\,b\,q\,a\,a\,b\,a\,b \;=\; a\,b\,q\,a\,a\,b\,a\,b\,\mathbb{B} \;=\; \mathbb{B}\,\mathbb{B}\,a\,b\,q\,a\,a\,b\,a\,b\,\mathbb{B} \;=\; \ldots$$

Język rozpoznawany przez maszynę $\mathcal{M}$:

$$L(\mathcal{M}) \;=\; \{\, w \in A^* \;:\; q_0\, w \longrightarrow^*_{\mathcal{M}} c \text{ dla jakiejś konfiguracji akceptującej } c \,\}$$

Języki częściowo rozstrzygalne

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

częściowo rozstrzygalnym, albo *rekurencyjnie przeliczalnym*,

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

Hierarchia Chomsky'ego:

- (typ 0) języki częściowo rozstrzygalne
- (typ 1) języki kontekstowe
- (typ 2) języki bezkontekstowe
- (typ 3) języki regularne

Plan

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

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

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
- $\delta : (Q - \{q_{\text{tak}}, q_{\text{nie}}\}) \times T \to Q \times T \times \{\leftarrow, \circlearrowleft, \rightarrow\}$

Przykład (maszyna deterministyczna)

- $q_{\mathsf{tak}} = \mathsf{ok}$
- $q_{\mathsf{nie}} = \mathsf{nok}$
- $\delta : (Q - \{\mathsf{ok}, \mathsf{nok}\}) \times T \to Q \times T \times \{\leftarrow, \circlearrowright, \to\}$

	a	b	$\mathbb{B}$	$\#$
$\mathsf{start}_{\to}$	$(\mathsf{cont}_{\to}, \#, \to)$	$(\mathsf{check}, b, \to)$	$(\mathsf{nok}, \mathbb{B}, \circlearrowright)$	$(\mathsf{nok}, \#, \circlearrowright)$
$\mathsf{cont}_{\to}$	$(\mathsf{cont}_{\to}, a, \to)$	$(\mathsf{cont}_{\to}, b, \to)$	$(\mathsf{start}_{\leftarrow}, \mathbb{B}, \leftarrow)$	$(\mathsf{start}_{\leftarrow}, \#, \leftarrow)$
$\mathsf{start}_{\leftarrow}$	$(\mathsf{cont}_{\leftarrow}, \#, \leftarrow)$	$(\mathsf{nok}, b, \circlearrowright)$	$(\mathsf{nok}, \mathbb{B}, \circlearrowright)$	$(\mathsf{nok}, \#, \circlearrowright)$
$\mathsf{cont}_{\leftarrow}$	$(\mathsf{cont}_{\leftarrow}, a, \leftarrow)$	$(\mathsf{cont}_{\leftarrow}, b, \leftarrow)$	$(\mathsf{nok}, \mathbb{B}, \circlearrowright)$	$(\mathsf{start}_{\to}, \#, \to)$
check	$(\mathsf{nok}, a, \circlearrowright)$	$(\mathsf{nok}, b, \circlearrowright)$	$(\mathsf{ok}, \mathbb{B}, \circlearrowright)$	$(\mathsf{ok}, \#, \circlearrowright)$

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

Czy tę maszynę można zdeterminizować?

	a	b	$\mathbb{B}$	$\#$
start	$(\text{start}_a, \#, \rightarrow)$	$(\text{start}_b, \#, \rightarrow)$	$(\text{ok}, \mathbb{B}, \circlearrowleft)$	
start_a	$(\text{start}_a, a, \rightarrow)$ $(\text{ret}, \#, \leftarrow)$	$(\text{start}_a, b, \rightarrow)$		
start_b	$(\text{start}_b, a, \rightarrow)$	$(\text{start}_b, b, \rightarrow)$ $(\text{ret}, \#, \leftarrow)$		
ret	$(\text{ret}, a, \leftarrow)$	$(\text{ret}, b, \leftarrow)$		$(\text{go}, \#, \rightarrow)$
go	$(\text{go}_a, \#, \rightarrow)$	$(\text{go}_b, \#, \rightarrow)$		$(\text{check}, \#, \circlearrowleft)$
go_a	$(\text{go}_a, a, \rightarrow)$	$(\text{go}_a, b, \rightarrow)$		$(\text{go}'_a, \#, \rightarrow)$
go_b	$(\text{go}_b, a, \rightarrow)$	$(\text{go}_b, b, \rightarrow)$		$(\text{go}'_b, \#, \rightarrow)$
go'_a	$(\text{ret}', \#, \leftarrow)$			$(\text{go}'_a, \#, \rightarrow)$
go'_b		$(\text{ret}', \#, \leftarrow)$		$(\text{go}'_b, \#, \rightarrow)$
ret'	$(\text{ret}, a, \leftarrow)$	$(\text{ret}, b, \leftarrow)$	$(\text{check}, \mathbb{B}, \rightarrow)$	$(\text{ret}', \#, \leftarrow)$
check			$(\text{ok}, \mathbb{B}, \circlearrowleft)$	$(\text{check}, \#, \rightarrow)$

