

CoMa Übung

(Satz von Rice, Nichtentscheidbarkeit)

nächste Woche: Infos zu Klausur

Satz von Rice: \mathcal{R} = Menge aller berechenbaren Funktionen

berechenbare Funktion: $f: X \rightarrow Y$ heißt berechenbar
 g.d.w. \exists TM M die für
 $x \in X$ in akzept. Zustand stehen bleibt
 und $f(x)$ auf das Band schreibt
 für $x \notin X$ hält M in nicht akz. Zustand
 oder hält gar nicht

Satz von Rice:

Die Sprache $L(S) := \{ \langle M \rangle \mid M \text{ berechnet Funktion } f \in S \}$
 ist nicht entscheidbar für $\emptyset \neq S \subsetneq \mathcal{R}$

anschaulich: - def. bestimmte Teilmenge von Funktionen

Bsp: Polynome, kürzeste Wege Algorithmen

\Rightarrow nicht autom. feststellbar, ob gegeb. Algor.
 (Rice) in diese Teilmenge fällt

entscheidbar: Sprache L ist entscheidbar g.d.w.

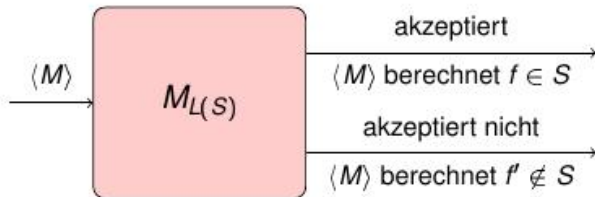
\exists TM M , die w akzeptiert s.d.w.
 $w \in L$ ist

Beweis:

- $S \neq \emptyset / \mathbb{R} \Rightarrow \exists f \in S, f' \in \mathbb{R}$
- sei $u: \emptyset \rightarrow \mathbb{N}$, die überall undef. Funktion
- $u \in S$ oder $u \notin S$, o.B.d.t.: $u \in S, f \in \mathbb{R} \setminus S$
- f ist berechenbar $\Rightarrow M_f$ ist TM die f berechnet

Idee der Konstruktion:

- Annahme: es existiert TM $M_{L(S)}$ die $L(S)$ entscheidet
- \rightarrow def. TM $M_{\overline{H}_\varepsilon}$ für das Komplement des speziellen Halteproblems
- \rightarrow wäre Widerspruch, da H_ε und \overline{H}_ε nicht rekursiv



Versuche:

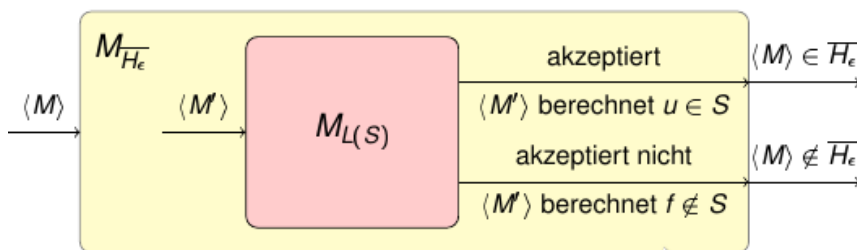
- $\langle M \rangle \in \overline{H}_\varepsilon \cong M$ hält nicht auf ε

$\rightarrow \langle M' \rangle$ soll Funktion u berechnen

- $\langle M \rangle \notin \overline{H}_\varepsilon \cong M$ hält auf ε

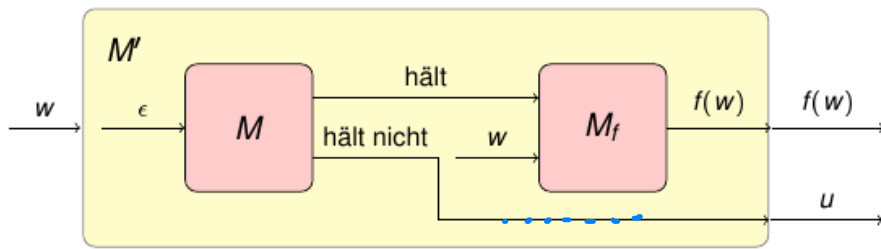
$\rightarrow \langle M' \rangle$ soll Funktion f berechnen

Zusammen:



Halteproblem durch TM für $L(S)$ gelöst

Konstruktion von $\langle M' \rangle$ aus $\langle M \rangle$:



↑ Interpretation von u , keine wirkliche Ausgabe

in Konstruktion wird Göddnummer $\langle M' \rangle$ aus $\langle M \rangle$ konstr. aber die Maschine nicht ausgeführt/simuliert

Insgesamt: $\langle M' \rangle$ ist "never-halting function" mit:

- $\langle M \rangle \in \overline{H}_\epsilon \cong M$ hält nicht auf $\epsilon \Rightarrow M'$ verhält sich wie Funktion u , bzw. berechnet u
- $\langle M \rangle \notin \overline{H}_\epsilon \cong M$ hält auf $\epsilon \Rightarrow M'$ verhält sich wie Funktion f , bzw. berechnet $f(w)$

da $u \in S$ und $f \notin S \Rightarrow$ können entscheiden ob $M \in \overline{H}_\epsilon$ \nexists

