Diagonalisierung

gegeben Menge $M = \{m_0, m_1, m_2, m_3, \dots\}$,
deren Elemente durch Werte von Attributen $a_0, a_1, a_2, a_3, \dots$

	a_0	a_1	a_2	a_3	$\dots$
m_0	$m_{0,0}$	$m_{0,1}$	$m_{0,2}$	$m_{0,3}$	
m_1	$m_{1,0}$	$m_{1,1}$	$m_{1,2}$	$m_{1,3}$	
m_2	$m_{2,0}$	$m_{2,1}$	$m_{2,2}$	$m_{2,3}$	
m_3	$m_{3,0}$	$m_{3,1}$	$m_{3,2}$	$m_{3,3}$	
$\vdots$					$\ddots$

neues Element:

	a_0	a_1	a_2	a_3	$\dots$
m	$\overline{m_{0,0}}$	$\overline{m_{1,1}}$	$\overline{m_{2,2}}$	$\overline{m_{3,3}}$	

wobei $\overline{m_{i,i}}$ irgendein Wert ungleich $m_{i,i}$ ist

m unterscheidet sich von jedem $m_k \in M$ durch Wert des Attributs a_k

Ziel: Element konstruieren, das nicht zu M gehört

gegeben durch Werte der Attribute

Zeile von Attributwerten, die sich von jeder Zeile in der Tabelle an mindestens einer Stelle unterscheidet

Beispiel (Georg Cantor):

Satz: $\mathbb{R} \cap [0;1]$ ist überabzählbar (d.h. nicht abzählbar).

Beweis: Angenommen, $\mathbb{R} \cap [0;1] = \{m_0, m_1, m_2, m_3, \dots\}$

$= \{m(i) \mid i \in \mathbb{N}\}$, wobei

$m: \mathbb{N} \to \mathbb{R}$

„Es gibt mehr reelle Zahlen als natürliche Zahlen."

$m_0 = 0, m_{0,0}\ m_{0,1}\ m_{0,2}\ m_{0,3} \dots$ ← Dezimalentwicklung der Zahl m_0 (unendlich) ($m_{i,j}$ sind Ziffern)

$m_1 = 0, m_{1,0}\ m_{1,1}\ m_{1,2}\ m_{1,3} \dots$

$m_2 = 0, m_{2,0}\ m_{2,1}\ m_{2,2}\ m_{2,3} \dots$

$m_3 = 0, m_{3,0}\ m_{3,1}\ m_{3,2}\ m_{3,3} \dots$

$\frac{1}{4} = m_4 = 0,2\ 5\ 0\ 0 \dots$

$\frac{1}{3} = m_5 = 0,3\ 3\ 3\ 3 \dots$

$1 = m_6 = 0,9\ 9\ 9\ 9$

Definiere

$m = 0, d_0 d_1 d_2 d_3 \dots,$

wobei $d_k = \begin{cases} 1, & \text{falls } m_{k,k} = 0 \\ 0, & \text{sonst} \end{cases}$

(d.h. $d_k \neq m_{k,k}$)

Beh.: Für alle k ist $m \neq m_k$.

A $m = m_k$, dann $d_k \neq m_{k,k}$ ↯

$m \in \mathbb{R} \cap [0;1] \setminus \{m_0, m_1, \dots\}$

Sei $\varphi_n : \mathbb{N} \to \mathbb{N}$ die einstellige Funktion über $\mathbb{N}$, die von der Turingmaschine M_n berechnet wird.

TM mit Gödelisierung n (bzw. $bin(n)$)

Satz: Es gibt nicht berechenbare Funktion $\varphi : \mathbb{N} \to \mathbb{N}$.

