Satz: Für jede konsistente, vollständige Henkin-Theorie Φ gilt:
$\mathfrak{I}_\Phi \models \Phi$; insbes: jede kons., vollst. Henkin-Th. hat Modell!

Bemerkung: Aus Def. $\mathfrak{I}_\Phi$ folgt für bel. kons. Theorie Φ
für bel. atomare Formeln φ: $\Phi \vdash \varphi \Leftrightarrow \mathfrak{I}_\Phi \models \varphi$

Vollständigkeit von Φ liefert die Analogie von $\Phi \vdash \varphi \vee \Phi \vdash \neg\varphi$
zu $\mathfrak{I}_\Phi \models \varphi \vee \mathfrak{I}_\Phi \models \neg\varphi$ und erlaubt es, die Äquivalenz $\Phi \vdash \varphi \Leftrightarrow \mathfrak{I}_\Phi \models \varphi$
fortzusetzen auf Formeln der Gestalt $\varphi \vee \psi$, $\neg\varphi$, $\varphi \wedge \psi$, ...
Schließlich Fortsetzbarkeit d. Äquivalenz auf Formeln der
Gestalt $\exists x \varphi$ durch Henkin-Eigenschaft.

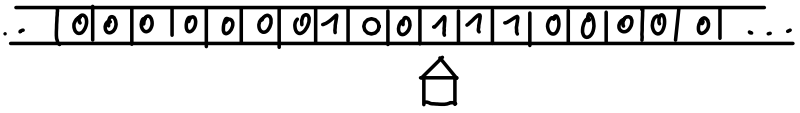
Nachdem Satz v. Henkin bewiesen ist, folgt V2 durch Einbettung
einer bel. konsistenten Theorie in eine konsistente, vollständige
Henkin-Theorie.

2009-01-27

Fr. 30.1. 14-16 Uhr i11 VO

Berechenbarkeit

Turing-Maschine



„Band" mit abzählbar unendlich vielen Feldern in denen jeweils
0 oder 1 steht.
Maschine steht mit Lese-/Schreibkopf auf einem Feld, und es gilt
gerade eine von endlich vielen Regeln $r_1, \ldots, r_n$ der Maschine, Regel i,
wenn die Maschine auf einem Feld mit 1 steht, besagt entweder: statt
1 schreiben: 0 oder 1, und dann nach Rechts oder nach Links gehen
und zur Regel j übergehen ($1 \leq j \leq n$), oder Regel ist undefiniert, wenn 1 auf

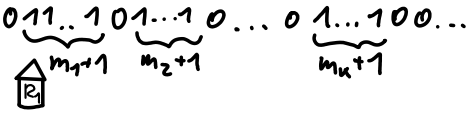
Feld steht, analog für 0.
Zu Beginn der Berechnung steht auf dem Band der „Input", es gilt Regel 1, dann läuft Maschine bis sie zu einer undefinierten Regel kommt, dann hält sie.

Definition: Turing-Maschine (mit n Regeln) def. als Funktion
$f : \mathrm{dom}(f) \subseteq \{1,\dots,n\} \times \{0,1\} \to \{1,\dots,n\} \times \{0,1\} \times \{R,L\}$

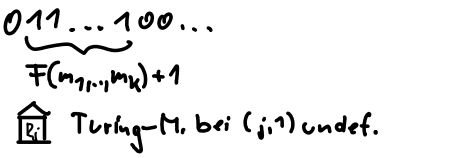
Definition: Eine Funktion $F: \mathrm{dom}(F) \subseteq \mathbb{N}_0 \to \mathbb{N}_0$ heißt partiell rekursiv, wenn eine Turing-Maschine existiert, die F berechnet, dh. eine Turing-Maschine, die, wenn sie auf einem Band, das einem Block von n+1 konsekutiven „1" enthält, sonst „0", auf dem Feld mit dem „1" ganz links unter Gültigkeit der Regel 1 gestartet wird
1) genau dann niemals hält, wenn $n \notin \mathrm{dom}\, F$
2) wenn $n \in \mathrm{dom}\, F$, hält mit Output $F(n)$
dh. auf dem Band steht ein Block von $F(n)+1$ Konsekutiven „1", sonst „0", Maschine steht auf „1" ganz links, während eine Regel, die bei „1" nicht def. ist gilt.

Notation: $\mathrm{dom}\, F$ ist der Definitionsbereich „domain" der Funktion F.

Analog $F: \mathrm{dom}\, F \subseteq \mathbb{N}_0^k \to \mathbb{N}_0$ partiell rekursiv, wenn Turing-Maschine existiert, die für alle $(m_1,\dots,m_k) \in \mathrm{dom}\, F$ mit Input



liefert Output



und für alle $(m_1,\dots,m_k) \in \mathbb{N}_0^k \setminus \mathrm{dom}\, F$ mit Input $(m_1,\dots,m_k)$ wie oben nicht hält.

Für $(m_1,\dots,m_k) \in \operatorname{dom} F$ schreibt $F(m_1,\dots,m_k)\downarrow$, für $(m_1,\dots,m_k) \notin \operatorname{dom} F$ schreibt $F(m_1,\dots,m_k)\uparrow$

Versuch einer Addition-Turing-Maschine:

Mit Input ...0 $\underbrace{11\dots1}_{n+1}$ 0 $\underbrace{11\dots10}_{m+1}$

wollen Output ..0 $\underbrace{11\dots10}_{n+m+1}$...

