universität innsbruck

Diskrete Mathematik

Ralph Bottesch
Burak Ekici
Johannes Koch
Georg Moser
David Obwaller
Vincent van Oostrom
Oleksandra Panasiuk

cbr.uibk.ac.at

Zusammenfassung der letzten LVA

Satz (Kontraposition des Schleifenlemmas)

Sei L eine formale Sprache über Σ, sodass:

- *für alle $n \in \mathbb{N}$ existiert ein Wort $w \in L$ mit $\ell(w) \geqslant n$, sodass*
- *für alle $x, y, z \in \Sigma^*$ mit $w = xyz$, $y \neq \epsilon$ und $\ell(xy) \leqslant n$ existiert $k \in \mathbb{N}$ mit $x(y)^k z \notin L$*

Dann ist L nicht regulär. ■

Beispiel

Sei $\Sigma = \{1\}$; dann ist

$$D = \{w \in \Sigma^* \mid \ell(w) \text{ ist eine Primzahl}\}$$

nicht regulär

Inhalte der Lehrveranstaltung (cont’d)

Reguläre Sprachen

deterministische Automaten, nichtdeterministische Automaten, endliche Automaten mit Epsilon-Übergängen, reguläre Ausdrücke, Abgeschossenheit, Schleifenlemma

Berechenbarkeitstheorie

deterministische TM, nichtdeterministische TM, universelle TMs, Äquivalenzen

Komplexitätstheorie

Grundlagen, die Klassen P und NP, polynomielle Reduktionen, logspace Reduktionen, die Klassen NLOGSPACE und PSPACE

Inhalte der Lehrveranstaltung (cont’d)

Reguläre Sprachen

deterministische Automaten, nichtdeterministische Automaten, endliche Automaten mit Epsilon-Übergängen, reguläre Ausdrücke, Abgeschossenheit, Schleifenlemma

Berechenbarkeitstheorie

deterministische TM, nichtdeterministische TM, universelle TMs, Äquivalenzen

Komplexitätstheorie

Grundlagen, die Klassen P und NP, polynomielle Reduktionen, logspace Reduktionen, die Klassen NLOGSPACE und PSPACE

Rekursiv, Rekursiv Aufzählbar, Co-Rekursiv Aufzählbar

Definition

Eine Sprache L (oder allgemein eine Menge) heißt

- rekursiv aufzählbar, wenn eine TM M existiert, sodass $L = \mathrm{L}(M)$
- co-rekursiv aufzählbar wenn sie das Komplement einer rekursiv aufzählbaren Sprache ist
- rekursiv, wenn es eine totale TM M gibt, sodass $L = \mathrm{L}(M)$

Rekursiv, Rekursiv Aufzählbar, Co-Rekursiv Aufzählbar

Definition

Eine Sprache L (oder allgemein eine Menge) heißt

- rekursiv aufzählbar, wenn eine TM M existiert, sodass $L = \mathrm{L}(M)$
- co-rekursiv aufzählbar wenn sie das Komplement einer rekursiv aufzählbaren Sprache ist
- rekursiv, wenn es eine totale TM M gibt, sodass $L = \mathrm{L}(M)$

Church-Turing-These

Jedes algorithmisch lösbare Problem ist auch mit Hilfe einer Turingmaschine lösbar

Rekursiv, Rekursiv Aufzählbar, Co-Rekursiv Aufzählbar

Definition

Eine Sprache L (oder allgemein eine Menge) heißt

- rekursiv aufzählbar, wenn eine TM M existiert, sodass $L = \mathrm{L}(M)$
- co-rekursiv aufzählbar wenn sie das Komplement einer rekursiv aufzählbaren Sprache ist
- rekursiv, wenn es eine totale TM M gibt, sodass $L = \mathrm{L}(M)$

Church-Turing-These

Jedes algorithmisch lösbare Problem ist auch mit Hilfe einer Turingmaschine lösbar(das gilt auch für Quantenrechner)

Satz

Sei Σ ein Alphabet und $L \subseteq \Sigma^*$ rekursiv; dann ist $\sim L$ rekursiv

Satz

Sei Σ ein Alphabet und $L \subseteq \Sigma^*$ rekursiv; dann ist $\sim L$ rekursiv

Beweis.

Da L rekursiv ist, gibt es eine totale TM M mit $L = \mathrm{L}(M)$. Wir definieren eine TM M', wobei der akzeptierende und der verwerfende Zustand von M vertauscht werden. Weil M total ist, ist auch M' total. Somit akzeptiert M' ein Wort genau dann, wenn M es verwirft und es folgt $\sim L = \mathrm{L}(M')$, d.h. $\sim L$ ist rekursiv. ∎

Satz

Sei Σ ein Alphabet und $L \subseteq \Sigma^*$ rekursiv; dann ist $\sim L$ rekursiv

Beweis.

Da L rekursiv ist, gibt es eine totale TM M mit $L = \mathrm{L}(M)$. Wir definieren eine TM M', wobei der akzeptierende und der verwerfende Zustand von M vertauscht werden. Weil M total ist, ist auch M' total. Somit akzeptiert M' ein Wort genau dann, wenn M es verwirft und es folgt $\sim L = \mathrm{L}(M')$, d.h. $\sim L$ ist rekursiv. ∎

Satz

Jede rekursive Menge ist rekursiv aufzählbar. Andererseits ist nicht jede rekursiv aufzählbare Menge rekursiv.

Satz

Sei Σ ein Alphabet und $L \subseteq \Sigma^*$ rekursiv; dann ist $\sim L$ rekursiv

Beweis.

Da L rekursiv ist, gibt es eine totale TM M mit $L = \mathrm{L}(M)$. Wir definieren eine TM M', wobei der akzeptierende und der verwerfende Zustand von M vertauscht werden. Weil M total ist, ist auch M' total. Somit akzeptiert M' ein Wort genau dann, wenn M es verwirft und es folgt $\sim L = \mathrm{L}(M')$, d.h. $\sim L$ ist rekursiv. ∎

Satz

Jede rekursive Menge ist rekursiv aufzählbar. Andererseits ist nicht jede rekursiv aufzählbare Menge rekursiv.

Beweis.

Der erste Teil des Satzes ist eine Konsequenz der Definitionen; den zweiten Teil holen wir nach ∎

Satz

Wenn L und $\sim L$ rekursiv aufzählbar sind, dann ist L rekursiv.

Satz

Wenn L und $\sim L$ rekursiv aufzählbar sind, dann ist L rekursiv.

