10b
Unentscheidbarkeit und das spezielle Halteproblem

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 10. Mai 2024
Basiert auf Folien von PD Dr. David Sabel

Entscheidbarkeit

Definition

Eine Sprache L ist entscheidbar, wenn es einen Algorithmus gibt, der bei Eingabe eines Wortes w in endlicher Zeit feststellt, ob $w \in L$ gilt oder nicht.

Entscheidbarkeit

Definition

Eine Sprache L ist entscheidbar, wenn es einen Algorithmus gibt, der bei Eingabe eines Wortes w in endlicher Zeit feststellt, ob $w \in L$ gilt oder nicht.

Präziser:

Definition

Eine Sprache $L \subseteq \Sigma^*$ ist entscheidbar, wenn die charakteristische Funktion von L, $\chi_L : \Sigma^* \to \{0, 1\}$ mit

$$\chi_L(w) = \begin{cases} 1 & \text{falls } w \in L \\ 0 & \text{sonst} \end{cases}$$

berechenbar ist.

Der χ_L-berechnende Algorithmus terminiert in jedem Fall und liefert ein Ergebnis.

Semi-Entscheidbarkeit

Definition

Eine Sprache L ist semi-entscheidbar, falls $\chi'_L : \Sigma^* \to \{0, 1\}$ mit

$$\chi'_L(w) = \begin{cases} 1 & \text{falls } w \in L \\ \text{undefiniert} & \text{sonst} \end{cases}$$

berechenbar ist.

Der χ'_L-berechnende Algorithmus terminiert nur, falls $w \in L$, und läuft anderenfalls endlos.

Zusammenhang zwischen Entscheidbarkeit und Semi-Entscheidbarkeit

Satz

Ein Sprache L ist genau dann entscheidbar, wenn L und $\overline{L}$ jeweils semi-entscheidbar sind.

Zusammenhang zwischen Entscheidbarkeit und Semi-Entscheidbarkeit

Satz

Ein Sprache L ist genau dann entscheidbar, wenn L und $\overline{L}$ jeweils semi-entscheidbar sind.

Beweis

$\Longrightarrow$ Konstruiere aus DTM, die χ_L berechnet, zwei DTMs, die χ'_L und $\chi'_{\overline{L}}$ berechnen.

Zusammenhang zwischen Entscheidbarkeit und Semi-Entscheidbarkeit

Satz

Ein Sprache L ist genau dann entscheidbar, wenn L und $\overline{L}$ jeweils semi-entscheidbar sind.

Beweis

$\Longrightarrow$ Konstruiere aus DTM, die χ_L berechnet, zwei DTMs, die χ'_L und $\chi'_{\overline{L}}$ berechnen.

$\Longleftarrow$ Gegeben seien DTMs M_L und $M_{\overline{L}}$, die χ'_L und $\chi'_{\overline{L}}$ berechnen.
Grundgedanke: Lasse M_L und $M_{\overline{L}}$ parallel laufen.
Konstruiere eine DTM, die χ_L berechnet:

1. Starte mit $i = 1$.
2. Simuliere i Schritte von M_L.
3. Wenn diese akzeptiert (mit Ausgabe 1), dann akzeptiere mit Ausgabe 1.
4. Ansonsten simuliere i Schritte von $M_{\overline{L}}$.
5. Wenn diese akzeptiert (mit Ausgabe 1), dann akzeptiere mit Ausgabe 0.
6. Ansonsten erhöhe i um 1 und starte von neuem. $\square$

Entscheidbarkeit von Komplement

Korollar

Wenn L entscheidbar ist, dann ist auch $\overline{L}$ entscheidbar.

Entscheidbarkeit von Komplement

Korollar

Wenn L entscheidbar ist, dann ist auch $\overline{L}$ entscheidbar.

Beweis Da L entscheidbar ist, sind L und $\overline{L}$ semi-entscheidbar.

Entscheidbarkeit von Komplement

Korollar

Wenn L entscheidbar ist, dann ist auch $\overline{L}$ entscheidbar.

Beweis Da L entscheidbar ist, sind L und $\overline{L}$ semi-entscheidbar.

Daher sind $\overline{\overline{L}} = L$ und $\overline{L}$ semi-entscheidbar.

Entscheidbarkeit von Komplement

Korollar

Wenn L entscheidbar ist, dann ist auch $\overline{L}$ entscheidbar.

