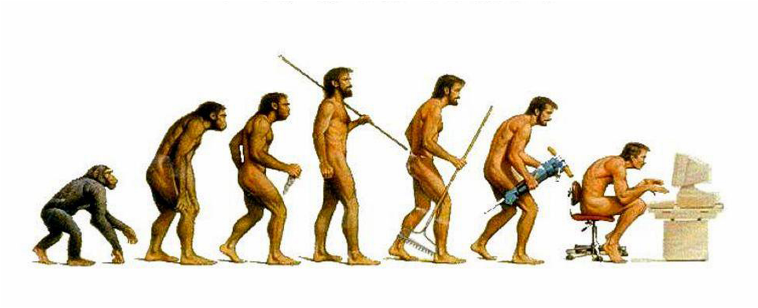


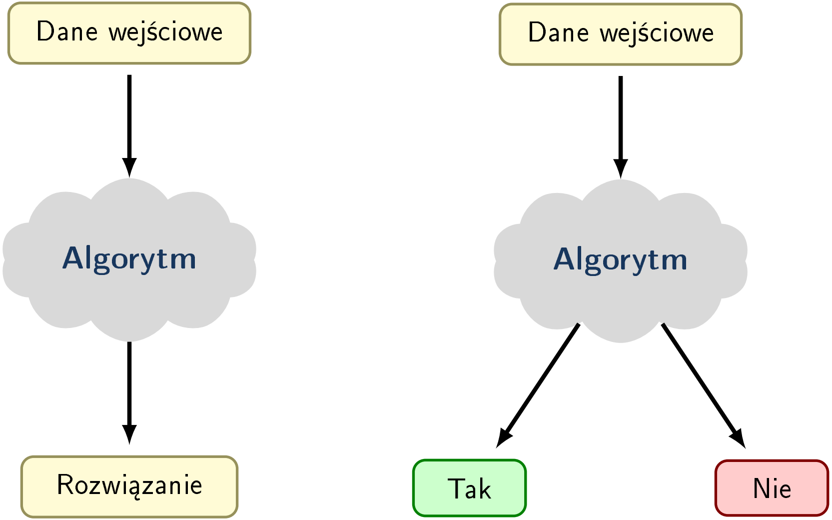
Teoria Obliczalności

Marcin Piątkowski

Wykład 7

Rozstrzygalność

Problemy obliczeniowe i decyzyjne



Problem vs język vs zbiór

Problem

$\mathcal{P}$ = „Liczba naturalna x jest parzysta"

Zbiór

$Z_{\mathcal{P}} = \{0, 2, 4, 6, 8, 10, 12, 14, 16, \ldots\}$

Język

$L_{\mathcal{P}} = \left\{w \in \{0,1\}^* : \ w \text{ jest binarnym zapisem liczby parzystej}\right\}$

Instancje pozytywne

$2 \in Z_{\mathcal{P}}$ $\quad 110100100 \in L_{\mathcal{P}}$

Instancje negatywne

$3 \notin Z_{\mathcal{P}}$ $\quad 101101 \notin L_{\mathcal{P}}$

Problem vs język vs zbiór

Problem

$\mathcal{P}$ = „Graf G jest spójny”

Zbiór

$Z_{\mathcal{P}} = \{G : \ G \text{ jest grafem spójnym}\}$

Język

$L_{\mathcal{P}} = \left\{w \in \Sigma^* : \ w \text{ jest reprezentacją/opisem grafu spójnego}\right\}$

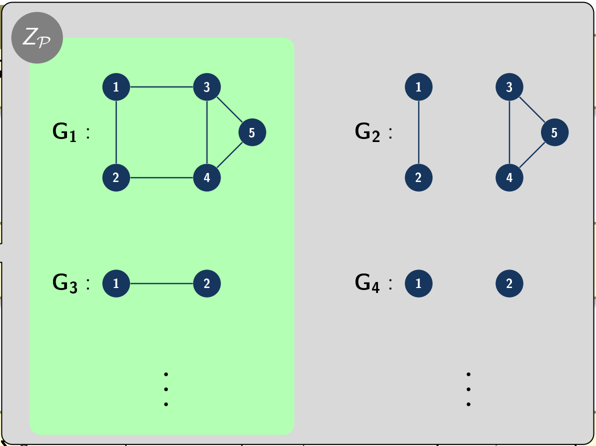
Problem vs język vs zbiór

Problem

$\mathcal{P}$ = „Graf G

Zbiór

$Z_{\mathcal{P}} = \{G$

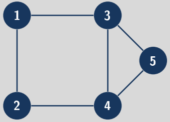


Język

$L_{\mathcal{P}} = \{w \in \Sigma^* : w \text{ jest reprezentacją/opisem grafu spójnego}\}$

