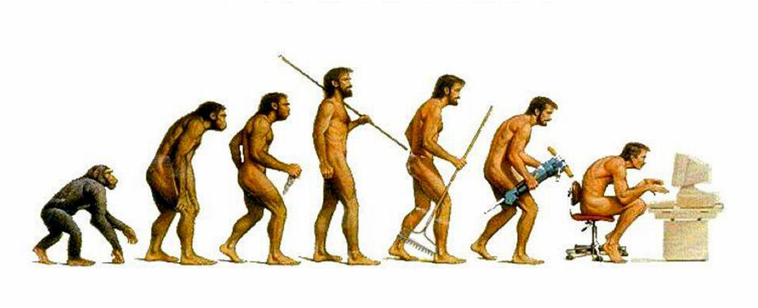


Teoria Obliczalności

Marcin Piątkowski

Wykład 6

Funkcje, programy i maszyny uniwersalne

s-m-n twierdzenie

Dla dowolnych $m, n \in \mathbb{N}$ $(m, n \geq 1)$ istnieje totalna i obliczalna funkcja $\mathbf{s}^{\mathbf{m}}_{\mathbf{n}} : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ taka, że dla dowolnych $\mathbf{e} \in \mathbb{N}$, $\overline{\mathbf{x}} \in \mathbb{N}^n$ oraz $\overline{y} \in \mathbb{N}^m$ zachodzi:

$$\phi^{(m+n)}_{\mathbf{e}}(\overline{\mathbf{x}}, \overline{y}) \;=\; \phi^{(m)}_{\mathbf{s}^{\mathbf{m}}_{\mathbf{n}}(\mathbf{e}, \overline{\mathbf{x}})}(\overline{y})$$

Twierdzenie o parametryzacji

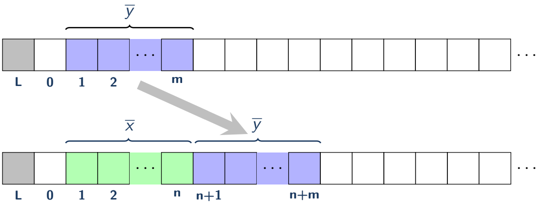
s-m-n twierdzenie

Dla dowolnych $m, n \in \mathbb{N}$ $(m, n \geq 1)$ istnieje totalna i obliczalna funkcja $\mathbf{s_n^m} : \mathbb{N}^{n+1} \to \mathbb{N}$ taka, że dla dowolnych $\mathbf{e} \in \mathbb{N}$, $\overline{\mathbf{x}} \in \mathbb{N}^n$ oraz $\overline{y} \in \mathbb{N}^m$ zachodzi:

$$\phi_{\mathbf{e}}^{(m+n)}(\overline{\mathbf{x}}, \overline{y}) \;=\; \phi^{(m)}_{\mathbf{s_n^m}(\mathbf{e}, \overline{\mathbf{x}})}(\overline{y})$$

$$\begin{tikzcd}
f(x,y) = x + y \arrow[r, squiggly] \arrow[d] & g(y) = f(\mathbf{5}, y) = \mathbf{5} + y \arrow[d] \\
\phi_{\mathbf{e}}^{(2)}(x,y) \arrow[r, squiggly] & \phi^{(1)}_{\mathbf{s_1^1}(\mathbf{e},\mathbf{5})}(y)
\end{tikzcd}$$

Twierdzenie o parametryzacji



P_s

`T(m,n+m)`
⋮
`T(1,n+1)` — // kopiowanie $\overline{y}$

`Z(1)`
⋮
`Z(n)` — // przygotowanie miejca na $\overline{x}$

`S(1)` — // x_1 razy
⋮
`S(n)` — // x_n razy

$\mathbf{P}'_{\mathbf{e}}$ — // P_e + modyfikacja skoków

$$
\begin{tikzcd}
\phi_{\mathbf{e}}^{(m+n)}(\overline{\mathbf{x}}, \overline{y}) \arrow[r] \arrow[d, dashed] & P_{\mathbf{e}} \arrow[d] \\
\phi_{\mathbf{s}_{\mathbf{n}}^{\mathbf{m}}(\mathbf{e}, \overline{\mathbf{x}})}^{(m)}(\overline{y}) \arrow[r] & P_{\mathbf{s}}
\end{tikzcd}
$$

