Tutorium 09-06-10

Notiztitel 5/6/2009

nicht entscheidbar

Ü 6.7: Für alle M gibt es x mit
$M(x) = U(x)$

(6.5 c): Kein Abschluss unter komplement mit TM beweise?

Gegeben M gesucht x mit $M(x) = U(x)$

$M = M_i$ $M(x) = U(\langle i, x \rangle)$

Tipp: wähle M' mit $M'(y) = M\langle y, y \rangle$

leicht zu konstruieren

Standard tupel funktion

Weil M' eine TM ist gibt es ein j
mit $M' = M_j$.

d.h.

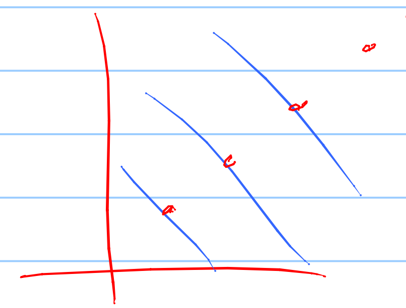
$U\langle j, y \rangle = M_j(y) = M'(y) = M\langle y, y \rangle$ für alle y

wähle $x = \langle j, j \rangle$

dann

$$U(x) = U\langle j, j\rangle = M_j(j) = M'(j)$$

$$= M\langle j, j\rangle = M(x)$$



6.5 c Zeige mit TM daß komplement eine semi-entscheidbare Sprache nicht unbedingt semi-entscheidbar ist

L semi-entscheidbar? ← partiell-charakt. Funktion ist b.b.

mit TM: es gibt eine TM M, die L akzeptiert aber nicht anhält für $w \notin L$

wir wissen Selbstanwendbarkeitsproblem
$S = \{ i \mid i \in \text{domain } \varphi_i \}$
semi-entscheidbar, aber $\bar{S}$ nicht

Argument neu mit TM

wähle $S_T = \{ \text{code}(M) \mid M \text{ hält bei Eingabe } \text{code}(M) \}$

S_T ist semi-entscheidbar

Konstruktion von U

Beweis: bei Eingabe $w \in \Sigma^*$

- teste ob $w = \text{code}(M)$ für ein M, wenn nein verwerfe (Endlosschleife)
- sonst simuliere M auf Eingabe w

$\overline{S_T}$ ist nicht semi-entscheidbar

Annahme $\overline{S_T}$ ist semi-entscheidbar und Maschine M akzeptiert $\overline{S_T}$

also

$M(w)$ hält $\Leftrightarrow w \in \overline{S_T}$

(und akzeptiert)

Was macht M bei Eingabe $w = \text{code}(M)$?

M hält auf w $\iff$ $w \in \overline{S_T}$

$\iff$ $w \notin S_T$

$\iff$ M hält nicht auf $\text{code}(M) = w$

↯

Weil das ein Widerspruch ist, kann M so nicht existieren, also $\overline{S_T}$ nicht semi-entscheidbar

---

6.6: TM M listet eine Sprache L auf, wenn M der Reihe nach alle Elemente generiert (und ggf. nie aufhört)

$\equiv$ aufzählbar

a) L ist semi-entscheidbar, wenn auflistbar

b) L ist entscheidbar, wenn auflistbar in 'kanonischer' Reihenfolge

↳ "Standard"

L kann nicht sagen, wenn aktuell generiertes Wort 'größer' ist als Testwort $w \in L$?
und w in Auflistung nicht vorkam

Annahme M listet L in kanonischer Reihenfolge auf. Wir konstruieren M', die L entscheidet wie folgt.

Bei Eingabe von w simuliere M bis entweder w generiert ist, oder das erste Wort generiert wird, das in der Reihenfolge nach w erscheint.
Im ersten Fall akzeptiere, andernfalls verwerfe.

Da M die Worte in kanonischer Reihenfolge aufzählt akzeptiert M'' ein Wort w g.d.w
$w \in L$.
Also L entscheidbar

Gegenrichtung: L entscheidbar

Konstruiere TM wie folgt
Generiere Worte aus Σ^* in kanonischer Reihenfolge und teste jeweils $w \in L$
wenn ja, schreibe w aufs Band
sonst generiere nächstes Wort ohne zu schreiben

Mathematik

Funktion total = immer definiert (terminiert immer)

Menge entscheidbar = es gibt eine totale 0-1-wertige berechenbare Funktion f mit $w \in L \iff f(x) = 1$

Informatik (Logik)

der Test ist total

semi-entscheidbar = der Test hält nicht für $w \notin L$

Der besseren Lesbarkeit wegen eine vorgeschriebene Lösung

6.5

a) Vereinigung semi-entscheidbarer Sprachen

Seien M_1, M_2 Maschinen, die L_1 bzw L_2 akzeptieren
(und für Elemente außerhalb der Sprachen nicht anhalten)

Konstruiere M für $L_1 \cup L_2$ wie folgt

- Schreibe die Eingabe w auf zwei Arbeitsbänder
- Simuliere M_1 auf Band 1, M_2 auf Band 2
  jeweils einen Schritt im Wechsel
- wenn $w \in L_1 \cup L_2$, dann muß eine der beiden
  Simulationen nach einer gewissen Zahl von Schritten
  anhalten und w akzeptieren.
- Ansonsten terminiert M nicht

b) analog

c) Komplement ist i.a. <u>nicht</u> semi-entscheidbar
Gegenbeispiel war das Selbstanwendbarkeitsproblem
$S = \{ i \mid \varphi_i(i) \text{ terminiert} \} = \{ i \mid i \in domain(\varphi_i) \}$

Beweis mit TM geht ähnlich.

Betrachte $S_T = \{ \text{code}(M) \mid \underbrace{M \text{ hält bei Eingabe code}(M)}_{\text{Selbstreferenz}} \}$

S_T ist semi-entscheidbar (simuliere M und übernehme Resultat)

Annahme $\overline{S_T}$ ist semi-entscheidbar.

Dann gibt es eine TM $\bar{M}$ mit

$$w \in \overline{S_T} \iff \bar{M} \text{ akzeptiert } w \text{ und terminiert}$$

Betrachte das Verhalten von $\bar{M}$ auf $w = \text{code}(\bar{M})$

$\bar{M}$ terminiert auf $w \iff \bar{M}$ akzeptiert w
$\iff w \in \overline{S_T}$

$\iff w \notin S_T$

$\iff M$ hält <u>nicht</u> auf w

Das ist ein Widerspruch, also kann M nicht existieren und $\overline{S_T}$ ist <u>nicht</u> semi-entscheidbar

d) folgt aus c)

<u>6.7</u> Sieht komplizierte aus, als es ist – deswegen der Hinweis

U sei die UTM, d.h. $U\langle i,n\rangle = M_i(n)$ ⇐

achtung – das ist kurznotation für "U verhält sich bei Eingabe von $\langle i,n\rangle$ wie M_i bei Eingabe n"

<u>Zeige</u>: für jede TM M gibt es ein x mit $U(x) = M(x)$

also genauer, ein $\langle i,j\rangle$ mit $M_i(j) = U\langle i,j\rangle = M\langle i,j\rangle$

<u>Idee</u> betrachte M' mit $M'(y) = M(\langle y,y\rangle)$

da M' offensichtlich als TM konstruierbar ist, gibt es eine Gödelnummer j für M' (d.h. $M' = M_j$)

also

$M\langle y,y\rangle = M'(y) = M_j(y) = U\langle j,y\rangle$ für <u>alle</u> y

also auch für $y = j$. Wähle $x = \langle j,j\rangle$