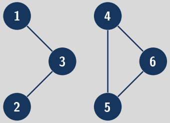
Problem vs język vs zbiór

▼ 5 $ 1 # 2 $ 1 # 3 $ 2 # 4 $ 3 # 4 $ 3 # 5 $ 4 # 5 $



$\Sigma = \{\blacktriangledown, \$, \#, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

▼ 6 $ 1 # 3 $ 2 # 3 $ 4 # 5 $ 4 # 6 $ 5 # 6 $



Język

$L_{\mathcal{P}} = \{w \in \Sigma^* : \; w \text{ jest reprezentacją/opisem grafu spójnego}\}$

Rozstrzygalność

Język $L \subseteq \Sigma^*$ nazywamy **rozstrzygalnym** jeśli istnieje maszyna Turinga, która zatrzymuje się dla każdego słowa $w \in \Sigma^*$ odpowiednio w stanie **akceptującym** jeśli $w \in L$ oraz w stanie **odrzucającym** jeśli $w \notin L$

Rozstrzygalność

Język $L \subseteq \Sigma^*$ nazywamy **rozstrzygalnym** jeśli istnieje maszyna Turinga, która zatrzymuje się dla każdego słowa $w \in \Sigma^*$ odpowiednio w stanie **akceptującym** jeśli $w \in L$ oraz w stanie **odrzucającym** jeśli $w \notin L$

Problem $\mathcal{P}_1$: „liczba $x \in \mathbb{N}$ jest parzysta” jest **rozstrzygalny**

Problem $\mathcal{P}_2$: „graf G jest spójny” jest **rozstrzygalny**

Rozstrzygalność

Problem „ϕ_x jest totalna" jest **nierozstrzygalny**

Problem „ϕ_x jest totalna" jest **nierozstrzygalny**

1. Przypuśćmy, że problem „ϕ_x jest totalna" jest rozstrzygalny

Rozstrzygalność

Problem „ϕ_x jest totalna" jest **nierozstrzygalny**

1. Przypuśćmy, że problem „ϕ_x jest totalna" jest rozstrzygalny

2. $$t(x,y) = \begin{cases} \phi_x(y) & \text{jeśli } \phi_x \text{ jest totalna} \\ 0 & \text{jeśli } \phi_x \text{ nie jest totalna} \end{cases} \Longrightarrow$$ totalna i obliczalna

Problem „ϕ_x jest totalna" jest **nierozstrzygalny**

1. Przypuśćmy, że problem „ϕ_x jest totalna" jest rozstrzygalny

2. $$t(x, y) = \begin{cases} \phi_x(y) & \text{jeśli } \phi_x \text{ jest totalna} \\ 0 & \text{jeśli } \phi_x \text{ nie jest totalna} \end{cases} \Longrightarrow$$ totalna i obliczalna

3. $g(x) = t(x, x) + 1 \quad \Longrightarrow \quad$ totalna i obliczalna

   $g = \phi_i \quad \Longrightarrow \quad g(y) = \phi_i(y) = t(i, y)$

Problem „ϕ_x jest totalna” jest **nierozstrzygalny**

1. Przypuśćmy, że problem „ϕ_x jest totalna” jest rozstrzygalny

2. $$t(x,y) \;=\; \begin{cases} \phi_x(y) & \text{jeśli } \phi_x \text{ jest totalna} \\ 0 & \text{jeśli } \phi_x \text{ nie jest totalna} \end{cases} \quad \Longrightarrow \quad \text{totalna i obliczalna}$$