Beweis.

- $\exists$ TM M_1, M_2 mit $L = \mathrm{L}(M_1)$ und $\sim(L) = \mathrm{L}(M_2)$

Satz

Wenn L und $\sim$L rekursiv aufzählbar sind, dann ist L rekursiv.

Beweis.

- $\exists$ TM M_1, M_2 mit $L = \mathrm{L}(M_1)$ und $\sim(L) = \mathrm{L}(M_2)$
- definiere TM M', sodass das Band zwei Hälften hat (oder eine 2-Band TM):

b	$\hat{b}$	a	b	a	a	a	a	b	a	a	a	…
c	c	c	d	d	d	c	$\hat{c}$	d	c	d	c	

Satz

Wenn L und $\sim$L rekursiv aufzählbar sind, dann ist L rekursiv.

Beweis.

- $\exists$ TM M_1, M_2 mit $L = \mathrm{L}(M_1)$ und $\sim(L) = \mathrm{L}(M_2)$
- definiere TM M', sodass das Band zwei Hälften hat (oder eine 2-Band TM):

b	$\hat{b}$	a	b	a	a	a	a	b	a	a	a	$\dots$
c	c	c	d	d	d	c	$\hat{c}$	d	c	d	c	

- M_1 wird auf der oberen und M_2 auf der unteren Hälfte simuliert

Satz

Wenn L und $\sim L$ rekursiv aufzählbar sind, dann ist L rekursiv.

Beweis.

- $\exists$ TM M_1, M_2 mit $L = \mathrm{L}(M_1)$ und $\sim(L) = \mathrm{L}(M_2)$
- definiere TM M', sodass das Band zwei Hälften hat (oder eine 2-Band TM):

b	$\hat{b}$	a	b	a	a	a	a	b	a	a	a	...
c	c	c	d	d	d	c	$\hat{c}$	d	c	d	c	

- M_1 wird auf der oberen und M_2 auf der unteren Hälfte simuliert
- wenn M_1 x akzeptiert, M' akzeptiert x
- wenn M_2 x akzeptiert, M' verwirft x

Satz

Wenn L und $\sim L$ rekursiv aufzählbar sind, dann ist L rekursiv.

Beweis.

- $\exists$ TM M_1, M_2 mit $L = \mathrm{L}(M_1)$ und $\sim(L) = \mathrm{L}(M_2)$
- definiere TM M', sodass das Band zwei Hälften hat (oder eine 2-Band TM):

b	$\hat{b}$	a	b	a	a	a	a	b	a	a	a	...
c	c	c	d	d	d	c	$\hat{c}$	d	c	d	c	

- M_1 wird auf der oberen und M_2 auf der unteren Hälfte simuliert
- wenn M_1 x akzeptiert, M' akzeptiert x
- wenn M_2 x akzeptiert, M' verwirft x ∎

Definition

Eine nichtdeterministische (einbändige) Turingmaschine N (kurz NTM) ist ein 9-Tupel

$$N = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)\,,$$

sodass

1. Q eine endliche Menge von Zuständen,
2. Σ eine endliche Menge von Eingabesymbolen,
3. $\Gamma \supseteq \Sigma$ eine endliche Menge von Bandsymbolen,
4. $\vdash \in \Gamma \setminus \Sigma$, der linke Endmarker,
5. $\sqcup \in \Gamma \setminus \Sigma$, $(\sqcup \neq \vdash)$, das Leerzeichen,
6. $\delta \colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\})$ die Übergangsfunktion,
7. $s \in Q$, der Startzustand,
8. $t \in Q$, der akzeptierende Zustand und
9. $r \in Q$, der verwerfende Zustand mit $t \neq r$.

Beispiel

Für die NTM $N = (\{s, q, r, t\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1\}, \vdash, \sqcup, \delta, s, t, r)$ mit δ gegeben durch

	$\vdash$	0	1	$\sqcup$
s	$\{(s, \vdash, \mathrm{R})\}$	$\{(s, 0, \mathrm{R}), (q, 0, \mathrm{R})\}$	$\{(s, 1, \mathrm{R})\}$	$\{(r, \sqcup, \mathrm{R})\}$
q	$\cdot$	$\{(r, 0, \mathrm{R})\}$	$\{(t, 1, \mathrm{R})\}$	$\{(r, \sqcup, \mathrm{R})\}$

Beispiel

Für die NTM $N = (\{s, q, r, t\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1\}, \vdash, \sqcup, \delta, s, t, r)$ mit δ gegeben durch

	$\vdash$	0	1	$\sqcup$
s	$\{(s, \vdash, \mathrm{R})\}$	$\{(s, 0, \mathrm{R}), (q, 0, \mathrm{R})\}$	$\{(s, 1, \mathrm{R})\}$	$\{(r, \sqcup, \mathrm{R})\}$
q	$\cdot$	$\{(r, 0, \mathrm{R})\}$	$\{(t, 1, \mathrm{R})\}$	$\{(r, \sqcup, \mathrm{R})\}$

es gilt $0011 \in \mathrm{L}(N)$

Beispiel

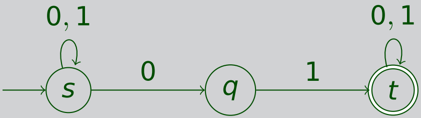
Für die NTM $N = (\{s, q, r, t\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1\}, \vdash, \sqcup, \delta, s, t, r)$ mit δ gegeben durch

	$\vdash$	0	1	$\sqcup$
s	$\{(s, \vdash, \mathrm{R})\}$	$\{(s, 0, \mathrm{R}), (q, 0, \mathrm{R})\}$	$\{(s, 1, \mathrm{R})\}$	$\{(r, \sqcup, \mathrm{R})\}$
q	$\cdot$	$\{(r, 0, \mathrm{R})\}$	$\{(t, 1, \mathrm{R})\}$	$\{(r, \sqcup, \mathrm{R})\}$

es gilt $0011 \in \mathrm{L}(N)$

Beispiel (Fortsetzung)

N entspricht dem folgendem NEA (plus Fangzustand r):



Beispiel (Fortsetzung)

Es ergibt sich für das Wort 0011 der folgende Berechnungsbaum:

$$
\begin{array}{c}
(s, \vdash 0011\sqcup^{\infty}, 0) \\
\downarrow \\
(s, \vdash 0011\sqcup^{\infty}, 1)
\end{array}
$$

Beispiel (Fortsetzung)