Beweis Da L entscheidbar ist, sind L und $\overline{L}$ semi-entscheidbar.

Daher sind $\overline{\overline{L}} = L$ und $\overline{L}$ semi-entscheidbar.

Daher ist $\overline{L}$ entscheidbar. $\square$

Rekursive Aufzählbarkeit

Definition

Eine Sprache $L \subseteq \Sigma^*$ heißt rekursiv aufzählbar, falls

- $L = \emptyset$ oder
- es eine totale berechenbare Funktion $f : \mathbb{N} \rightarrow \Sigma^*$ gibt, sodass $L = \bigcup_{i \in \mathbb{N}} f(i)$.
  Man sagt dann „f zählt L auf“.

Beispiel für die rekursive Aufzählbarkeit

Lemma

Die Sprache Σ^* ist rekursiv aufzählbar.

Beispiel für die rekursive Aufzählbarkeit

Lemma

Die Sprache Σ^* ist rekursiv aufzählbar.

Beweis Wir zeigen nur den Fall $|\Sigma| = 1$. Sei $\Sigma = \{a\}$ und $n \in \mathbb{N}$.

Beispiel für die rekursive Aufzählbarkeit

Lemma

Die Sprache Σ^* ist rekursiv aufzählbar.

Beweis Wir zeigen nur den Fall $|\Sigma| = 1$. Sei $\Sigma = \{a\}$ und $n \in \mathbb{N}$.

Konstruiere eine 2-Band-Turingmaschine M, die n in Binärzahldarstellung auf das Eingabeband erhält.

Beispiel für die rekursive Aufzählbarkeit

Lemma

Die Sprache Σ^* ist rekursiv aufzählbar.

Beweis Wir zeigen nur den Fall $|\Sigma| = 1$. Sei $\Sigma = \{a\}$ und $n \in \mathbb{N}$.

Konstruiere eine 2-Band-Turingmaschine M, die n in Binärzahldarstellung auf das Eingabeband erhält.

M erzeugt auf dem anderen Band das leere Wort.

Beispiel für die rekursive Aufzählbarkeit

Lemma

Die Sprache Σ^* ist rekursiv aufzählbar.

Beweis Wir zeigen nur den Fall $|\Sigma| = 1$. Sei $\Sigma = \{a\}$ und $n \in \mathbb{N}$.

Konstruiere eine 2-Band-Turingmaschine M, die n in Binärzahldarstellung auf das Eingabeband erhält.

M erzeugt auf dem anderen Band das leere Wort.

Anschließend zählt M die Zahl auf dem Eingabeband um 1 herunter und fügt bei jedem Herunterzählen ein a hinzu.

Beispiel für die rekursive Aufzählbarkeit

Lemma

Die Sprache Σ^* ist rekursiv aufzählbar.

Beweis Wir zeigen nur den Fall $|\Sigma| = 1$. Sei $\Sigma = \{a\}$ und $n \in \mathbb{N}$.

Konstruiere eine 2-Band-Turingmaschine M, die n in Binärzahldarstellung auf das Eingabeband erhält.

M erzeugt auf dem anderen Band das leere Wort.

Anschließend zählt M die Zahl auf dem Eingabeband um 1 herunter und fügt bei jedem Herunterzählen ein a hinzu.

Dies wird wiederholt bis auf dem Eingabeband 0 steht.

Beispiel für die rekursive Aufzählbarkeit

Lemma

Die Sprache Σ^* ist rekursiv aufzählbar.

Beweis Wir zeigen nur den Fall $|\Sigma| = 1$. Sei $\Sigma = \{a\}$ und $n \in \mathbb{N}$.

Konstruiere eine 2-Band-Turingmaschine M, die n in Binärzahldarstellung auf das Eingabeband erhält.

M erzeugt auf dem anderen Band das leere Wort.

Anschließend zählt M die Zahl auf dem Eingabeband um 1 herunter und fügt bei jedem Herunterzählen ein a hinzu.

Dies wird wiederholt bis auf dem Eingabeband 0 steht.

Dann steht auf dem anderen Band $f(n) = \underbrace{a \cdots a}_{n}$. □

Rekursiv aufzählbar = semi-entscheidbar

Satz

Eine Sprache ist genau dann rekursiv aufzählbar, wenn sie semi-entscheidbar ist.

Rekursiv aufzählbar = semi-entscheidbar

Satz

Eine Sprache ist genau dann rekursiv aufzählbar, wenn sie semi-entscheidbar ist.