$$2^{x+1} \cdot \overbrace{\left(2^{a_0} + 2^{a_0+a_1+1} + \ldots + 2^{a_0+a_1+\ldots+a_n+n}\right)}^{\tau(a_0, a_1, \ldots, a_n) + 1} + 2^{x}$$

$$\Vert$$

$$\underbrace{2^{x} + 2^{x+a_0+1} + 2^{x+a_0+a_1+2} + \ldots + 2^{x+a_0+a_1+\ldots+a_n+n+1}}_{\tau(x, a_0, a_1, \ldots, a_n) + 1}$$

$$\begin{tikzcd}
\boxed{\tau(a_0, a_1, \ldots, a_n) + 1} \arrow[r, bend left, "\texttt{push(x)}"] & \boxed{\tau(x, a_0, a_1, \ldots, a_n) + 1} \arrow[l, bend left, "\texttt{pop()}"]
\end{tikzcd}$$

Funkcją uniwersalną dla n-argumentowych funkcji obliczalnych nazywamy funkcję $\Psi_U^{(n)} : \mathbf{N}^{n+1} \longrightarrow \mathbf{N}$ taką, że:

$$\forall_{\mathbf{e} \in \mathbf{N}} \forall_{\overline{\mathbf{x}} \in \mathbf{N}^n} \Psi_U^{(n)}(\mathbf{e}, \overline{\mathbf{x}}) \simeq \phi_{\mathbf{e}}^{(n)}(\overline{\mathbf{x}})$$

Funkcją uniwersalną dla n-argumentowych funkcji obliczalnych nazywamy funkcję $\Psi_U^{(n)} : \mathbf{N}^{n+1} \longrightarrow \mathbf{N}$ taką, że:

$$\forall_{\mathbf{e} \in \mathbf{N}} \; \forall_{\overline{\mathbf{x}} \in \mathbf{N}^n} \; \Psi_U^{(n)}(\mathbf{e}, \overline{\mathbf{x}}) \;\simeq\; \phi_{\mathbf{e}}^{(n)}(\overline{\mathbf{x}})$$

Twierdzenie

Każda funkcja uniwersalna $\Psi_U^{(n)}$ jest **częściowo rekurencyjna**

? Uzasadnij, że funkcja uniwersalna jest częściowo rekurencyjna

Funkcją uniwersalną dla n-argumentowych funkcji obliczalnych nazywamy funkcję $\Psi_U^{(n)} : \boldsymbol{N}^{n+1} \longrightarrow \boldsymbol{N}$ taką, że:

$$\forall_{\mathbf{e}\in \boldsymbol{N}}\ \forall_{\overline{\mathbf{x}}\in \boldsymbol{N}^n}\ \Psi_U^{(n)}(\mathbf{e}, \overline{\mathbf{x}}) \simeq \phi_{\mathbf{e}}^{(n)}(\overline{\mathbf{x}})$$

Obliczalna

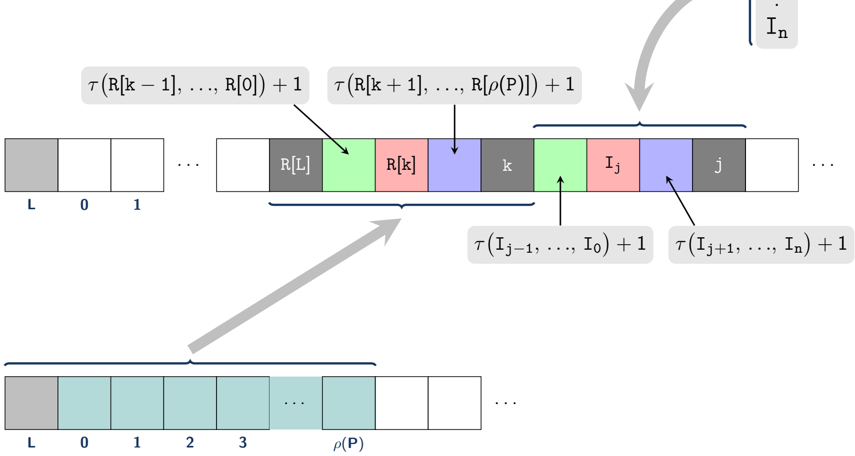
Twierdzenie

Każda funkcja uniwersalna $\Psi_U^{(n)}$ jest **częściowo rekurencyjna**

Program uniwersalny – dowolny program P_U obliczający funkcję uniwersalną Ψ_U

Program uniwersalny

$$\phi_{\mathbf{e}}^{(n)}(\overline{\mathtt{x}}) \rightsquigarrow P_{\mathbf{e}}\langle \overline{\mathtt{x}} \rangle$$