3. $$g(x) = t(x,x) + 1 \quad \Longrightarrow \quad \text{totalna i obliczalna}$$
   $$g = \phi_i \quad \Longrightarrow \quad g(y) \;=\; \phi_i(y) \;=\; t(i,y)$$

4. $$t(i,i)+1 \overset{\text{definicja}}{=} g(i) \overset{\text{obliczalność}}{=} t(i,i)$$
   $$t(i,i)+1 \neq t(i,i)$$

Problem „ϕ_x jest totalna" jest **nierozstrzygalny**

1. ~~Przypuśćmy, że problem „ϕ_x jest totalna" jest rozstrzygalny~~

2. $$t(x,y) = \begin{cases} \phi_x(y) & \text{jeśli } \phi_x \text{ jest totalna} \\ 0 & \text{jeśli } \phi_x \text{ nie jest totalna} \end{cases} \Longrightarrow$$ totalna i obliczalna

3. $g(x) = t(x,x) + 1 \Longrightarrow$ totalna i obliczalna

   $g = \phi_i \Longrightarrow g(y) = \phi_i(y) = t(i,y)$

4. $$\begin{tikzcd} & g(i) \arrow[dl, equal, "\text{definicja}"'] \arrow[dr, equal, "\text{obliczalność}"] & \\ t(i,i)+1 \arrow[rr, phantom, "\neq"] & & t(i,i) \end{tikzcd}$$

Problem „$x \in D_x$" jest **nierozstrzygalny**

Problem „$x \in D_x$" jest **nierozstrzygalny**

1. Przypuśćmy, że problem „$x \in D_x$" jest rozstrzygalny

Rozstrzygalność

Problem „$x \in D_x$" jest **nierozstrzygalny**

1. Przypuśćmy, że problem „$x \in D_x$" jest rozstrzygalny

2. $g(x) = \begin{cases} 1 & x \notin D_x \\ \infty & x \in D_x \end{cases}$ jest obliczalna $\implies$ $g = \phi_m$

Rozstrzygalność

Problem „$x \in D_x$'' jest **nierozstrzygalny**

1. Przypuśćmy, że problem „$x \in D_x$'' jest rozstrzygalny

2. $$g(x) = \begin{cases} 1 & x \notin D_x \\ \infty & x \in D_x \end{cases} \quad \text{jest obliczalna} \quad \Longrightarrow \quad g = \phi_m$$

3. $$m \in D_m \iff m \in D_g \iff m \notin D_m$$

Problem „$x \in D_x$” jest **nierozstrzygalny**

1. ~~Przypuśćmy, że problem „$x \in D_x$” jest rozstrzygalny~~

2. $g(x) \;=\; \begin{cases} 1 & x \notin D_x \\ \infty & x \in D_x \end{cases}$ jest obliczalna $\implies$ $g = \phi_m$

3. $m \in D_m \iff m \in D_g \iff m \notin D_m$

Zbiory rekurencyjne

Zbiór $A \subseteq \boldsymbol{N}^n$ nazywamy **rekurencyjnym** jeśli jego funkcja charakterystyczna $c_A : \boldsymbol{N}^n \to \boldsymbol{N}$ określona następująco

$$c_A(\overline{x}) = \begin{cases} 1 & \overline{x} \in A \\ 0 & \overline{x} \notin A \end{cases}$$

jest obliczalna

✓ Zbiór liczb parzystych jest rekurencyjny

✗ Zbiór $\{x \in \boldsymbol{N} : \ x \in D_x\}$ nie jest rekurencyjny

Zbiory rekurencyjne

Zbiór $A \subseteq \boldsymbol{N}^n$ nazywamy **rekurencyjnym** jeśli jego funkcja charakterystyczna $c_A : \boldsymbol{N}^n \to \boldsymbol{N}$ określona następująco

$$c_A(\overline{x}) = \begin{cases} 1 & \overline{x} \in A \\ 0 & \overline{x} \notin A \end{cases}$$