Determinizacja?

Pytanie

Czy dla każdej maszyny Turinga istnieje równoważna maszyna deterministyczna?

Twierdzenie

Dla każdej maszyny Turinga istnieje równoważna maszyna deterministyczna.

Dowód:

...	$\mathbb{B}$	q, a	a	b	a	b	$\#$	b	q', a	b	a	b	$\#$	b	...

Wniosek

Możemy ograniczyć się do maszyn deterministycznych.

Plan

Warianty maszyn Turinga

Maszyny z:

- taśmą jednostronnie nieskończoną
- wieloma taśmami
- „taśmą” wielowymiarową
- ...

Maszyny z taśmą jednostronnie nieskończoną

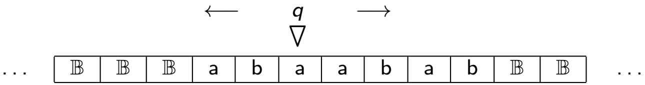
$\vdash$	a	b	a	a	b	a	b	$\mathbb{B}$	$\mathbb{B}$	...

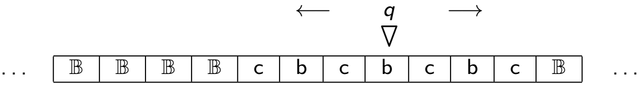
Pytanie

Czy maszyny z taśmą jednostronnie nieskończoną potrafią symulować maszyny z taśmą dwustronnie nieskończoną?

...	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	a	b	a	a	b	a	b	$\mathbb{B}$	$\mathbb{B}$	...

$\vdash$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$				...
	a	b	a	a	b	a	b	$\mathbb{B}$	$\mathbb{B}$		

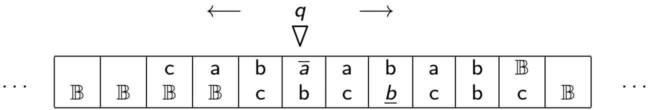




$$\delta \subseteq Q \times T^2 \times Q \times T^2 \times \{\leftarrow, \circlearrowleft, \rightarrow\}^2$$

Pytanie

Czy maszyny jednotaśmowe potrafią symulować maszyny wielotaśmowe?



Maszyny z taśmą dwuwymiarową

...

	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	
	$\mathbb{B}$	$\mathbb{B}$	a	b	b	c	q	a	a	a	$\mathbb{B}$	$\mathbb{B}$	
...	$\mathbb{B}$	a	b	a	a	b	∇	a	a	b	$\mathbb{B}$	$\mathbb{B}$	...
	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	a	b	a	a	b	a	b	$\mathbb{B}$	$\mathbb{B}$	
	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	c	b	c	b	c	b	c	$\mathbb{B}$	
	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	$\mathbb{B}$	

...

$$\delta \subseteq Q \times T \times Q \times T \times \{\leftarrow, \circlearrowleft, \rightarrow, \uparrow, \downarrow\}$$

Pytanie

Czy maszyny z taśmą jednowymiarową potrafią symulować maszyny z taśmą dwuwymiarową?

...	a	$\overline{\mathbb{B}}$	#	$\overline{\mathbb{B}}$	a	b	b	c	a	$\overline{\mathbb{B}}$	#	$\overline{\mathbb{B}}$	$\overline{\mathbb{B}}$	a	c	...

Modele obliczeń równoważne maszynom Turinga

- 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
- ...

W następnym odcinku:

Rozstrzygalność i obliczalność