

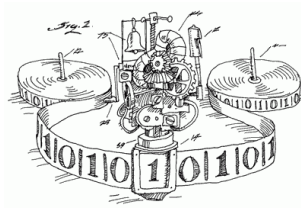
Języki, automaty i obliczenia

Wykład 9: Maszyny Turinga

Sławomir Lasota

Uniwersytet Warszawski

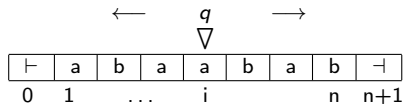
23 kwietnia 2025



1 Maszyny Turinga

2 Deterministyczne maszyny Turinga

3 Warianty maszyn Turinga



Automaty dwukierunkowe to...

maszyny Turinga z taśmą tylko do odczytu.

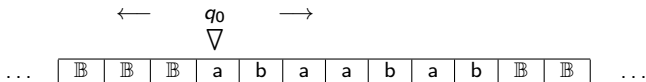
(Niedeterministyczna) *maszyna Turinga* $\mathcal{M} = (A, Q, q_0, q_{\text{tak}}, T, \mathbb{B}, \delta)$

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

Co oznacza przejście?

(q, a, q', a', k) : w stanie q czytaj a , zmień stan na q' , **zapisz a'** , zmień pozycję wg. k

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



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

$$\delta \subseteq (Q - \{\text{ok}\}) \times T \times Q \times T \times \{\leftarrow, \circ, \rightarrow\}$$

$$\delta : (Q - \{\text{ok}\}) \times T \rightarrow \mathcal{P}(Q \times T \times \{\leftarrow, \circ, \rightarrow\})$$

Pytanie

Jaki język rozpoznaje ta maszyna?

Przykład (relacja przejścia)

	a	b	$\mathbb{B}$	$\#$
start	$(\text{start}_a, \#, \rightarrow)$	$(\text{start}_b, \#, \rightarrow)$	$(\text{ok}, \mathbb{B}, \circ)$	
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	$(\text{go}_a, \#, \rightarrow)$	$(\text{go}_b, \#, \rightarrow)$		$(\text{check}, \#, \circ)$
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			$(\text{ok}, \mathbb{B}, \circ)$	$(\text{check}, \#, \rightarrow)$

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

Pytanie

Jak zmodyfikować tę maszynę, aby *obliczała* funkcję:

$$w \mapsto w w?$$

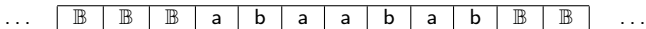
$$F(\mathcal{M}) : w \mapsto ww$$

$$T = \{a, b, \hat{a}, \hat{b}, \underline{a}, \underline{b}, \mathbb{B}\}$$

Maszyna $\mathcal{M}$

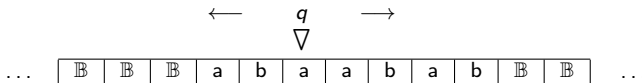
powtarzaj, w zależności od symbolu $x \in \{a, b\}$ pod głowicą:

- $x \in \{a, b\}$
 - 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, przesunij 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 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 B; wtedy B oznacza „nieużywane” pozycje taśmy.
- Poza obszarem używanym przez maszynę, na wszystkich pozycjach taśmy jest symbol B.
- Nieistotne, na której pozycji taśmy zaczyna się obszar używany przez maszynę (przesunięcie tego obszaru w prawo lub w lewo nic istotnego nie zmienia).

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



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

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

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

Umowa

$$a b q a a b a b = \mathbb{B} a b q a a b a b = a b q a a b a b \mathbb{B} = \mathbb{B} \mathbb{B} a b q a a b a b \mathbb{B} = \dots$$

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

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

Przejścia pomiędzy konfiguracjami $c \rightarrow_{\mathcal{M}} c'$:

$(w, v \in T^*)$

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

$$w q a v \rightarrow_{\mathcal{M}} w q' a' v$$

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

$$w q a v \rightarrow_{\mathcal{M}} w a' q' v$$

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

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

Umowa

$$abqaaabab = \mathbb{B}abqaaabab = abqaaabab\mathbb{B} = \mathbb{B}\mathbb{B}abqaaabab\mathbb{B} = \dots$$

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

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

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

1 Maszyny Turinga

2 Deterministyczne maszyny Turinga

3 Warianty maszyn Turinga

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

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

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

Przykład (maszyna deterministyczna)

- $q_{\text{tak}} = \text{ok}$
- $q_{\text{nie}} = \text{nok}$
- $\delta : (Q - \{\text{ok}, \text{nok}\}) \times T \rightarrow Q \times T \times \{\leftarrow, \circ, \rightarrow\}$

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

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

Czy tę maszynę można zdeterminizować?

	a	b	$\mathbb{B}$	$\#$
start	$(\text{start}_a, \#, \rightarrow)$	$(\text{start}_b, \#, \rightarrow)$	$(\text{ok}, \mathbb{B}, \circ)$	
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	$(\text{go}_a, \#, \rightarrow)$	$(\text{go}_b, \#, \rightarrow)$		$(\text{check}, \#, \circ)$
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			$(\text{ok}, \mathbb{B}, \circ)$	$(\text{check}, \#, \rightarrow)$

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.

- 1 Maszyny Turinga
- 2 Deterministyczne maszyny Turinga
- 3 **Warianty maszyn Turinga**

Maszyny z:

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

Maszyny z taśmą jednostronnie nieskończoną

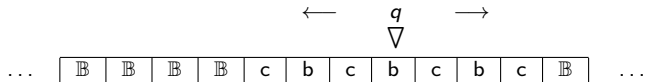
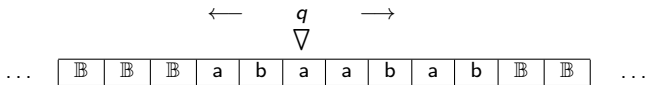
⊢	a	b	a	a	b	a	b	B	B	...
---	---	---	---	---	---	---	---	---	---	-----

Pytanie

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

...	B	B	B	a	b	a	a	b	a	b	B	B	...
-----	---	---	---	---	---	---	---	---	---	---	---	---	-----

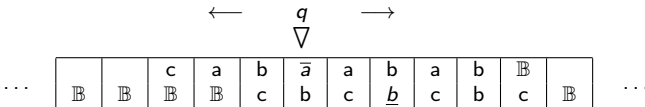
⊢	B	B	B	B	B	B	B	B			...
	a	b	a	a	b	a	b	B	B		



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

Pytanie

Czy maszyny jednotaśmowe potrafią symulować maszyny wielotaśmowe?



Maszyny z taśmą dwuwymiarową

...

	B	B	B	B	B	B	B	B	B	B	B	
	B	B	a	b	b	c	q ▽	a	a	a	B	B
...	B	a	b	a	a	b		a	a	b	B	B
	B	B	B	a	b	a		b	a	b	B	B
	B	B	B	B	c	b	c	b	c	b	c	B
	B	B	B	B	B	B	B	B	B	B	B	

...

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

Pytanie

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

...

a	$\bar{B}$	#	$\bar{B}$	a	b	b	c	a	$\bar{B}$	#	$\bar{B}$	$\bar{B}$	a	c
---	-----------	---	-----------	---	---	---	---	---	-----------	---	-----------	-----------	---	---

 ...

- 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