jest obliczalna

✓ Zbiór liczb parzystych jest rekurencyjny

✗ Zbiór $\{x \in \boldsymbol{N} : \ x \in D_x\}$ nie jest rekurencyjny

✓ Zbiór nieskierowanych grafów spójnych jest rekurencyjny

Zbiory rekurencyjne

funkcja c_A
jest obliczalna

problem „$\bar{x} \in A$”
jest rozstrzygalny

zbiór A jest
rekurencyjny

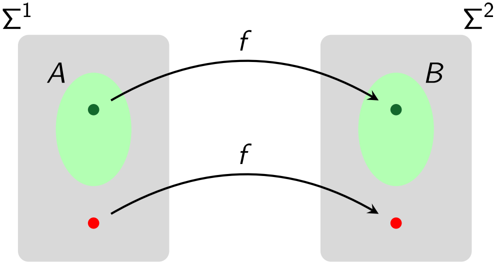
Własności zbiorów rekurencyjnych

Twierdzenie

Dla ustalonego $n \geq 1$ zbiór wszystkich podzbiorów rekurencyjnych $A \subseteq \boldsymbol{N}^n$ jest zamknięty na operacje teoriomnogościowe

- $\emptyset$ oraz $\boldsymbol{N}^n$ są rekurencyjne
- dopełnienie: $c_{\boldsymbol{N}^n \setminus A}(\overline{x}) \;=\; 1 - c_A(\overline{x})$
- suma: $c_{A \cup B}(\overline{x}) \;=\; \max\Big(c_A(\overline{x}), c_B(\overline{x})\Big)$
- przekrój: $c_{A \cap B}(\overline{x}) \;=\; c_A(\overline{x}) \cdot c_B(\overline{x})$

Redukcja problemu obliczeniowego

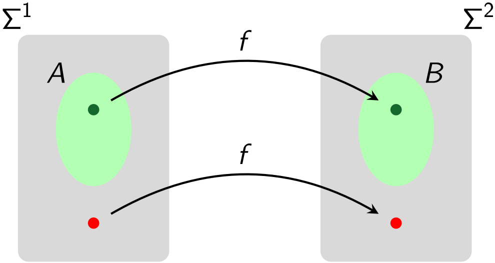


Redukcja A do B

Język (problem) $A \subseteq \Sigma_1^*$ jest redukowalny do języka (problemu) $B \subseteq \Sigma_2^*$ jeśli istnieje totalna i obliczalna funkcja $f : \Sigma_1^* \longrightarrow \Sigma_2^*$ taka, że

$$\forall_{x \in \Sigma_1^*}\ x \in A \iff f(x) \in B$$

Redukcja problemu obliczeniowego



Obserwacja

Jeśli problem A redukuje się do problemu B:

- ☞ Możliwe jest wykorzystanie rozwiązania problemu B w celu rozwiązania problemu A
- ☞ Rozwiązanie problemu A nie może być trudniejsze niż rozwiązanie problemu B

Wykorzystanie redukcji

Twierdzenie

Niech A i B będą językami, zaś funkcja f redukcją A do B. Jeśli język B jest **rozstrzygalny**, również język A jest **rozstrzygalny**.

Wykorzystanie redukcji

Twierdz

Istnieje rozstrzygająca go maszyna M_B

Niech A i B będą językami, zaś funkcja f redukcją A do B. Jeśli język B jest **rozstrzygalny**, również język A jest **rozstrzygalny**.

Konstrukcja maszyny M_A rozstrzygającej A

- ☞ Oblicz wartość $f(x)$
- ☞ Uruchom maszynę M_B na wejściu $f(x)$
- ☞ Jako wynik zwróć wynik działania maszyny M_B

Twierdzenie

Niech A i B będą językami, zaś funkcja f redukcją A do B. Jeśli język B jest **rozstrzygalny**, również język A jest **rozstrzygalny**.

Konstrukcja maszyny M_A rozstrzygającej A

- ☞ Oblicz wartość $f(x)$
- ☞ Uruchom maszynę M_B na wejściu $f(x)$
- ☞ Jako wynik zwróć wynik działania maszyny M_B

Wniosek

Jeśli **nierozstrzygalny** problem A jest redukowalny do problemu B, to problem B również jest **nierozstrzygalny**

Twierdzenie

Niech A i B będą językami, zaś funkcja f redukcją A do B. Jeśli język B jest **rozstrzygalny**, również język A jest **rozstrzyg**…

Konstrukcja maszyny M_A roz…

- ☞ Oblicz wartoś…
- ☞ U…
- …ania maszyny M_B

! Istnienie redukcji problemu A do problemu B nic nie mówi na temat rozstrzygalności żadnego z nich. Daje ona jedynie możliwość rozwiązania problemu A korzystając z rozwiązania problemu B oraz uzależnia rozstrzygalność A od rozstrzygalności B.

