Satz Es gibt eine universelle Turingmaschine.

Bew.: Entwerfe U als 3-Band TM:

1. Schritt: Überprüfe syntaktische Korrektheit der Eingabe entsprechend der Konvention für $\langle M \rangle$

2. Schritt Kopiere $\langle M \rangle$ auf Band 2 und lösche diesen Teil von Band 1, so dass nur w dort stehen bleibt. Schreibe Zustand 0^i (am Anfang $i=1$) auf Band 3.

3. Schritt: Schritt-für-Schritt Simulation von M:

U liest von Band 1 Buchstabe X_j

U liest von Band 3 aktuellen Zustand 0^i

U sucht auf Band 2 den Block 110^i10^j1

und führt die Fortsetzung $0^k10^l10^m$ aus

neuer Zustand — überschreibe X_j durch X_l — Kopfbewegung D_m auf Band 1.

$\boxtimes$

§2 Unentscheidbarkeit

Kanonische Ordnung auf Σ^*:

- fixiere (sog. lexikographische) Ordnung auf Σ, z.B., falls $\Sigma = \{0,1\}$: $0 < 1$.
- erweitere lexikographische Ordnung auf Σ^n:

$$(x_1, \dots, x_n) < (y_1, \dots, y_n) \iff \exists i \in \{1, \dots, n\} : x_1 = y_1, \dots, x_{i-1} = y_{i-1} \text{ und } x_i < y_i.$$

- erweitere auf Σ^*:

$x < y \iff x$ kürzer als y, oder falls x, y gleich lang: $(x_1, \dots, x_n) < (y_1, \dots, y_n)$.

<u>Def.</u>: Sei M_i die TM, deren Gödelnummer in der kanonischen Ordnung aller Gödelnummern an i-ter Stelle steht. Sei $w_i \in \{0,1\}^*$ das Wort, das in $\{0,1\}^*$ an i-ter Stelle steht. Definiere die <u>Diagonalsprache</u> D als

$$D = \{ w_i : M_i \text{ akzeptiert } w_i \text{ nicht} \} \subseteq \{0,1\}^*$$

Satz Die Diagonalsprache D ist nicht rekursiv.

Bew.: Ang. D ist rekursiv. Dann gibt es eine TM M_j, die auf allen Eingaben hält und genau die Wörter aus D akzeptiert. Wende M_j auf w_j an.

1. Fall: $w_j \in D$

Nach Annahme akzeptiert M_j das Wort w_j.
Das kann nach Definition von D aber nicht sein.
Widerspruch.

2. Fall: $w_j \notin D$

Nach Annahme akzeptiert M_j das Wort w_j nicht.
Das kann nach Def. von D aber nicht sein.
Widerspruch. $\boxtimes$

Bem.:
- vgl. diesen Beweis zum Cantorschen Beweis der Überabzählbarkeit von $\mathbb{R}$ mit Hilfe des Diagonalarguments.
- D scheint sehr künstlich / konstruiert / an den Haaren herbeigezogen zu sein. Es ist aber der Ausgangspunkt für weitere Unentscheidbarkeitsbeweise.

Def.: Seien $L_1, L_2 \subseteq \Sigma^*$ Sprachen.
L_1 heißt auf L_2 <u>reduzierbar</u> (Notation $L_1 \leq L_2$), falls es eine TM gibt, die die Fkt. $f: \Sigma^* \to \Sigma^*$ berechnet, so dass für alle $w \in \Sigma^*$ gilt

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

Lemma Falls $L_1 \leq L_2$ und L_2 rekursiv, dann ist auch L_1 rekursiv.

Bew.: Sei M_2 die zu L_2 gehörige TM.
Definiere eine TM M_1 wie folgt: Bei Eingabe von w wendet M_1 die Fkt. f an. Nun simuliert M_1 die Berechnung von M_2 bei Eingabe $f(w)$. M_1 akzeptiert genau dann, wenn M_2 Eingabe $f(w)$ akzeptiert. ☒

Bem.:
- Aus dem Lemma folgt, dass wenn $L_1 \leq L_2$ und L_1 nicht rekursiv ist, auch L_2 nicht rekursiv ist.
- $L_1 \leq L_2 \mathrel{\hat{=}}$ „L_1 ist nicht schwieriger als L_2".

Lemma Sei $L \subseteq \Sigma^*$ eine Sprache und sei $\overline{L} = \{w \in \Sigma^* : w \notin L\}$ die zu L komplementäre Sprache. Falls L rekursiv ist, so ist auch $\overline{L}$ rekursiv und umgekehrt.

Bew.: Sei M eine TM, die auf allen Eingaben w hält und w akzeptiert gdw. $w \in L$. Konstruiere TM M' für $\overline{L}$, wie folgt: Für Eingabe $w \in \Sigma^*$ simuliere M. Falls M akzeptiert, akzeptiert M' nicht und umgekehrt. $\boxtimes$

Def.: Das Halteproblem ist die folgende Sprache

$$H = \{\langle M \rangle w : M \text{ hält bei Eingabe } w\}.$$

Satz Das Halteproblem ist nicht rekursiv.

Bew.: Ang. H ist rekursiv und M ist eine TM, die auf allen Eingaben w hält und w genau dann akzeptiert, wenn $w \in H$. Sei i so, dass w das i-te Wort aus $\{0,1\}^*$ ist. Betrachte $\langle M_i \rangle$ die i-te Gödelnummer. Wende M auf $\langle M_i \rangle w$ an.

Ist $\langle M_i \rangle w \in H$, dann hält M_i auf w. Wende universelle TM an, um M_i auf w zu simulieren. M_i akzeptiert w genau dann, wenn $w \notin D$, d.h. $w \in \overline{D}$.

Ist $\langle M_i \rangle w \notin H$, dann hält M_i auf w nicht. D.h. $w \in D$, also $w \notin \overline{D}$.
Somit ist $\overline{D}$ rekursiv. ↯ ☒

<u>Konsequenz:</u> Es gibt kein Computerprogramm, das die Korrektheit von allgemeinen Programmen überprüfen kann.

<u>Noch extremer:</u> Man kann zeigen (Satz von Rice), dass man keine nichttriviale Eigenschaft von allgemeinen Programmen berechnen kann.