Satz von Rice

Def. siehe 27 Sätze.

Der Satz besagt, dass es mit einem Algo. nicht möglich ist zu bestimmen, ob eine TM ein best. Verhalten aufweist oder nicht.

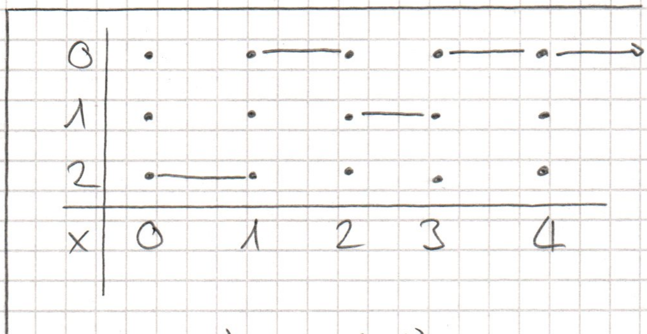
Bsp: Es gibt kein Algo. der entscheidet, ob eine TM auf einen bestimmten Output hält oder nicht.

Rekursion

add: $\mathbb{N}^2 \longrightarrow \mathbb{N}$, $(x,y) \mapsto (x+y)$ Subt: $\mathbb{N}^2 \longrightarrow \mathbb{N}$, $(x,y) \mapsto (x-y)$

Vorgehen: – Für versch. Möglichkeiten f berechnen

zB $f(0) = 2$
$f(1) = 0$
$f(2) = 1$
$f(3) = 0$
$f(n) = 0$



$\Rightarrow f(x) = add(add(g(x), sg(x)), sg(x))$
$g(x) = subt(sg(P(P(x))), sg(P(x)))$

Wie ist $L_1 \leq_m L_2$ definiert?

$w \in L_1 \iff f(w) \in L_2$
resp. $w \notin L_1 \iff f(w) \notin L_2$

Alphabet $\Sigma = \{0,1\}$ und Sprachen $L_1, L_2 \subset \Sigma^*$ wobei gilt
$L_1 = \{0, 1, 00, 11, 000, 111\}$
$L_2 = \{0^k 1^k \mid k \in \mathbb{N}^*\}$

Gilt $L_1 \leq_m L_2$ oder $L_2 \leq_m L_1$ oder beides?

$f: w \longrightarrow \begin{cases} 01, & \text{falls } w \in L_1 \\ 00, & \text{sonst} \end{cases}$

$f: w \longrightarrow \begin{cases} 00, & \text{falls } w \in L_2 \\ 01, & \text{sonst} \end{cases}$

Welche 4 Parameter def. eine KK?

- Berechnungsmodell
- Berechnungsmodus
- betrachtete Ressource
- Eine Schranke für die betrachtete Ressource

Erklären sie möglichst genau was die KK P ist und geben sie ein Element als Bsp. an.

Zur Klasse P gehören Sprachen, die in polynomialer Laufzeit gelöst werden können, z.B add.

KK NP

Zur Klasse NP gehören Sprachen, bei denen in polynomialer Zeit entschieden werden kann ob die gegebene Lösung stimmt. Es ist aber nicht klar, ob das allgemeine Poblem auch in polynomialer Zeit lösbar ist, z.B. subset sum Problem.

DTIME

Gilt $L_1 \in DTIME_\Sigma(n)$?

→ Ja, kann in linearer Zeit mit einer TM gelöst werden

Gilt $L_2 \in DTIME_\Sigma(n^2)$?

→ Ja, kann in quad. Zeit...

Gilt $L_3 \in DTIM_\Sigma(\log(n))$?