Es ergibt sich für das Wort 0011 der folgende Berechnungsbaum:

$$
\begin{tikzcd}[column sep=tiny]
& (s, \vdash 0011\sqcup^{\infty}, 0) \arrow[d] & \\
& (s, \vdash 0011\sqcup^{\infty}, 1) \arrow[dl] \arrow[dr] & \\
(s, \vdash 0011\sqcup^{\infty}, 2) & & (q, \vdash 0011\sqcup^{\infty}, 2)
\end{tikzcd}
$$

Beispiel (Fortsetzung)

Es ergibt sich für das Wort 0011 der folgende Berechnungsbaum:

$$
\begin{tikzcd}
 & & (s, \vdash 0011\sqcup^{\infty}, 0) \arrow[d] & \\
 & & (s, \vdash 0011\sqcup^{\infty}, 1) \arrow[dl] \arrow[dr] & \\
 & (s, \vdash 0011\sqcup^{\infty}, 2) \arrow[dl] \arrow[dr] & & (q, \vdash 0011\sqcup^{\infty}, 2) \\
(s, \vdash 0011\sqcup^{\infty}, 3) & & (q, \vdash 0011\sqcup^{\infty}, 3) &
\end{tikzcd}
$$

Beispiel (Fortsetzung)

Es ergibt sich für das Wort 0011 der folgende Berechnungsbaum:

$$
\begin{tikzcd}[column sep=tiny]
 & & (s, \vdash 0011\sqcup^\infty, 0) \arrow[d] & & \\
 & & (s, \vdash 0011\sqcup^\infty, 1) \arrow[dl] \arrow[dr] & & \\
 & (s, \vdash 0011\sqcup^\infty, 2) \arrow[dl] \arrow[dr] & & (q, \vdash 0011\sqcup^\infty, 2) \arrow[dr] & \\
(s, \vdash 0011\sqcup^\infty, 3) & & (q, \vdash 0011\sqcup^\infty, 3) & & (r, \vdash 0011\sqcup^\infty, 3)
\end{tikzcd}
$$

Beispiel (Fortsetzung)

Es ergibt sich für das Wort 0011 der folgende Berechnungsbaum:

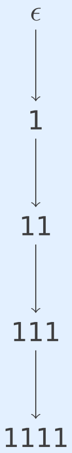
$$\begin{tikzcd}[column sep=tiny]
 & & (s,\vdash 0011\sqcup^\infty,0) \arrow[d] & & \\
 & & (s,\vdash 0011\sqcup^\infty,1) \arrow[dl] \arrow[dr] & & \\
 & (s,\vdash 0011\sqcup^\infty,2) \arrow[dl] \arrow[dr] & & (q,\vdash 0011\sqcup^\infty,2) \arrow[dr] & \\
(s,\vdash 0011\sqcup^\infty,3) \arrow[d] & & (q,\vdash 0011\sqcup^\infty,3) & & (r,\vdash 0011\sqcup^\infty,3) \\
\vdots & & & &
\end{tikzcd}$$

Beispiel (Fortsetzung)

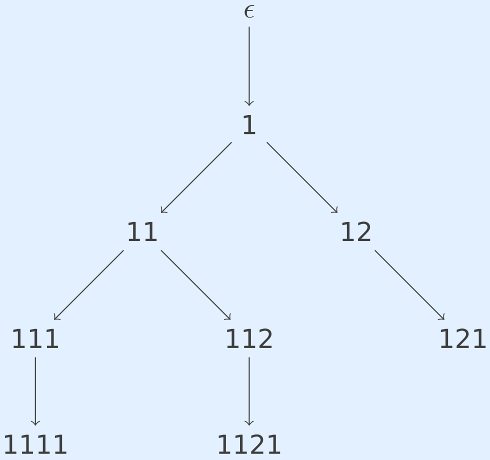
Es ergibt sich für das Wort 0011 der folgende Berechnungsbaum:

$$
\begin{tikzcd}[column sep=tiny]
 & & (s, \vdash 0011 \sqcup^\infty, 0) \arrow[d] & & \\
 & & (s, \vdash 0011 \sqcup^\infty, 1) \arrow[dl] \arrow[dr] & & \\
 & (s, \vdash 0011 \sqcup^\infty, 2) \arrow[dl] \arrow[dr] & & (q, \vdash 0011 \sqcup^\infty, 2) \arrow[dr] & \\
(s, \vdash 0011 \sqcup^\infty, 3) \arrow[d] & & (q, \vdash 0011 \sqcup^\infty, 3) \arrow[d] & & (r, \vdash 0011 \sqcup^\infty, 3) \\
\vdots & & (\textcolor{red}{t}, \vdash 0011 \sqcup^\infty, 4) & &
\end{tikzcd}
$$

Adressierung des Berechnungsbaumes



deterministisch



nichtdeterministisch

Satz

Sei N eine NTM. Dann existiert eine DTM M, sodass $\mathsf{L}(M) = \mathsf{L}(N)$. Umgekehrt wird jede von einer DTM akzeptierte Sprache auch von einer NTM akzeptiert.

Satz

Sei N eine NTM. Dann existiert eine DTM M, sodass $\mathrm{L}(M) = \mathrm{L}(N)$. Umgekehrt wird jede von einer DTM akzeptierte Sprache auch von einer NTM akzeptiert.

Beweis des ersten Teils.

Sei N eine NTM; wir konstruieren eine dreibändige DTM M mit $\mathrm{L}(N) = \mathrm{L}(M)$; sei x das Eingabewort für N:

Satz

Sei N eine NTM. Dann existiert eine DTM M, sodass $\mathrm{L}(M) = \mathrm{L}(N)$*. Umgekehrt wird jede von einer DTM akzeptierte Sprache auch von einer NTM akzeptiert.*

Beweis des ersten Teils.

Sei N eine NTM; wir konstruieren eine dreibändige DTM M mit $\mathrm{L}(N) = \mathrm{L}(M)$; sei x das Eingabewort für N:

- Das erste Band von M wird immer nur diese Eingabe x enthalten

Satz

Sei N eine NTM. Dann existiert eine DTM M, sodass $\mathrm{L}(M) = \mathrm{L}(N)$. Umgekehrt wird jede von einer DTM akzeptierte Sprache auch von einer NTM akzeptiert.

Beweis des ersten Teils.