Uniwersalna maszyna Turinga

- $\mathbb{U}$ — uniwersalna maszyna Turinga
- $\langle \mathbf{M} \rangle$ – reprezentacja maszyny **M**
- $\mathbb{U}(\langle \mathbf{M} \rangle, x)$ — symulacja działania maszyny **M** na danych x

Szczegóły reprezentacji $\langle \mathrm{M} \rangle$

- Reprezentacja zbioru stanów maszyny **M**
- Reprezentacja alfabetu taśmy maszyny **M**
- Kodowanie funkcji przejścia maszyny **M**

Uniwersalna maszyna Turinga

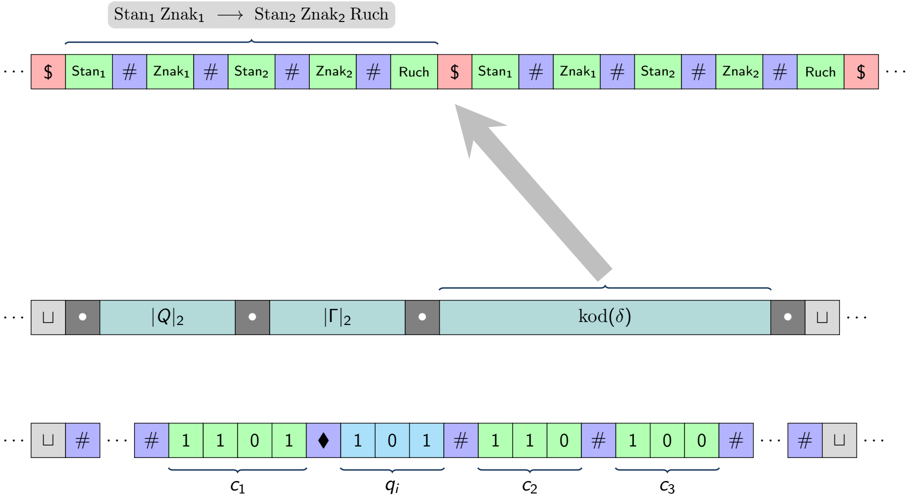
$$
\begin{aligned}
Q &\rightsquigarrow \{1, 2, \ldots, |Q|\} \\
\Gamma &\rightsquigarrow \{1, 2, \ldots, |\Gamma|\} \\
q_0 &\rightsquigarrow 1 \\
q_{ACC} &\rightsquigarrow |Q| \\
q_{REJ} &\rightsquigarrow |Q| - 1
\end{aligned}
$$

$\langle M \rangle$

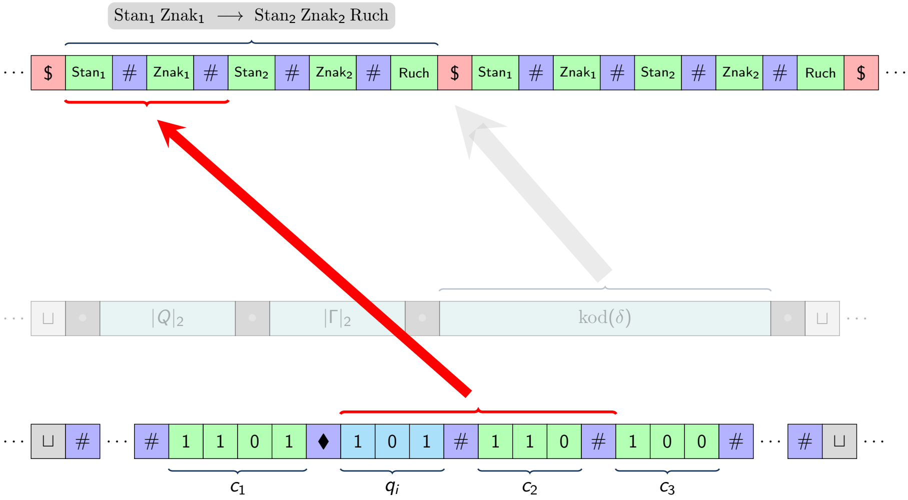
$\cdots$	$\sqcup$	$\bullet$	$\lvert Q\rvert_2$	$\bullet$	$\lvert\Gamma\rvert_2$	$\bullet$	$\mathrm{kod}(\delta)$	$\bullet$	$\sqcup$	$\cdots$