Regel Inhalt		Regel Inhalt Richtung
$(1,1)$	$\longmapsto$	$(1,1,R)$
$(1,0)$	$\longmapsto$	$(2,1,R)$
$(2,1)$	$\longmapsto$	$(2,1,R)$
$(2,0)$	$\longmapsto$	$(3,0,L)$
$(3,1)$	$\longmapsto$	$(4,0,L)$
$(4,1)$	$\longmapsto$	$(5,0,L)$
$(5,1)$	$\longmapsto$	$(5,1,L)$
$(5,0)$	$\longmapsto$	$(6,0,R)$

$f: \operatorname{dom} f \subseteq \{1,..6\} \times \{0,1\} \to \{1,..,6\} \times \{0,1\} \times \{R,L\}$

$\operatorname{dom} f = \{(1,1),(1,0),(2,1),(2,0),(2,1),(3,1),(4,1),(5,1),(5,0)\}$

Definition: Eine partiell rekursive Fkt $F: \operatorname{dom} F \subseteq \mathbb{N}_0^k \to \mathbb{N}$ heißt total rekursiv, wenn $\operatorname{dom} F = \mathbb{N}_0^k$

Beispiel einer totalen $(F: \mathbb{N}_0^k \to \mathbb{N}_0)$ Funktion, die nicht rekursiv ist.

Sei für $n,k \in \mathbb{N}_0$ $b(n,k) = c \in \mathbb{N}_0$ maximal, sodass c als Wert $F(k)$ einer durch eine Turing-Maschine mit $\leq n$ Regeln berechneten Funktion auftritt.

Da es nur endlich viele Turing-Maschinen mit $\leq n$ Regeln gibt, können diese auch nur endl. viele versch. Outputs bei Input k liefern.

Sei $G(n) := b(n,n)+1$, dann ist, wenn F durch eine Turing-Maschine mit m Regeln berechnet wird, $F(m) \leq b(m,m) < G(m)$, also $G \neq F$
Daher G keine rekursive Funktion.

Ü: Ebenso „busy beaver" Funktion nicht rekursiv
$b(m)$ = max Länge eines Blocks von „1" (sonst „0") der als Output einer Turing-Maschine mit $\leq m$ Regeln, die schließlich hält, gestartet auf leerem (nur „0") Band, auftritt.

Sei A bel. abzählbar unendl. Alphabet, und A^* die Menge aller endlichen Wörter mit Buchst. $\in A$.
Kann $\varphi: A^* \to \mathbb{N}_0$ injektiv A^* in $\mathbb{N}_0$ abbilden (Wörter $\in A^*$ eind. durch nat. Zahlen codieren). Kann durch Turing-Maschinen auch Funktionen $F: \operatorname{dom} F \subseteq (A^*)^k \to A^*$ berechnen, indem man die Funktion F^* mit $F^*(\varphi(w_1),\dots,\varphi(w_k)) := \varphi(F(w_1,\dots,w_k))$ betrachtet.

z.B. könnte man, eindeutige Primfaktorzerlegung in $\mathbb{Z}$ ausnutzend, $\varphi: A^* \to \mathbb{N}_0$ mit $A = \{a_0, a_1, \dots\}$ definieren als $\varphi(a_{n_1}, a_{n_2}, \dots, a_{n_k})$
$= 2^{n_1} \cdot 3^{n_2} \cdot \dots \cdot p_k^{n_k} = p_1^{n_1} \cdots p_k^{n_k}$ wobei p_i die i-te Primzahl

So kann man Turing-Maschinen als Wörter über dem Alphabet $\mathbb{N}_0 \cup \{(,), R, L, \mapsto\}$ auffassen und als natürliche Zahlen codieren

Halteproblem für Turing-Maschinen: betrachte die Funktion
$H: \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ sodass

$$H(n,k) = \begin{cases} 1 & \text{wenn Turing-M. Nr. } n \text{ hält schließlich bei Input } k \\ 0 & \text{wenn Turing-M. Nr. } n \text{ hält schließlich bei Input } k \text{ nicht.} \end{cases}$$

$$H_0(n) = \begin{cases} 1 & \text{wenn Turing-M. Nr. } n \text{ hält schließlich bei Input leeres Band} \\ 0 & \text{wenn Turing-M. Nr. } n \text{ hält schließlich bei Input leeres Band nicht} \end{cases}$$

Nicht rekursiv, zuerst zeigen, die Funktion $G: \mathbb{N}_0 \to \mathbb{N}_0$ geg. durch $G(n)$ ist max. Anzahl von Schritten, die eine Turing-Maschine mit

$\leq n$ Regeln, die schließlich hält, bei Input leeres Band zurücklegen kann, ist nicht rekursiv (sonst wäre „Busy-Beaver" rekursiv)

Wenn H rekursiv wäre, dann auch G_i; man müsste nur diejenigen Maschinen die stehenbleiben werden, ablaufen lassen und Ergebnisse vergleichen.