→ Ja, $L \in DTIME(1)$

$L_1 = \{ 01, 0101, 010101, \ldots \}$

$L_2 = \{ \varepsilon, 01, 0011, 000111, \ldots \}$

$L_3 = \{ \varepsilon, 01, 0011, \ldots \}$

Turing-Tabelle von M

111 | 010101010 | 11 | 0100010 1000 1000 | 11 | 01000100 1000 100 | 111

δ	Q_1	T_1	Q_2	T_2	⟷
$0^{0},1\,0^{1},1\,0^{0},1\,0^{1},1\,0^{1}$	q_0	0	q_0	0	L
$0^{0},100^{2},10^{1},1000^{3},1000^{3}$	q_0	1	q_0	b	R
$0^{0},1000^{3},100^{1},1000^{3},100^{2}$	q_0	b	q_1	b	H
i+1 j k+1 l m					

Q_1 : 1 × 0 → q_0
2 × 0 → q_1

T_1 : 1 × 0 → 0
2 × 0 → 1
3 × 0 → b

⟷ : 1 × 0 → L
2 × 0 → H
3 × 0 → R

⇒ Bed. für Codierung beachten!
zB. i+1 ⇒ 1 abziehen!

Wie ist L_{univ} definiert?

In L_{univ} sind alle Wörter der Form <M>ω enthalten, bei denen M den Input ω akzeptiert. <M> ist die Gödelnummer der TM M.

Univ. TM mit Input (siehe Tabelle oben)

… 111 | 1101 = ω

$q_0 1101 \vdash b\, q_0 101 \vdash bb\, q_0 01 \vdash b\, q_0\, b01 \vdash b\, q_1\, b01$

↳ $q_1 \in F$

Ist das obige Inputwort Element von L_{univ}?

Das Wort ist in L_{univ} weil $\omega \in L(M)$ bzw $M(\omega) = 1$

Welcher Klasse gehören folgende Sprachen an?

09.01.17

(4)

$L_1 := \{011000, 0011000000, 000111000000000\}$

$L_2 := \{0^{2k}1110^k \mid k \in \mathbb{N}\}$

$L_3 := \{0^k1110000^l \mid k, l \in \mathbb{N}\}$

L_4 besteht aus allen Wörtern, die gleich oft die Buchstabenfolge 01 und 10 enthalten.

L_1: regulär L_2: kontextfrei L_3: regulär L_4: regulär

Forts.: Rekursion

Ist die folgende Sprache rekursiv?

$L(S) := \{<M> \mid M \in TM_\Sigma$ und M berechnet die Funktion $f\}$

$\Rightarrow$ Nein

Präzise Antwort

- 3 Punkte von Satz von Rice auflisten

$\Rightarrow$ $L(S)$ ist nicht rekursiv!

Primitiv rekursiv: Grundfunktionen

1. Z die 0-stellige Null-Funktion. Das entspricht der Konstanten $0 \in \mathbb{N}$
2. S die 1-stellige Nachfolgerfunktion
$$S: \mathbb{N} \to \mathbb{N}$$
$$x \to x+1$$
3. P die 1-stellige Vorgängerfunktion
$$P: \mathbb{N} \to \mathbb{N}$$
$$x \to \begin{cases} x-1 & \text{falls } x > 0 \text{ ist,} \\ 0, & \text{sonst} \end{cases}$$
4. U_j^n di n-stellige Projektionsfunktionen
$$U_j^n(x_1 \ldots x_n) = x_j$$

(3-stellig: $sub(x, y+1) = h(x, y, sub(x,y))$

	Input		Output			
δ	Q	Γ	Q	Γ	⟷	
Start	q_0	b	q_S	0	R	
	q_0	0	$q_{Lö}$	b	R	
	q_0	1	q_R	1	R	
	q_S	b	q_H	0	R	→ 1. Fall (00)
	q_H	–	–	–	–	
Input löschen	$q_{Lö}$	1	$q_{Lö}$	b	R	
	$q_{Lö}$	0	$q_{Lö}$	b	R	→ 3. Fall
	$q_{Lö}$	b	q_H	b	R	
nach rechts gehen	q_R	0	q_R	0	R	
	q_R	1	q_R	1	R	
	q_R	b	q_{Div}	b	L	
x/2	q_{Div}	1	q_{Add}	b	L	
	q_{Div}	0	q_{Add}	b	L	
	(q_{Div}	b	q_{Add}	b	L)	
+1	q_{Add}	0	q_H	1	R	
	q_{Add}	1	q_{Add}	0	L	
	q_{Add}	b	q_H	1	R