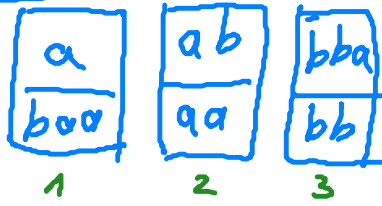
- Konsequenzen:
- keine Antivirensoftware möglich, die schädliches Verhalten von bel. Software erkennt \square
 - keine "Testsoftware" kann für andere Software berechnen, ob diese für jede Eingabe korrekt funktioniert

Halteproblem oft für Reduktionen verwendet
 \rightarrow zeigt Nichtentscheidbarkeit

Relativ abstraktes Problem, jetzt ein einfacheres:

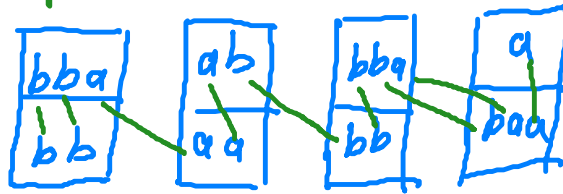
Postisches Korrespondenz Problem:

Eingabe: Dominosteine
(bel. viele Steine pro Typ)

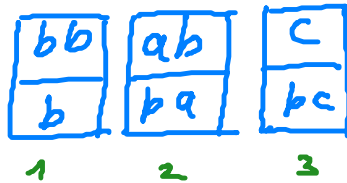


gesucht: endl. Steinfolge, s.d. Steine nebeneinander oben und unten gleiches Wort besitzen

Lösung im Beispiel: 3, 2, 3, 1



weiteres Beispiel:



Lösung:
 • 1, 2, 3
 • 1, 2, 2, ..., 2, 3
 • 1, 2, 3, 1, 2, 3, ..., 1, 2, 3

Postische Korrespondenz Problem ist nicht entscheidbar.

Beweisskizze: - simuliere TM mit Steinen
 - speichere „Geschichte“ des Bands der TM

typischer Zustand: 10100101 q_z 00101

Historie/Geschichte: $q_0 101 \# 1 q_1 01 \# 11 q_2 1 \# 1 q_3 0 \dots$

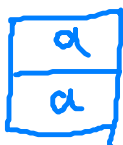
startzustand, $\delta(q_0, 1) = (q_1, R)$
 $w = 101$

diese Historie soll oben und unten auf Dominosteinen stehen

Arten von Dominosteinen:



startstein, $w = \text{Bandeingabe}$, $q_0 = \text{startzustand}$



Kopierstein, gleiches Zeichen oben und unten für alle $a \in \Sigma$

- | |
|-----------|
| $q_i a$ |
| $b q_j D$ |

 Übergangsstein: $\delta(q_i, a) = (q_j, b, D)$
 bsp:

$q_7 0$
$1 q_4$

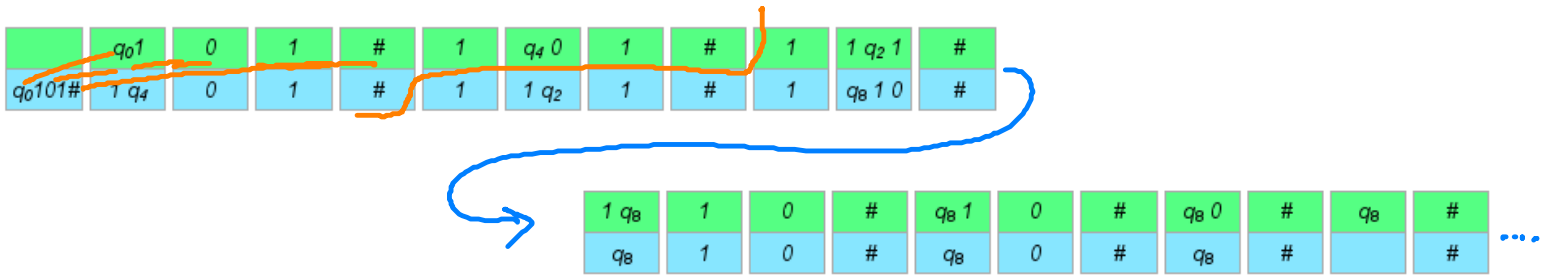
 für $\delta(q_7, 0) = (q_4, 1, R)$

$q_7 0$
$q_4 0$

 $\delta(q_7, 0) = (q_4, 0, N)$



Beispielablauf: $q_0 1 0 1 \# 1 q_4 0 1 \# 1 q_2 1 \# 1 q_8 1 0 \#$



• zusätzlich am Ende die zeilen die zuviel sind entfernen mit extra steinen

Kleensche Rekursion Theorem:

- $f(x, y)$ ist berechenbare Funktion *
- $\Rightarrow \exists$ berech. Funktion $n(y)$, s.d. $\langle n(y) \rangle = \langle f(n(y), y) \rangle \forall y$
- \rightarrow für alle Turing vollständigen Programmiersprachen ex. ein Programm, das seinen Quellcode ausgibt (z. B. in Java)

Aufgabe: Schreibe Java Programm, das seinen Quellcode ausgibt.
 \rightarrow Code aus Datei lesen nicht erlaubt

Problem: triviales kopieren in `sys.out` ... funktioniert nicht. „Ausserum“ fehlt.

* Version in Übung war fehlerhaft