Sei N eine NTM; wir konstruieren eine dreibändige DTM M mit $\mathrm{L}(N) = \mathrm{L}(M)$; sei x das Eingabewort für N:

- Das erste Band von M wird immer nur diese Eingabe x enthalten
- Auf dem zweiten Band simulieren wir die Rechenschritte von N bezüglich eines Weges im Berechnungsbaum

Satz

Sei N eine NTM. Dann existiert eine DTM M, sodass $\mathrm{L}(M) = \mathrm{L}(N)$. Umgekehrt wird jede von einer DTM akzeptierte Sprache auch von einer NTM akzeptiert.

Beweis des ersten Teils.

Sei N eine NTM; wir konstruieren eine dreibändige DTM M mit $\mathrm{L}(N) = \mathrm{L}(M)$; sei x das Eingabewort für N:

- Das erste Band von M wird immer nur diese Eingabe x enthalten
- Auf dem zweiten Band simulieren wir die Rechenschritte von N bezüglich eines Weges im Berechnungsbaum
- Schließlich dient das dritte Band dazu, den aktuellen Weg zu adressieren

Satz

Sei N eine NTM. Dann existiert eine DTM M, sodass $\mathrm{L}(M) = \mathrm{L}(N)$. *Umgekehrt wird jede von einer DTM akzeptierte Sprache auch von einer NTM akzeptiert.*

Beweis des ersten Teils.

Sei N eine NTM; wir konstruieren eine dreibändige DTM M mit $\mathrm{L}(N) = \mathrm{L}(M)$; sei x das Eingabewort für N:

- Das erste Band von M wird immer nur diese Eingabe x enthalten
- Auf dem zweiten Band simulieren wir die Rechenschritte von N bezüglich eines Weges im Berechnungsbaum
- Schließlich dient das dritte Band dazu, den aktuellen Weg zu adressieren

Mit Hilfe der Adressierung von Wegen können wir nun das Verhalten von N in M simulieren

Beweis (Fortsetzung).

1. Anfangs enthält Band 1 von M das Eingabewort x, die Bänder 2 und 3 sind leer

Beweis (Fortsetzung).

1. Anfangs enthält Band 1 von M das Eingabewort x, die Bänder 2 und 3 sind leer
2. Kopiere den Inhalt von Band 1 auf Band 2

Beweis (Fortsetzung).

1. Anfangs enthält Band 1 von M das Eingabewort x, die Bänder 2 und 3 sind leer
2. Kopiere den Inhalt von Band 1 auf Band 2
3. Verwende Band 2, um die Rechenschritte von N auf x zu simulieren. Bei jeder Stelle in der Berechnung, in der die Übergangsfunktion δ mehrere Möglichkeiten zulässt, sieht M auf Band 3 nach, welche Möglichkeit gewählt werden soll

Beweis (Fortsetzung).

1. Anfangs enthält Band 1 von M das Eingabewort x, die Bänder 2 und 3 sind leer
2. Kopiere den Inhalt von Band 1 auf Band 2
3. Verwende Band 2, um die Rechenschritte von N auf x zu simulieren. Bei jeder Stelle in der Berechnung, in der die Übergangsfunktion δ mehrere Möglichkeiten zulässt, sieht M auf Band 3 nach, welche Möglichkeit gewählt werden soll
4. Wenn die Simulation von N akzeptiert, dann akzeptiere

Beweis (Fortsetzung).

1. Anfangs enthält Band 1 von M das Eingabewort x, die Bänder 2 und 3 sind leer
2. Kopiere den Inhalt von Band 1 auf Band 2
3. Verwende Band 2, um die Rechenschritte von N auf x zu simulieren. Bei jeder Stelle in der Berechnung, in der die Übergangsfunktion δ mehrere Möglichkeiten zulässt, sieht M auf Band 3 nach, welche Möglichkeit gewählt werden soll
4. Wenn die Simulation von N akzeptiert, dann akzeptiere
5. Ersetze das Wort auf Band 3 durch seinen unmittelbaren Nachfolger in der graduiert-lexikographischen Ordnung auf Wörtern $\leq_{\text{gradlex}}$

Beweis (Fortsetzung).

1. Anfangs enthält Band 1 von M das Eingabewort x, die Bänder 2 und 3 sind leer
2. Kopiere den Inhalt von Band 1 auf Band 2
3. Verwende Band 2, um die Rechenschritte von N auf x zu simulieren. Bei jeder Stelle in der Berechnung, in der die Übergangsfunktion δ mehrere Möglichkeiten zulässt, sieht M auf Band 3 nach, welche Möglichkeit gewählt werden soll
4. Wenn die Simulation von N akzeptiert, dann akzeptiere
5. Ersetze das Wort auf Band 3 durch seinen unmittelbaren Nachfolger in der graduiert-lexikographischen Ordnung auf Wörtern $\leq_{\text{gradlex}}$
6. Gehe zu Schritt (2)

Beweis (Fortsetzung).

1. Anfangs enthält Band 1 von M das Eingabewort x, die Bänder 2 und 3 sind leer
2. Kopiere den Inhalt von Band 1 auf Band 2
3. Verwende Band 2, um die Rechenschritte von N auf x zu simulieren. Bei jeder Stelle in der Berechnung, in der die Übergangsfunktion δ mehrere Möglichkeiten zulässt, sieht M auf Band 3 nach, welche Möglichkeit gewählt werden soll
4. Wenn die Simulation von N akzeptiert, dann akzeptiere
5. Ersetze das Wort auf Band 3 durch seinen unmittelbaren Nachfolger in der graduiert-lexikographischen Ordnung auf Wörtern $\leq_{\text{gradlex}}$
6. Gehe zu Schritt (2)

∎

Entscheidbarkeit, Unentscheidbarkeit

Definition

Sei Σ ein Alphabet. Eine Eigenschaft P von Wörtern über Σ heißt

- entscheidbar genau dann, wenn die Menge $\{x \in \Sigma^* \mid x \text{ hat Eigenschaft } P\}$ rekursiv ist
- semi-entscheidbar genau dann, wenn die Menge $\{x \in \Sigma^* \mid x \text{ hat Eigenschaft } P\}$ rekursiv aufzählbar ist

Entscheidbarkeit, Unentscheidbarkeit

Definition

Sei Σ ein Alphabet. Eine Eigenschaft P von Wörtern über Σ heißt

- entscheidbar genau dann, wenn die Menge $\{x \in \Sigma^* \mid x \text{ hat Eigenschaft } P\}$ rekursiv ist
- semi-entscheidbar genau dann, wenn die Menge $\{x \in \Sigma^* \mid x \text{ hat Eigenschaft } P\}$ rekursiv aufzählbar ist