…sek

Jeśli **nierozstrzygalny** problem A jest redukowalny do problemu B, to problem B również jest **nierozstrzygalny**

Problem stopu

$$\mathcal{P}_{STOP} = \Big\{ \big(M, x\big) : \text{ maszyna } \ M \text{ zatrzymuje się na danych } x \Big\}$$

Problem stopu

$$\mathcal{P}_{STOP} = \Big\{(M, x) :\ \text{maszyna}\ \ M \text{ zatrzymuje się na danych } x\Big\}$$

Twierdzenie

Problem stopu P_{STOP} jest nierozstrzygalny

Problem stopu

$$\mathcal{P}_{STOP} = \Big\{ (M, x) : \text{ maszyna } \ M \text{ zatrzymuje się na danych } x \Big\}$$

Twierdzenie

Problem stopu P_{STOP} jest nierozstrzygalny

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

$$\mathcal{P}_{ACC} \longrightarrow \mathcal{P}_{STOP}$$

Problem stopu

Istnieje rozstrzygająca go maszyna M_S

1. Przypuśćmy, że $\mathcal{P}_{STOP}$ jest rozstrzygalny

Problem stopu

1. Przypuśćmy, że $\mathcal{P}_{STOP}$ jest rozstrzygalny

2. **Maszyna M_A**

   - ☞ Uruchom maszynę M_S na danych wejściowych (M, x)
   - ☞ Jeśli maszyna M_S odrzuci $(M, x) \implies$ **odrzuć**
   - ☞ Jeśli maszyna M_S zaakceptuje $(M, x) \implies$ symuluj działanie maszyny M na x
   - ☞ Zwróć wynik (**akceptacja**/**odrzucenie**) zwrócony przez maszynę M

1

Przypuśćmy, że $\mathcal{P}_{STOP}$ jest rozstrzygalny

2

Maszyna M_A

- ☞ Uruchom maszynę M_S na danych wejściowych (M, x)
- ☞ Jeśli maszyna M_S odrzuci $(M, x) \implies$ **odrzuć**
- ☞ Jeśli maszyna M_S zaakceptuje $(M, x) \implies$ symuluj działanie maszyny M na x
- ☞ Zwróć wynik (**akceptacja**/**odrzucenie**) zwrócony przez maszynę M

3

Jeśli maszyna M_S rozstrzygałaby problem $\mathcal{P}_{STOP}$ maszyna M_A rozstrzygałaby problem $\mathcal{P}_{ACC}$. Wiemy jednak, że problem $\mathcal{P}_{ACC}$ jest **nierozstrzygalny**. Zatem również $\mathcal{P}_{STOP}$ **nie może** być rozstrzygalny.

Problem stopu

1

~~Przypuśćmy, że $\mathcal{P}_{STOP}$ jest rozstrzygalny~~

2

Maszyna M_A

- ☞ Uruchom maszynę M_S na danych wejściowych (M, x)
- ☞ Jeśli maszyna M_S odrzuci $(M, x) \Longrightarrow$ **odrzuć**
- ☞ Jeśli maszyna M_S zaakceptuje $(M, x) \Longrightarrow$ symuluj działanie maszyny M na x
- ☞ Zwróć wynik (**akceptacja**/**odrzucenie**) zwrócony przez maszynę M

3

Jeśli maszyna M_S rozstrzygałaby problem $\mathcal{P}_{STOP}$ maszyna M_A rozstrzygałaby problem $\mathcal{P}_{ACC}$. Wiemy jednak, że problem $\mathcal{P}_{ACC}$ jest **nierozstrzygalny**. Zatem również $\mathcal{P}_{STOP}$ **nie może** być rozstrzygalny.