$\cdots$	$\sqcup$	#	$\cdots$	#	1	1	0	1	$\blacklozenge$	1	0	1	#	1	1	0	#	1	0	0	#	$\cdots$	#	$\sqcup$	$\cdots$		
					c_1					q_i				c_2				c_3									

Uniwersalna maszyna Turinga



Uniwersalna maszyna Turinga



Problem akceptowania wejścia

Problem

Weryfikacja poprawności działania programu

Problem akceptowania wejścia

$$\mathcal{P}_{ACC} = \Big\{ \big(M, x \big) : \text{ Maszyna Turinga } M \text{ akceptuje dane wejciowe } x \Big\}$$

Problem akceptowania wejścia

$$\mathcal{P}_{ACC} = \Big\{ (M, x) : \text{ Maszyna Turinga } M \text{ akceptuje dane wejciowe } x \Big\}$$

M_1

$\mathtt{q_0\ a \longrightarrow q_{ACC}\ a\ S}$

$\mathtt{q_0\ b \longrightarrow q_1\ \sqcup\ R}$

$\mathtt{q_1\ a \longrightarrow q_1\ \sqcup\ R}$

$\mathtt{q_1\ b \longrightarrow q_1\ \sqcup\ R}$

$\mathtt{q_1\ \sqcup \longrightarrow q_1\ \sqcup\ R}$

$(M_1,\ abaab)$ ✓

$(M_1,\ babba)$ ✗

Problem akceptowania wejścia

$$\mathcal{P}_{ACC} = \Big\{\big(M, x\big) : \text{ Maszyna Turinga } M \text{ akceptuje dane wejciowe } x\Big\}$$

M_1

$q_0\ a \longrightarrow q_{ACC}\ a\ S$
$q_0\ b \longrightarrow q_1\ \sqcup\ R$
$q_1\ a \longrightarrow q_1\ \sqcup\ R$
$q_1\ b \longrightarrow q_1\ \sqcup\ R$
$q_1\ \sqcup \longrightarrow q_1\ \sqcup\ R$

M_2

$q_0\ a \longrightarrow q_0\ a\ R$
$q_0\ b \longrightarrow q_1\ a\ S$
$q_0\ \sqcup \longrightarrow q_0\ \sqcup\ L$
$q_1\ a \longrightarrow q_0\ a\ L$
$q_1\ \sqcup \longrightarrow q_1\ a\ S$
$q_1\ b \longrightarrow q_{ACC}\ \sqcup\ S$

$\big(M_1,\ abaab\big)$ ✓

$\big(M_1,\ babba\big)$ ✗

Problem akceptowania wejścia

$$\mathcal{P}_{ACC} = \Big\{ (M, x) : \text{ Maszyna Turinga } M \text{ akceptuje dane wejciowe } x \Big\}$$

Maszyna rozpoznająca $\mathcal{P}_{ACC}$

- ☞ Uruchom maszynę M na x
- ☞ Jeśli M zatrzyma się w stanie akceptującym $\Rightarrow$ **akceptuj**
- ☞ Jeśli M zatrzyma się w stanie odrzucającym $\Rightarrow$ **odrzuć**

Założenie

1 $$\mathbb{M}_R\Big(\langle M\rangle, x\Big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } x \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } x \end{cases}$$

Problem akceptowania wejścia

Założenie

1. $$\mathbb{M}_R\Big(\langle M\rangle, x\Big) \;=\; \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } x \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } x \end{cases}$$

2. $$\mathbb{M}_A\Big(\langle M\rangle\Big) \;=\; \mathbb{M}_R\Big(\langle M\rangle, \langle M\rangle\Big)$$

Problem akceptowania wejścia

Założenie

1 $$\mathbb{M}_R\Big(\langle M\rangle, x\Big) \;=\; \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } x \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } x \end{cases}$$

2 $$\mathbb{M}_A\Big(\langle M\rangle\Big) \;=\; \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } \langle M\rangle \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } \langle M\rangle \end{cases}$$

Problem akceptowania wejścia

Założenie

1. $$\mathbb{M}_R\big(\langle M\rangle, x\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } x \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } x \end{cases}$$

2. $$\mathbb{M}_A\big(\langle M\rangle\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } \langle M\rangle \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } \langle M\rangle \end{cases}$$