Beispiel

Sei $P(x) := x$ ist ein Palindrom gerader Länge; dann ist P entscheidbar

Entscheidbarkeit, Unentscheidbarkeit

Definition

Sei Σ ein Alphabet. Eine Eigenschaft P von Wörtern über Σ heißt

- entscheidbar genau dann, wenn die Menge $\{x \in \Sigma^* \mid x \text{ hat Eigenschaft } P\}$ rekursiv ist
- semi-entscheidbar genau dann, wenn die Menge $\{x \in \Sigma^* \mid x \text{ hat Eigenschaft } P\}$ rekursiv aufzählbar ist

Beispiel

Sei $P(x) := x$ ist ein Palindrom gerader Länge; dann ist P entscheidbar

Beispiel

Jedes entscheidbare Problem ist semi-entscheidbar

Bemerkung

Ein Problem P ist

- semi-entscheidbar, wenn es eine TM M gibt, deren Sprache alle Wörter sind, welche die Eigenschaft P haben

Bemerkung

Ein Problem P ist

- semi-entscheidbar, wenn es eine TM M gibt, deren Sprache alle Wörter sind, welche die Eigenschaft P haben
- entscheidbar, wenn es eine totale TM M gibt, sodass M genau jene Wörter akzeptiert, welche die Eigenschaft P haben

Grenzen der Berechenbarkeit

Was können Turingmaschinen?

- Codierung von Turingmaschinen
- universelle Turingmaschine; eine TM als Universalinterpreter

Grenzen der Berechenbarkeit

Was können Turingmaschinen?

- Codierung von Turingmaschinen
- universelle Turingmaschine; eine TM als Universalinterpreter

Warum hat die Informatik dann Grenzen?

- Definition einer TM mit einem anderen Verhalten (in Bezug auf Termination) $\overline{LD}$ als alle anderen
- Unentscheidbarkeit des Halteproblems

Codierung von TMs

TMs können codiert werden indem alle notwendigen Informationen als Wörter über $\{0, 1\}$ dargestellt werden:

1. Anzahl der Zustände
2. Übergangsfunktion
3. Eingabe- und Bandalphabet
4. ...

Codierung von TMs

TMs können codiert werden indem alle notwendigen Informationen als Wörter über $\{0, 1\}$ dargestellt werden:

1. Anzahl der Zustände
2. Übergangsfunktion
3. Eingabe- und Bandalphabet
4. ...

Beispiel

sei $M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$ eine TM; Codierung über $\{0, 1\}$

$$0^n \ 1 \ 0^m \ 1 \ 0^k \ 1 \ 0^s \ 1 \ 0^t \ 1 \ 0^r \ 1 \ 0^u \ 1 \ 0^v \ 1 \quad \cdots$$

entspricht $Q = \{0, \ldots, n-1\}$, $\Gamma = \{0, \ldots, m-1\}$, $\Sigma = \{0, \ldots, k-1\}$, $(k \leqslant m)$, s Startzustand, t akzeptierend, r verwerfend, u linker Endmarker, v Blanksymbol

Codierung von TMs

TMs können codiert werden indem alle notwendigen Informationen als Wörter über $\{0,1\}$ dargestellt werden:

1. Anzahl der Zustände
2. Übergangsfunktion
3. Eingabe- und Bandalphabet
4. ...

Beispiel

sei $M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$ eine TM; Codierung über $\{0,1\}$

$$0^n \; 1 \; 0^m \; 1 \; 0^k \; 1 \; 0^s \; 1 \; 0^t \; 1 \; 0^r \; 1 \; 0^u \; 1 \; 0^v \; 1 \quad \cdots$$

entspricht $Q = \{0, \dots, n-1\}$, $\Gamma = \{0, \dots, m-1\}$, $\Sigma = \{0, \dots, k-1\}$, $(k \leqslant m)$, s Startzustand, t akzeptierend, r verwerfend, u linker Endmarker, v Blanksymbol; das Zeichen 1 dient als Trennzeichen der Codierung

Beispiel (Fortsetzung)

betrachte M und kodiere $\delta(p, a) = (q, b, d)$, wobei $c = 0$ wenn $d = \mathrm{L}$ und $c = 1$ wenn $d = \mathrm{R}$

$$0^p \; 1 \; 0^a \; 1 \; 0^q \; 1 \; 0^b \; 1 \; 0^c \; 1$$

Beispiel (Fortsetzung)

betrachte M und kodiere $\delta(p, a) = (q, b, d)$, wobei $c = 0$ wenn $d = \mathrm{L}$ und $c = 1$ wenn $d = \mathrm{R}$

$$0^p \ 1 \ 0^a \ 1 \ 0^q \ 1 \ 0^b \ 1 \ 0^c \ 1$$

Beispiel

Wir kodieren $M' = (\{s, p, t, r\}, \{0, 1\}, \{0, 1, \vdash, \sqcup\}, \vdash, \sqcup, \delta, s, t, r)$ mit

	$\vdash$	0	1	$\sqcup$
s	$(s, \vdash, \mathrm{R})$	$(s, 0, \mathrm{R})$	$(s, 1, \mathrm{R})$	$(p, \sqcup, \mathrm{L})$
p	$(t, \vdash, \mathrm{R})$	$(t, 1, \mathrm{L})$	$(p, 0, \mathrm{L})$	$\cdot$

Zunächst erhalten wir

$$\underbrace{0000}_{n=4} 1 \underbrace{0000}_{m=4} 1 \underbrace{00}_{k=2} 1 \underbrace{\epsilon}_{s} 1 \underbrace{00}_{t} 1 \underbrace{000}_{r} 1 \underbrace{00}_{\vdash} 1 \underbrace{000}_{\sqcup} 1 \cdots$$

und etwa $\delta(p, \vdash) = (t, \vdash, \mathrm{R})$ wird zu $010^2 10^2 10^2 101$

Definition

eine TM U heißt universell (UTM), wenn bei Eingabe

Definition

eine TM U heißt universell (UTM), wenn bei Eingabe

- des Codes $\ulcorner M \urcorner$ einer TM M
- und des Codes $\ulcorner x \urcorner$ einer Eingabe x für M