Beweis

$\Longrightarrow$ Sei f die totale berechenbare Funktion, die L aufzählt.
Dann berechnet der folgende Algorithmus $\chi'_L(w)$:

für $i = 0, 1, 2, 3, \ldots$ **tue**
  wenn $f(i) = w$ **dann**
    stoppe und gib 1 aus
  Ende
Ende

Rekursiv aufzählbar = semi-entscheidbar

Satz

Eine Sprache ist genau dann rekursiv aufzählbar, wenn sie semi-entscheidbar ist.

Beweis (Fortsetzung)

$\Longleftarrow$ Wenn $L = \emptyset$, dann ist L rekursiv aufzählbar.

Rekursiv aufzählbar = semi-entscheidbar

Satz

Eine Sprache ist genau dann rekursiv aufzählbar, wenn sie semi-entscheidbar ist.

Beweis (Fortsetzung)

$\Longleftarrow$ Wenn $L = \emptyset$, dann ist L rekursiv aufzählbar.

Sonst, sei M eine Turingmaschine, die χ'_L berechnet und sei $u \in L$ ein Wort. Wir müssen zeigen, dass es eine totale und berechenbare Funktion f gibt, die L aufzählt. Wir konstruieren eine Turingmaschine M', die f berechnet.

Rekursiv aufzählbar = semi-entscheidbar

Satz

Eine Sprache ist genau dann rekursiv aufzählbar, wenn sie semi-entscheidbar ist.

Beweis (Fortsetzung)

$\Longleftarrow$ Wenn $L = \emptyset$, dann ist L rekursiv aufzählbar.

Sonst, sei M eine Turingmaschine, die χ'_L berechnet und sei $u \in L$ ein Wort. Wir müssen zeigen, dass es eine totale und berechenbare Funktion f gibt, die L aufzählt. Wir konstruieren eine Turingmaschine M', die f berechnet.

Sei n eine Eingabe. Wir interpretieren n als $c(x, y)$.

Rekursiv aufzählbar = semi-entscheidbar

Satz

Eine Sprache ist genau dann rekursiv aufzählbar, wenn sie semi-entscheidbar ist.

Beweis (Fortsetzung)

$\Longleftarrow$ Wenn $L = \emptyset$, dann ist L rekursiv aufzählbar.

Sonst, sei M eine Turingmaschine, die χ'_L berechnet und sei $u \in L$ ein Wort. Wir müssen zeigen, dass es eine totale und berechenbare Funktion f gibt, die L aufzählt. Wir konstruieren eine Turingmaschine M', die f berechnet.

Sei n eine Eingabe. Wir interpretieren n als $c(x, y)$.

M' simuliert y Schritte von M bei Eingabe $g(x)$, wobei g die Σ^* aufzählende Funktion ist.

Rekursiv aufzählbar = semi-entscheidbar

Satz

Eine Sprache ist genau dann rekursiv aufzählbar, wenn sie semi-entscheidbar ist.

Beweis (Fortsetzung)

$\Longleftarrow$ Wenn $L = \emptyset$, dann ist L rekursiv aufzählbar.

Sonst, sei M eine Turingmaschine, die χ'_L berechnet und sei $u \in L$ ein Wort. Wir müssen zeigen, dass es eine totale und berechenbare Funktion f gibt, die L aufzählt. Wir konstruieren eine Turingmaschine M', die f berechnet.

Sei n eine Eingabe. Wir interpretieren n als $c(x, y)$.

M' simuliert y Schritte von M bei Eingabe $g(x)$, wobei g die Σ^* aufzählende Funktion ist.

Wenn M nach y Schritten $g(x)$ akzeptiert, dann akzeptiert M' mit Ausgabe $g(x)$. Sonst, akzeptiert M' mit Ausgabe u.