3. $$\mathbb{M}_O\big(\langle M\rangle\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } \mathbb{M}_A \text{ nie akceptuje } M \\ \textbf{odrzucaj} & \text{jeśli } \mathbb{M}_A \text{ akceptuje } M \end{cases}$$

Problem akceptowania wejścia

Założenie

1. $$\mathbb{M}_R\big(\langle M\rangle, x\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } x \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } x \end{cases}$$

2. $$\mathbb{M}_A\big(\langle M\rangle\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } \langle M\rangle \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } \langle M\rangle \end{cases}$$

3. $$\mathbb{M}_O\big(\langle M\rangle\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ nie akceptuje } \langle M\rangle \\ \textbf{odrzucaj} & \text{jeśli } M \text{ akceptuje } \langle M\rangle \end{cases}$$

Problem akceptowania wejścia

Założenie

1. $$\mathbb{M}_R\big(\langle M\rangle, x\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } x \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } x \end{cases}$$

2. $$\mathbb{M}_A\big(\langle M\rangle\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } \langle M\rangle \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } \langle M\rangle \end{cases}$$

3. $$\mathbb{M}_O\big(\langle M\rangle\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ nie akceptuje } \langle M\rangle \\ \textbf{odrzucaj} & \text{jeśli } M \text{ akceptuje } \langle M\rangle \end{cases}$$

4. $$\mathbb{M}_O\big(\langle \mathbb{M}_O\rangle\big) = \, ?$$

Problem akceptowania wejścia

Założenie

1. $$\mathbb{M}_R\big(\langle M\rangle, x\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } x \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } x \end{cases}$$

2. $$\mathbb{M}_A\big(\langle M\rangle\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } \langle M\rangle \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } \langle M\rangle \end{cases}$$

3. $$\mathbb{M}_O\big(\langle M\rangle\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ nie akceptuje } \langle M\rangle \\ \textbf{odrzucaj} & \text{jeśli } M \text{ akceptuje } \langle M\rangle \end{cases}$$

4. $$\mathbb{M}_O\big(\langle \mathbb{M}_O\rangle\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } \mathbb{M}_O \text{ nie akceptuje } \langle \mathbb{M}_O\rangle \\ \textbf{odrzucaj} & \text{jeśli } \mathbb{M}_O \text{ akceptuje } \langle \mathbb{M}_O\rangle \end{cases}$$

Problem akceptowania wejścia

Założenie

1. $\mathbb{M}_R\big(\langle M\rangle, x\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } M \text{ akceptuje } x \\ \textbf{odrzucaj} & \text{jeśli } M \text{ nie akceptuje } x \end{cases}$

2. $\mathbb{M}_A\big(\langle M\rangle\big) =$
   - **akceptuj** jeśli M akceptuj…
   - **odrzucaj** jeśli … $\langle M\rangle$

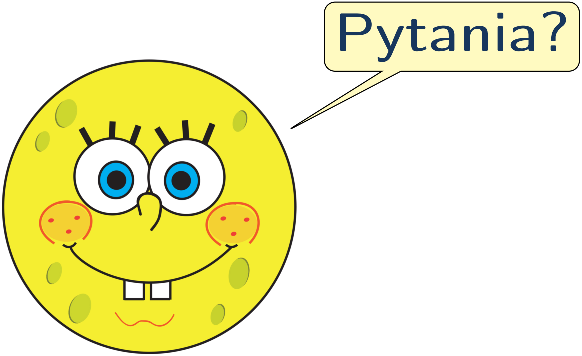
3. $\mathbb{M}_O\big(\langle M$ …
   - … jeśli M nie akceptuje $\langle M\rangle$
   - **odrzucaj** jeśli M akceptuje $\langle M\rangle$

! $\mathbb{M}_O$ **akceptuje** $\langle\mathbb{M}_O\rangle \iff \mathbb{M}_O$ **nie akceptuje** $\langle\mathbb{M}_O\rangle$

4. $\mathbb{M}_O\big(\langle\mathbb{M}_O\rangle\big) = \begin{cases} \textbf{akceptuj} & \text{jeśli } \mathbb{M}_O \text{ nie akceptuje } \langle\mathbb{M}_O\rangle \\ \textbf{odrzucaj} & \text{jeśli } \mathbb{M}_O \text{ akceptuje } \langle\mathbb{M}_O\rangle \end{cases}$

Twierdzenie

Problem akceptowania danych wejściowych przez maszynę Turinga jest nierozstrzygalny


Pytania?