Definition

eine TM U heißt universell (UTM), wenn bei Eingabe

- des Codes $\ulcorner M \urcorner$ einer TM M
- und des Codes $\ulcorner x \urcorner$ einer Eingabe x für M

die TM U, die TM M auf x simuliert,

Definition

eine TM U heißt universell (UTM), wenn bei Eingabe

- des Codes $\ulcorner M \urcorner$ einer TM M
- und des Codes $\ulcorner x \urcorner$ einer Eingabe x für M

die TM U, die TM M auf x simuliert, das heißt

$$\mathrm{L}(U) = \{\ulcorner M \urcorner \# \ulcorner x \urcorner \mid x \in \mathrm{L}(M)\}$$

Definition

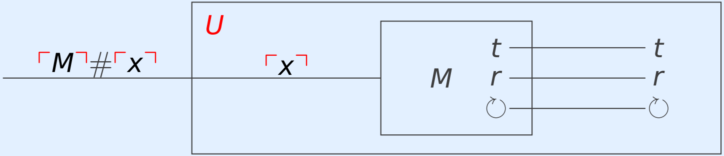
eine TM U heißt universell (UTM), wenn bei Eingabe

- des Codes $\ulcorner M \urcorner$ einer TM M
- und des Codes $\ulcorner x \urcorner$ einer Eingabe x für M

die TM U, die TM M auf x simuliert, das heißt

$$L(U) = \{\ulcorner M \urcorner \# \ulcorner x \urcorner \mid x \in L(M)\}$$

UTM schematisch



Simulation durch eine Universelle Turingmaschine

Notation

Vereinfachend läßt man oft den Hinweis auf die Kodierung weg:

$$\mathsf{L}(U) = \{M\#x \mid x \in \mathsf{L}(M)\}$$

Simulation durch eine Universelle Turingmaschine

Notation

Vereinfachend läßt man oft den Hinweis auf die Kodierung weg:

$$\mathsf{L}(U) = \{M\#x \mid x \in \mathsf{L}(M)\}$$

Simulation

1. UTM U kontrolliert Korrektheit der Codes; wenn inkorrekt, verwirft U

Simulation durch eine Universelle Turingmaschine

Notation

Vereinfachend läßt man oft den Hinweis auf die Kodierung weg:

$$\mathrm{L}(U) = \{M\#x \mid x \in \mathrm{L}(M)\}$$

Simulation

1. UTM U kontrolliert Korrektheit der Codes; wenn inkorrekt, verwirft U
2. U simuliert M mit 3 Bändern auf der Eingabe x
   - Band 1 enthält die Beschreibung von M
   - Band 2 enthält das dekodierte Eingabewort x
   - Band 3 enthält den simulierten Bandinhalt des Bandes von M

Simulation durch eine Universelle Turingmaschine

Notation

Vereinfachend läßt man oft den Hinweis auf die Kodierung weg:

$$\mathsf{L}(U) = \{M\#x \mid x \in \mathsf{L}(M)\}$$

Simulation

1. UTM U kontrolliert Korrektheit der Codes; wenn inkorrekt, verwirft U
2. U simuliert M mit 3 Bändern auf der Eingabe x
   - Band 1 enthält die Beschreibung von M
   - Band 2 enthält das dekodierte Eingabewort x
   - Band 3 enthält den simulierten Bandinhalt des Bandes von M
3. wenn M akzeptiert, so akzeptiert U; wenn M verwirft, so verwirft U

Lemma

Angenommen U eine UTM und M eine beliebige TM. Dann existiert eine Spezialisierung von U, genannt U_M, die M auf allen Eingaben simuliert.

Lemma

Angenommen U eine UTM und M eine beliebige TM. Dann existiert eine Spezialisierung von U, genannt U_M, die M auf allen Eingaben simuliert.

Beweis.

- Wir betrachten eine Variante U' von U, sodass das zweite Band von U' die Beschreibung der zu simulierenden TMs enthält und das erste Band die (dekodierte) Eingabe
- Nun spezialisieren wir U' zur gesuchten TM U_M indem wir den Code von M fix auf das zweite Band schreiben (also hardcoden)
- Nach Definition führt U_M alle Schritte von M auf der Eingabe x aus

Lemma

Angenommen U eine UTM und M eine beliebige TM. Dann existiert eine Spezialisierung von U, genannt U_M, die M auf allen Eingaben simuliert.

Beweis.

- Wir betrachten eine Variante U' von U, sodass das zweite Band von U' die Beschreibung der zu simulierenden TMs enthält und das erste Band die (dekodierte) Eingabe
- Nun spezialisieren wir U' zur gesuchten TM U_M indem wir den Code von M fix auf das zweite Band schreiben (also hardcoden)
- Nach Definition führt U_M alle Schritte von M auf der Eingabe x aus

■

Lemma

Angenommen U eine UTM und M eine beliebige TM. Dann existiert eine Spezialisierung von U, genannt U_M, die M auf allen Eingaben simuliert.

Beweis.

- Wir betrachten eine Variante U' von U, sodass das zweite Band von U' die Beschreibung der zu simulierenden TMs enthält und das erste Band die (dekodierte) Eingabe
- Nun spezialisieren wir U' zur gesuchten TM U_M indem wir den Code von M fix auf das zweite Band schreiben (also hardcoden)
- Nach Definition führt U_M alle Schritte von M auf der Eingabe x aus

∎

Bemerkung

Metaprogrammierung oder Programmiermakros stammen von UTMs

Definition

definiere Halteproblem und Zugehörigkeitsproblem von TMs

$$\mathrm{HP} := \{M\#x \mid M \text{ hält bei Eingabe } x\}$$

$$\mathrm{MP} := \{M\#x \mid x \in \mathrm{L}(M)\}$$

Definition

definiere Halteproblem und Zugehörigkeitsproblem von TMs

$$\mathrm{HP} := \{M\#x \mid M \text{ hält bei Eingabe } x\}$$

$$\mathrm{MP} := \{M\#x \mid x \in \mathrm{L}(M)\}$$

Definition

1. M_x ist TM (mit Eingabealphabet $\{0, 1\}$), deren Code (mit Kodierungsalphabet $\{0, 1\}$) gleich x
2. wenn x kein Code, definiere M_x beliebig

Definition

definiere Halteproblem und Zugehörigkeitsproblem von TMs

$$\mathrm{HP} := \{M\#x \mid M \text{ hält bei Eingabe } x\}$$

$$\mathrm{MP} := \{M\#x \mid x \in \mathrm{L}(M)\}$$

Definition

1. M_x ist TM (mit Eingabealphabet $\{0, 1\}$), deren Code (mit Kodierungsalphabet $\{0, 1\}$) gleich x
2. wenn x kein Code, definiere M_x beliebig