Beweis:

Def.-bereich $\mathbb{N}$

	0	1	2	3	…
φ_0	$\varphi_0(0)$	$\varphi_0(1)$	$\varphi_0(2)$	$\varphi_0(3)$	
φ_1	$\varphi_1(0)$	$\varphi_1(1)$	$\varphi_1(2)$	$\varphi_1(3)$	
φ_2	$\varphi_2(0)$	$\varphi_2(1)$	$\varphi_2(2)$	$\varphi_2(3)$	
φ_3	$\varphi_3(0)$	$\varphi_3(1)$	$\varphi_3(2)$	$\varphi_3(3)$	
⋮					⋱

alle ber. Funktionen $\mathbb{N} \to \mathbb{N}$

$\varphi_i(j)$ ist Wert der Fkt. φ_i an der Stelle j,

$\varphi_i(j) = \uparrow$, falls $\varphi_i(j)$ undef.

Definiere φ wie folgt:

$$\varphi(k) = \begin{cases} 1, & \text{falls } \varphi_k(k) \text{ def. ist und } \varphi_k(k) = 0, \\ 0, & \text{sonst} \end{cases}$$

(also $\varphi(k) \neq \varphi_k(k)$)

Also: Es gibt kein $k \in \mathbb{N}$ mit $\varphi = \varphi_k$ ∎

Korollar: K ist nicht entscheidbar.

Beweis: Falls K entscheidbar wäre, dann wäre φ berechenbar. ∎

Satz: Es gibt Sprachen, die nicht rekursiv aufzählbar sind.

Beweis: Sei M_n das Semi-Entscheidungsverfahren, das von TM mit Gödelisierung n berechnet wird.

alle Eingaben

	0	1	2	3	...
M_0	$\underline{\downarrow}$	$\uparrow$	$\uparrow$	$\downarrow$	
M_1	$\uparrow$	$\underline{\uparrow}$	$\downarrow$	$\uparrow$	
M_2	$\downarrow$	$\downarrow$	$\underline{\downarrow}$	$\downarrow$	
M_3	$\uparrow$	$\uparrow$	$\uparrow$	$\underline{\downarrow}$	
$\vdots$					

alle Semi-Entscheidungsalgorithmen

$\uparrow \;\hat{=}\; M_i(j)$ hält nicht

$\downarrow \;\hat{=}\; M_i(j)$ hält

Definiere Sprache L_d wie folgt:

$k \in L_d \Leftrightarrow$ TM M_d bei Eingabe d (bzw. des d-ten Wortes) hält nicht.

A $L_d = L(M_k)$ für ein $k \in \mathbb{N}$.

Dann $k \in L_d \Leftrightarrow k \notin L(M_k)$

↯ $\blacksquare$

Korollar: $\overline{K}$ ist nicht r.a. und K ist nicht entscheidbar.

Bew.: $L_d = \overline{K}$ $\boxtimes$

Satz: Es gibt eine totale WHILE-berechenbare Funktion, die nicht LOOP-berechenbar ist.

aus $\mathbb{N} \to \mathbb{N}$

Behaupt: Ω WHILE-bs., nicht LOOP-br.

Erwähnt: Ackermann-Fkt., total, WHILE-bs., aber nicht LOOP-bs. (ohne Beweis)

zweistellige

Beweis: Seien $P_0, P_1, P_2, P_3 \dots$ alle LOOP-Programme in quasilexikogr. Ordnung.

Sei $f_k : \mathbb{N} \to \mathbb{N}$ die einstellige LOOP-Fkt., die von P_k berechnet wird.

Eingabe-Werte

	0	1	2	3	...
f_0	$f_0(0)$	$f_0(1)$	$f_0(2)$	$f_0(3)$	...
f_1	$f_1(0)$	$f_1(1)$	$f_1(2)$	$f_1(3)$	...
f_2	$f_2(0)$	$f_2(1)$	$f_2(2)$	$f_2(3)$	...
f_3	$f_3(0)$	$f_3(1)$	$f_3(2)$	$f_3(3)$	...
⋮					

alle einstell. LOOP-br. Fkt.

Def. $f : \mathbb{N} \to \mathbb{N}$ wie folgt:

$$f(k) = \begin{cases} 0, & \text{falls } f_k(k) \neq 0 \\ 1, & \text{sonst} \end{cases}$$

also $f(k) \neq f_k(k)$

Also: $f \neq f_k$ für alle $k \in \mathbb{N}$, also ist f nicht LOOP-ber., aber f ist WHILE-bs. ■