Twierdzenie Rice'a

Twierdzenie Rice'a – wersja I

Niech $\mathcal{B}$ będzie **właściwym** i **niepustym** podzbiorem zbioru wszystkich funkcji obliczalnych. Wówczas problem „$\phi_x \in \mathcal{B}$" jest **nierozstrzygalny**. Równoważnie, zbiór $B = \{x \in \mathbf{N} : \ \phi_x \in \mathcal{B}\}$ **nie jest rekurencyjny**.

Twierdzenie Rice'a – wersja II

Niech $\mathcal{B}$ będzie **właściwym** i **niepustym** podzbiorem zbioru wszystkich języków rozpoznawalnych przez maszyny Turinga. Wówczas problem „$L(M) \in \mathcal{B}$" jest **nierozstrzygalny** (zbiór $B = \{M : \ L(M) \in \mathcal{B}\}$ **nie jest rekurencyjny**).

Twierdzenie Rice'a – wersja III

Zbiór $B = \{x \in \mathbf{N} : \ \phi_x \in \mathcal{B}\}$ jest **rekurencyjny** (problem „$\phi_x \in \mathcal{B}$" jest **rozstrzygalny**) wtedy i tylko wtedy, gdy $B = \emptyset$ lub $B = \mathbf{N}$.

Twierdzenie Rice'a – dowód

0. $\mathcal{B}$ – **niepusty** i **właściwy** podzbiór zbioru wszystkich języków rozpoznawalnych przez maszyny Turinga

Twierdzenie Rice'a – dowód

0 $\mathcal{B}$ – **niepusty** i **właściwy** podzbiór zbioru wszystkich języków rozpoznawalnych przez maszyny Turinga

1 Bez straty ogólności możemy założyć, że język pusty $\emptyset \notin \mathcal{B}$

W przeciwnym przypadku rozważamy dopełnienie $\mathcal{B}$

Twierdzenie Rice'a – dowód

0. $\mathcal{B}$ – **niepusty** i **właściwy** podzbiór zbioru wszystkich języków rozpoznawalnych przez maszyny Turinga
1. Bez straty ogólności możemy założyć, że język pusty $\emptyset \notin \mathcal{B}$
2. $\mathcal{B} \neq \emptyset$, zatem istnieje język $L \in \mathcal{B}$ rozpoznawany przez maszynę M_L

Twierdzenie Rice'a – dowód

0. $\mathcal{B}$ – **niepusty** i **właściwy** podzbiór zbioru wszystkich języków rozpoznawalnych przez maszyny Turinga

1. Bez straty ogólności możemy założyć, że język pusty $\emptyset \notin \mathcal{B}$

2. $\mathcal{B} \neq \emptyset$, zatem istnieje język $L \in \mathcal{B}$ rozpoznawany przez maszynę M_L

3. Dla pary (M, x) tworzymy maszynę M_x działającą na wejściu y według schematu:
   - ☞ Symuluj działanie maszyny M na wejściu x
   - ☞ Jeśli M odrzuci $x \Longrightarrow$ odrzuć
   - ☞ Jeśli M zaakceptuje x, symuluj działanie M_L na wejściu y
   - ☞ Jeśli maszyna M_L zatrzyma się, M_x zwróć wynik jej działania

Twierdzenie Rice'a – dowód

0. $\mathcal{B}$ – **niepusty** i **właściwy** podzbiór zbioru wszystkich języków rozpoznawalnych przez maszyny Turinga

1. Bez straty ogólności możemy założyć, że język pusty $\emptyset \notin \mathcal{B}$

2. $\mathcal{B} \neq \emptyset$, zatem istnieje język $L \in \mathcal{B}$ rozpoznawany przez maszynę M_L

3. Dla pary (M, x) tworzymy maszynę M_x działającą na wejściu y według schematu:
   - Symuluj działanie maszyny M na wejściu x
   - Jeśli M odrzuci $x \Longrightarrow$ odrzuć
   - Jeśli M zaakceptuje x, symuluj działanie M_L na wejściu y
   - Jeśli maszyna M_L zatrzyma się, M_x zwróć wynik jej działania

4. **(a)** M akceptuje $x \Longrightarrow M_x$ akceptuje y lub nie zatrzymuje się $\Longrightarrow L(M_x) = L \in \mathcal{B}$