Rekursiv aufzählbar = semi-entscheidbar

Beweis (Fortsetzung)

M' berechnet die Funktion

$$f(n) = \begin{cases} w & \text{falls } w = g(\mathit{left}(n)) \text{ und } M \text{ akzeptiert } w \text{ in } \mathit{right}(n) \text{ Schritten} \\ u & \text{sonst} \end{cases}$$

Rekursiv aufzählbar = semi-entscheidbar

Beweis (Fortsetzung)

M' berechnet die Funktion

$$f(n) = \begin{cases} w & \text{falls } w = g(\mathit{left}(n)) \text{ und } M \text{ akzeptiert } w \text{ in } \mathit{right}(n) \text{ Schritten} \\ u & \text{sonst} \end{cases}$$

f zählt L auf, da für jedes Wort w ein x existiert mit $g(x) = w$ und ein y existiert, sodass M mit Eingabe w nach y Schritten akzeptiert. $\square$

Zusammenfassung: Äquivalente Eigenschaften

Die folgenden Eigenschaften sind äquivalent:

Zusammenfassung: Äquivalente Eigenschaften

Die folgenden Eigenschaften sind äquivalent:

- L ist vom Typ 0.

Zusammenfassung: Äquivalente Eigenschaften

Die folgenden Eigenschaften sind äquivalent:

- L ist vom Typ 0.
- L ist semi-entscheidbar.

Zusammenfassung: Äquivalente Eigenschaften

Die folgenden Eigenschaften sind äquivalent:

- L ist vom Typ 0.
- L ist semi-entscheidbar.
- L ist rekursiv aufzählbar.

Zusammenfassung: Äquivalente Eigenschaften

Die folgenden Eigenschaften sind äquivalent:

- L ist vom Typ 0.
- L ist semi-entscheidbar.
- L ist rekursiv aufzählbar.
- Es gibt eine Turingmaschine M, die L akzeptiert (d.h. $L(M) = L$).

Zusammenfassung: Äquivalente Eigenschaften

Die folgenden Eigenschaften sind äquivalent:

- L ist vom Typ 0.
- L ist semi-entscheidbar.
- L ist rekursiv aufzählbar.
- Es gibt eine Turingmaschine M, die L akzeptiert (d.h. $L(M) = L$).
- χ'_L ist Turing-, WHILE- bzw. GOTO-berechenbar.

Zusammenfassung: Äquivalente Eigenschaften

Die folgenden Eigenschaften sind äquivalent:

- L ist vom Typ 0.
- L ist semi-entscheidbar.
- L ist rekursiv aufzählbar.
- Es gibt eine Turingmaschine M, die L akzeptiert (d.h. $L(M) = L$).
- χ'_L ist Turing-, WHILE- bzw. GOTO-berechenbar.
- Es gibt berechenbare Funktionen, die L als Wertebereich haben (nämlich die L aufzählende Funktion).

Zusammenfassung: Äquivalente Eigenschaften

Die folgenden Eigenschaften sind äquivalent:

- L ist vom Typ 0.
- L ist semi-entscheidbar.
- L ist rekursiv aufzählbar.
- Es gibt eine Turingmaschine M, die L akzeptiert (d.h. $L(M) = L$).
- χ'_L ist Turing-, WHILE- bzw. GOTO-berechenbar.
- Es gibt berechenbare Funktionen, die L als Wertebereich haben (nämlich die L aufzählende Funktion).
- Es gibt berechenbare Funktionen, die L als Definitionsbereich haben (nämlich χ'_L).

Rekursiv aufzählbar $\neq$ abzählbar

Eine Sprache L ist abzählbar, wenn es eine totale Funktion $f : \mathbb{N} \to L$ gibt, sodass $\bigcup_{i \in \mathbb{N}} f(i) = L$.

Beachte: Abzählbarkeit fordert nicht, dass f berechenbar ist.

Gödelisierung von Turingmaschinen

Ziel: Stelle Turingmaschinenbeschreibung als natürliche Zahl in Binärzahldarstellung dar.

Andere Turingmaschinen können die Beschreibung als Eingabe erhalten, erzeugen usw.

Gödelisierung von Turingmaschinen

Sei $(Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ eine DTM mit $\Sigma = \{0, 1\}$ und

- $\Gamma = \{a_0, \ldots, a_k\}$, wobei $a_0 = \square$, $a_1 = \#$, $a_2 = 0$, $a_3 = 1$
- $Z = \{z_0, \ldots, z_n\}$
- $E = \{z_n\}$

Gödelisierung von Turingmaschinen

Sei $(Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ eine DTM mit $\Sigma = \{0, 1\}$ und

- $\Gamma = \{a_0, \ldots, a_k\}$, wobei $a_0 = \square$, $a_1 = \#$, $a_2 = 0$, $a_3 = 1$
- $Z = \{z_0, \ldots, z_n\}$
- $E = \{z_n\}$