Definition

definiere Halteproblem und Zugehörigkeitsproblem von TMs

$$\text{HP} := \{M\#x \mid M \text{ hält bei Eingabe } x\}$$

$$\text{MP} := \{M\#x \mid x \in \text{L}(M)\}$$

Definition

1. M_x ist TM (mit Eingabealphabet $\{0, 1\}$), deren Code (mit Kodierungsalphabet $\{0, 1\}$) gleich x
2. wenn x kein Code, definiere M_x beliebig

Aufzählung aller Turingmaschinen

$$M_\epsilon, M_0, M_1, M_{00}, M_{01}, M_{10}, M_{11}, M_{000}, \ldots$$

(geordnet bezüglich der graduiert lexikographischen Ordnung)

Definition

definiere Halteproblem und Zugehörigkeitsproblem von TMs

$$\mathrm{HP} := \{M\#x \mid M \text{ hält bei Eingabe } x\}$$

$$\mathrm{MP} := \{M\#x \mid x \in \mathrm{L}(M)\}$$

Definition

1. M_x ist TM (mit Eingabealphabet $\{0, 1\}$), deren Code (mit Kodierungsalphabet $\{0, 1\}$) gleich x
2. wenn x kein Code, definiere M_x beliebig

Aufzählung aller Turingmaschinen

$$M_\epsilon, M_0, M_1, M_{00}, M_{01}, M_{10}, M_{11}, M_{000}, \ldots$$

(geordnet bezüglich der graduiert lexikographischen Ordnung)

Zweidimensionale Matrix

Indiziert mit Wörtern $w \in \{0,1\}^*$ und andererseits mit den Turingmaschinen

Zweidimensionale Matrix

Indiziert mit Wörtern $w \in \{0,1\}^*$ und andererseits mit den Turingmaschinen

	ϵ	0	1	00	01	10	11	000	001	010	...
M_ϵ	!	$\circlearrowright$	$\circlearrowright$	!	!	$\circlearrowright$	!	$\circlearrowright$	!	!	
M_0	$\circlearrowright$	$\circlearrowright$	!	!	$\circlearrowright$	!	!	$\circlearrowright$	$\circlearrowright$	!	
M_1	$\circlearrowright$	!	$\circlearrowright$	!	$\circlearrowright$	!	!	$\circlearrowright$	$\circlearrowright$	!	
M_{00}	!	$\circlearrowright$	$\circlearrowright$	!	!	!	!	$\circlearrowright$	$\circlearrowright$	!	
M_{01}	!	!	!	!	$\circlearrowright$	$\circlearrowright$	$\circlearrowright$	!	!	$\circlearrowright$	...
M_{10}	!	!	$\circlearrowright$	!	!	$\circlearrowright$	!	!	$\circlearrowright$	!	
M_{11}	!	!	$\circlearrowright$	$\circlearrowright$	!	$\circlearrowright$	!	$\circlearrowright$	!	$\circlearrowright$	
M_{000}	!	!	!	!	$\circlearrowright$	!	!	$\circlearrowright$	!	$\circlearrowright$	
M_{001}	$\circlearrowright$	!	!	!	!	$\circlearrowright$	!	!	!	!	
$\vdots$						$\vdots$					$\ddots$

Behauptung

die dem Komplement der Diagonale entsprechende Sprache $\overline{LD}$ wird von keiner TM in der Aufzählung akzeptiert

Behauptung

die dem Komplement der Diagonale entsprechende Sprache $\overline{LD}$ wird von keiner TM in der Aufzählung akzeptiert

Beweis.

sei $\Sigma \supseteq \{!, \circlearrowright\}$ ein Alphabet

$s_0, s_1, s_2, \ldots$ eine Folge unendlicher Wörter über $\{!, \circlearrowright\}$

$$
\begin{aligned}
s_0 &= s_{00}s_{01}s_{02}s_{03}s_{04}\ldots \\
s_1 &= s_{10}s_{11}s_{12}s_{13}s_{14}\ldots \\
s_1 &= s_{20}s_{21}s_{22}s_{23}s_{24}\ldots \\
&\vdots
\end{aligned}
$$

dann ist die Folge

$$
d_n = \begin{cases} \circlearrowright & \text{wenn } s_{nn} = ! \\ ! & \text{wenn } s_{nn} = \circlearrowright \end{cases}
$$

eine neue Folge

Behauptung

die dem Komplement der Diagonale entsprechende Sprache $\overline{LD}$ wird von keiner TM in der Aufzählung akzeptiert

Beweis.

sei $\Sigma \supseteq \{!, \circlearrowright\}$ ein Alphabet

$s_0, s_1, s_2, \ldots$ eine Folge unendlicher Wörter über $\{!, \circlearrowright\}$

$$
\begin{aligned}
s_0 &= s_{00}s_{01}s_{02}s_{03}s_{04}\ldots \\
s_1 &= s_{10}s_{11}s_{12}s_{13}s_{14}\ldots \\
s_1 &= s_{20}s_{21}s_{22}s_{23}s_{24}\ldots \\
&\vdots
\end{aligned}
$$

dann ist die Folge

$$
d_n = \begin{cases} \circlearrowright & \text{wenn } s_{nn} = ! \\ ! & \text{wenn } s_{nn} = \circlearrowright \end{cases}
$$

eine neue Folge ∎

Satz

HP *ist nicht rekursiv, aber rekursiv aufzählbar*

Satz

HP *ist nicht rekursiv, aber rekursiv aufzählbar*

Beweis.

wir zeigen zunächst Nicht-Rekursivität

1. angenommen $\exists$ totale TM K, sodass $\mathrm{HP} = \mathrm{L}(K)$

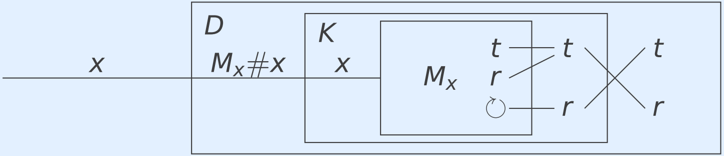
Satz

HP *ist nicht rekursiv, aber rekursiv aufzählbar*

Beweis.

wir zeigen zunächst Nicht-Rekursivität

1. angenommen $\exists$ totale TM K, sodass $\mathrm{HP} = \mathrm{L}(K)$
2. Definition von TM D