   (b) M nie akceptuje $x \Longrightarrow M_x$ nie akceptuje $y \Longrightarrow L(M_x) = \emptyset \notin \mathcal{B}$

Odrzuca lub nie zatrzymuje się

Twierdzenie Rice'a – dowód

0 $\mathcal{B}$ – **niepusty** i **właściwy** podzbiór zbioru wszystkich języków rozpoznawalnych przez maszyny Turinga

1 Bez straty ogólności możemy założyć, że język pusty $\emptyset \notin \mathcal{B}$

2 $\mathcal{B} \neq \emptyset$, zatem istnieje język $L \in \mathcal{B}$ rozpoznawany przez maszynę M_L

3 Dla pary (M, x) tworzymy maszynę M_x działającą na wejściu y według schematu:

- Symuluj działanie maszyny M na wejściu x
- Jeśli M odrzuci $x \Longrightarrow$ odrzuć
- Jeśli M zaakceptuje x, symuluj działanie M_L na wejściu y
- Jeśli maszyna M_L zatrzyma się, M_x zwróć wynik jej działania

4
(a) M akceptuje $x \Longrightarrow M_x$ akceptuje y lub nie zatrzymuje się $\Longrightarrow L(M_x) = L \in \mathcal{B}$

(b) M nie akceptuje $x \Longrightarrow M_x$ nie akceptuje $y \Longrightarrow L(M_x) = \emptyset \notin \mathcal{B}$

5 Zatem $L(M_x) \in \mathcal{B} \iff M_A$ akceptuje x

$\mathcal{P}_{ACC}$ jest **nierozstrzygalny** $\Longrightarrow$ „$L(M_x) \in \mathcal{B}$" jest **nierozstrzygalny**

Przykład I

Problem wejścia: „$y \in D_x$" jest nierozstrzygalny

1. Rozważamy zbiór $\mathcal{B} = \left\{ f \in C_1 : \ y \in D_f \right\}$
2. $g(x) = 17 \ \in \ \mathcal{B} \quad \Longrightarrow \quad \mathcal{B}$ jest niepusty
3. $f_\emptyset \ \notin \ \mathcal{B} \quad \Longrightarrow \quad \mathcal{B}$ jest właściwym podzbiorem zbioru wszystkich funkcji obliczalnych

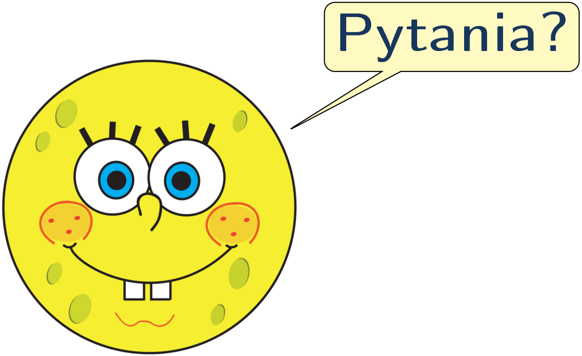
Zbiór $\mathcal{B}$ jest **właściwym** i **niepustym** podzbiorem zbioru wszystkich funkcji obliczalnych. Zatem na mocy twierdzenia Rice'a **nie jest rekurencyjny**. Równoważnie, problem „$y \in D_x$" jest **nierozstrzygalny**.

Przykład II

Problem wyjścia: „$y \in \mathrm{Im}_x$" jest nierozstrzygalny

1. Rozważamy zbiór $\mathcal{B} = \left\{ f \in C_1 : \; y \in f(D_f) \right\}$
2. $g(x) = x \; \in \; \mathcal{B} \implies \mathcal{B}$ jest niepusty
3. $f_\emptyset \; \notin \; \mathcal{B} \implies \mathcal{B}$ jest właściwym podzbiorem zbioru wszystkich funkcji obliczalnych

Zbiór $\mathcal{B}$ jest **właściwym** i **niepustym** podzbiorem zbioru wszystkich funkcji obliczalnych. Zatem na mocy twierdzenia Rice'a **nie jest rekurencyjny**. Równoważnie, problem „$y \in D_x$" jest **nierozstrzygalny**.


Pytania?