19.6.13

CoMa Übung

(Satz von Rice, Nichtentscheidbarkeit)

nächste Woche: Infos zu Klausur

Satz von Rice:

$\mathcal{R}$ = Menge aller berechenbaren Funktionen

berechenbare Funktion: $f: X \to Y$ heißt berechenbar g.dw. $\exists$ TM M die für $x \in X$ im 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
$\underset{\text{(Rice)}}{\Rightarrow}$ nicht autom. feststellbar, ob gegeb. Algor. in diese Teilmenge fällt

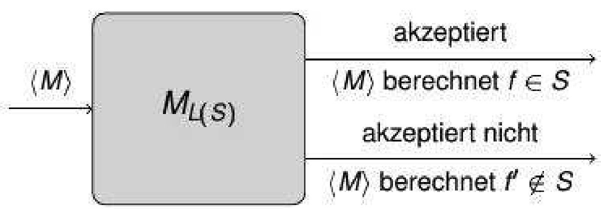
<u>entscheidbar:</u> Sprache L ist entscheidbar g.d.w. $\exists$ TM M, die w akzeptiert g.d.w $w \in L$ ist

Beweis: - $S \neq \emptyset / R \Rightarrow \exists f \in S, f' \in R$

- sei $u: \emptyset \to \mathbb{N}$, die überall undef. Funktion
- $u \in S$ oder $u \notin S$, o.B.d.A: $u \in S$, $f \in R \setminus S$
- 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 $\overline{H_\varepsilon}$ nicht rekursiv



Versuche:

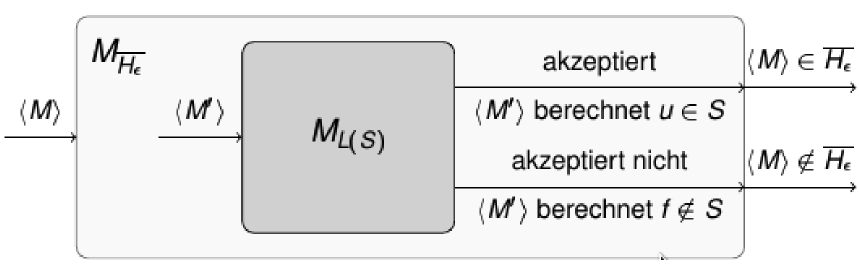
- $\langle M \rangle \in \overline{H_\varepsilon} \cong M$ hält nicht auf ε

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

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

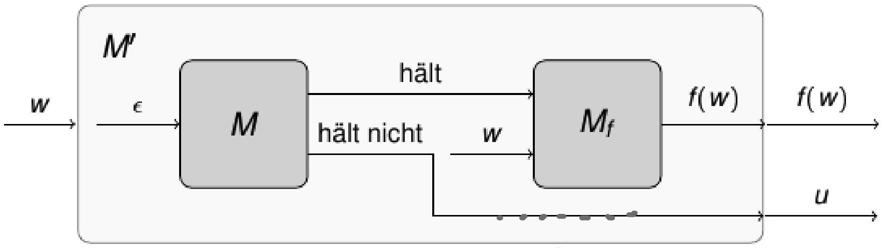
$\rightarrow \langle M' \rangle$ soll Funktion f berechnen

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 Gödelnummer $\langle M' \rangle$ aus $\langle M \rangle$ konstr.
aber die Maschine <u>nicht</u> ausgeführt/simuliert

Insgesamt: $\langle M' \rangle$ ist neue „Funktion" mit:

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

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

□

<u>Konsequenzen:</u>

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

Halteproblem oft für Reduktionen verwendet
→ zeigt Nichtentscheidbarkeit
Relativ abstraktes Problem, jetzt ein einfacheres:

Postsches Korrespondenz Problem:

Eingabe: Dominosteine (bel. viele Steine pro Typ)

1	2	3
a	ab	bba
baa	aa	bb

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

Lösung im Beispiel: 3, 2, 3, 1

3	2	3	1
bba	ab	bba	a
bb	aa	bb	baa

weiteres Beispiel:

1	2	3
bb	ab	c
b	ba	bc

Lösung:
- 1, 2, 3
- 1, 2, 2, ..., 2, 3
- 1, 2, 3, 1, 2, 3 ... 1, 2, 3

Postsche Korrespondenz Problem ist nicht entscheidbar.

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

typischer Zustand: $10100101\,q_z\,00101$

Historie/Geschichte: $\underbrace{q_0 101}_{\substack{\text{startzustand,}\\ w=101}} \# 1 q_4 01 \# 11 q_z 1 \# 1 q_8 10 \ldots\ldots$

$\delta(q_0, 1) = (q_4, 1, R)$

diese Historie soll oben und unten auf Dominosteinen stehen

Arten von Dominosteinen:

- Startstein, w = Bandeingabe, q_0 = Startzustand

$q_0 w \#$

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

a
a

$q_i\ a$
$b\ q_j\ D$

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

bsp:

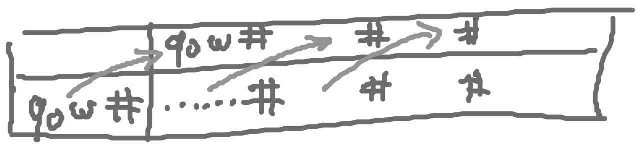
$q_7\ 0$
$1\ q_4$

für $\delta(q_7, 0) = (q_4, 1, R)$

$q_7\ 0$
$q_4\ 0$

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

Steinfolge der Art:



Beispielablauf: $q_0 101 \# 1 q_4 01 \# 11 q_2 1 \# 1 q_8 10 \#$

	$q_0 1$	0	1	#	1	$q_4\ 0$	1	#	1	$1\ q_2\ 1$	#
$q_0 101\#$	$1\ q_4$	0	1	#	1	$1\ q_2$	1	#	1	$q_8\ 1\ 0$	#

$1\ q_8$	1	0	#	$q_8\ 1$	0	#	$q_8\ 0$	#	q_8	#	....
q_8	1	0	#	q_8	0	#	q_8	#		#	

- zusätzlich am Ende die Zeichen 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 Turingvollständigen 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. „Aussenrum" fehlt.

* Version in Übung war fehlerhaft