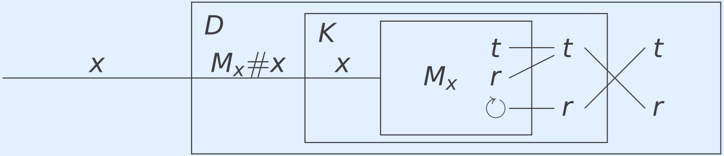
Satz

HP *ist nicht rekursiv, aber rekursiv aufzählbar*

Beweis.

wir zeigen zunächst Nicht-Rekursivität

1. angenommen $\exists$ totale TM K, sodass $\mathrm{HP} = \mathrm{L}(K)$
2. Definition von TM D



3. D akzeptiert genau die dem Komplement der Diagonale entsprechende Sprache

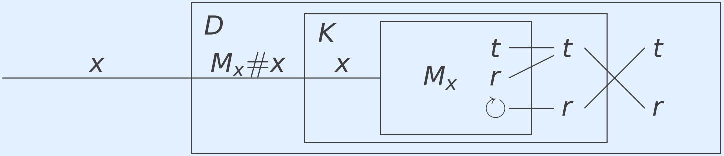
Satz

HP *ist nicht rekursiv, aber rekursiv aufzählbar*

Beweis.

wir zeigen zunächst Nicht-Rekursivität

1. angenommen $\exists$ totale TM K, sodass HP $=$ L(K)
2. Definition von TM D



3. D akzeptiert genau die dem Komplement der Diagonale entsprechende Sprache
4. Verhalten von D verschieden von jeder TM M_x in der Aufzählung

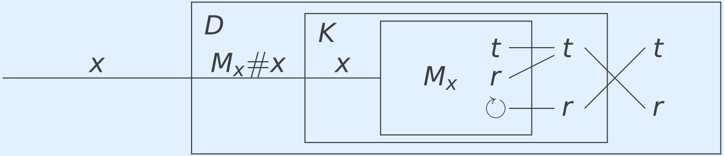
Satz

HP *ist nicht rekursiv, aber rekursiv aufzählbar*

Beweis.

wir zeigen zunächst Nicht-Rekursivität

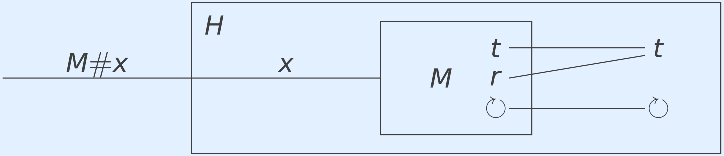
1. angenommen $\exists$ totale TM K, sodass $\mathrm{HP} = \mathrm{L}(K)$
2. Definition von TM D



3. D akzeptiert genau die dem Komplement der Diagonale entsprechende Sprache
4. Verhalten von D verschieden von jeder TM M_x in der Aufzählung

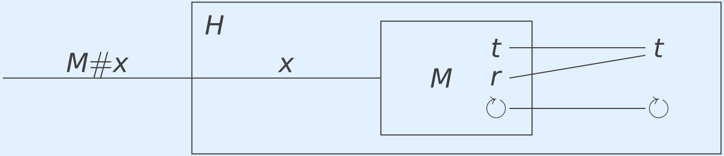
Beweis (Fortsetzung).

Nun skizzieren wir, warum HP rekursiv aufzählbar ist; dazu konstruieren wir die folgende TM H



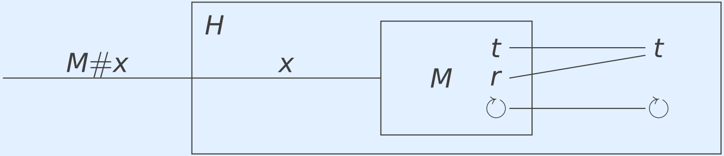
Beweis (Fortsetzung).

Nun skizzieren wir, warum HP rekursiv aufzählbar ist; dazu konstruieren wir die folgende (nicht notwendigerweise totale) TM H



Beweis (Fortsetzung).

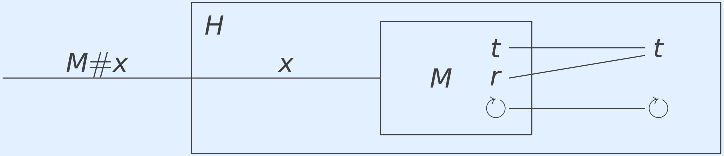
Nun skizzieren wir, warum HP rekursiv aufzählbar ist; dazu konstruieren wir die folgende (nicht notwendigerweise totale) TM H



∎

Beweis (Fortsetzung).

Nun skizzieren wir, warum HP rekursiv aufzählbar ist; dazu konstruieren wir die folgende (nicht notwendigerweise totale) TM H



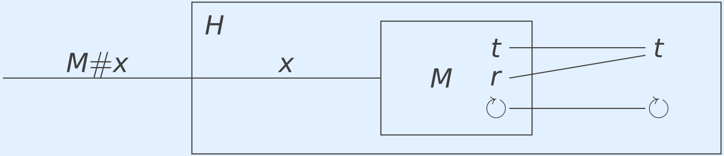
∎

Folgerung

Die Menge $\sim$HP *ist nicht rekursiv aufzählbar*

Beweis (Fortsetzung).

Nun skizzieren wir, warum HP rekursiv aufzählbar ist; dazu konstruieren wir die folgende (nicht notwendigerweise totale) TM H



∎

Folgerung

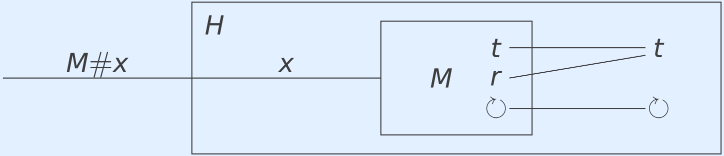
Die Menge $\sim$HP *ist nicht rekursiv aufzählbar*

Beweis.

Angenommen $\sim$HP wäre rekursiv aufzählbar; dann wären HP und $\sim$HP rekursiv aufzählbar und somit auch HP rekursiv. Widerspruch

Beweis (Fortsetzung).

Nun skizzieren wir, warum HP rekursiv aufzählbar ist; dazu konstruieren wir die folgende (nicht notwendigerweise totale) TM *H*



$\blacksquare$

Folgerung

Die Menge $\sim$HP *ist nicht rekursiv aufzählbar*

Beweis.

Angenommen $\sim$HP wäre rekursiv aufzählbar; dann wären HP und $\sim$HP rekursiv aufzählbar und somit auch HP rekursiv. Widerspruch $\blacksquare$