Für $\delta(z_p, a_i) = (z_q, a_j, D)$ erzeuge Wort über Alphabet $\{0, 1, \#\}$:

$$\#\#bin(p)\#bin(i)\#bin(q)\#bin(j)\#bin(D_m)$$

mit $D_m = 0$, falls $D = L$, $D_m = 1$, falls $D = R$, $D_m = 2$, falls $D = N$

Gödelisierung von Turingmaschinen

Sei $(Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ eine DTM mit $\Sigma = \{0, 1\}$ und

- $\Gamma = \{a_0, \ldots, a_k\}$, wobei $a_0 = \square$, $a_1 = \#$, $a_2 = 0$, $a_3 = 1$
- $Z = \{z_0, \ldots, z_n\}$
- $E = \{z_n\}$

Für $\delta(z_p, a_i) = (z_q, a_j, D)$ erzeuge Wort über Alphabet $\{0, 1, \#\}$:

$$\#\#bin(p)\#bin(i)\#bin(q)\#bin(j)\#bin(D_m)$$

mit $D_m = 0$, falls $D = L$, $D_m = 1$, falls $D = R$, $D_m = 2$, falls $D = N$

Für δ schreibe alle δ-Wörter hintereinander.

Gödelisierung von Turingmaschinen

Sei $(Z, \Sigma, \Gamma, \delta, z_0, \Box, E)$ eine DTM mit $\Sigma = \{0, 1\}$ und

- $\Gamma = \{a_0, \ldots, a_k\}$, wobei $a_0 = \Box$, $a_1 = \#$, $a_2 = 0$, $a_3 = 1$
- $Z = \{z_0, \ldots, z_n\}$
- $E = \{z_n\}$

Für $\delta(z_p, a_i) = (z_q, a_j, D)$ erzeuge Wort über Alphabet $\{0, 1, \#\}$:

$$\#\#bin(p)\#bin(i)\#bin(q)\#bin(j)\#bin(D_m)$$

mit $D_m = 0$, falls $D = L$, $D_m = 1$, falls $D = R$, $D_m = 2$, falls $D = N$

Für δ schreibe alle δ-Wörter hintereinander.

Schließlich: Kodiere Alphabet $\{0, 1, \#\}$ durch $\{0 \mapsto 00, 1 \mapsto 01, \# \mapsto 11\}$.
Wende dies auf die Kodierung von δ an.

Gödelisierung von Turingmaschinen

Sei $(Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ eine DTM mit $\Sigma = \{0, 1\}$ und

- $\Gamma = \{a_0, \ldots, a_k\}$, wobei $a_0 = \square$, $a_1 = \#$, $a_2 = 0$, $a_3 = 1$
- $Z = \{z_0, \ldots, z_n\}$
- $E = \{z_n\}$

Für $\delta(z_p, a_i) = (z_q, a_j, D)$ erzeuge Wort über Alphabet $\{0, 1, \#\}$:

$$\#\#bin(p)\#bin(i)\#bin(q)\#bin(j)\#bin(D_m)$$

mit $D_m = 0$, falls $D = L$, $D_m = 1$, falls $D = R$, $D_m = 2$, falls $D = N$

Für δ schreibe alle δ-Wörter hintereinander.

Schließlich: Kodiere Alphabet $\{0, 1, \#\}$ durch $\{0 \mapsto 00, 1 \mapsto 01, \# \mapsto 11\}$.
Wende dies auf die Kodierung von δ an.

Wir bezeichnen mit w_M die so kodierte DTM M.

Gödelisierung von Turingmaschinen

- Nicht jedes Wort über $\{0, 1\}$ entspricht der Kodierung einer DTM.
- Sei $\widehat{M}$ eine beliebige aber feste DTM.
- Definiere für jedes $w \in \{0, 1\}^*$ die zugehörige DTM M_w:

$$M_w := \begin{cases} M & \text{falls } w = w_M \\ \widehat{M} & \text{sonst} \end{cases}$$

Spezielles Halteproblem

Definition

Das spezielle Halteproblem ist die Sprache

$$K := \{w \in \{0,1\}^* \mid M_w \text{ hält für Eingabe } w\}$$

Unentscheidbarkeit des speziellen Halteproblems

Satz

Das spezielle Halteproblem ist nicht entscheidbar (und damit unentscheidbar).

Unentscheidbarkeit des speziellen Halteproblems

Satz

Das spezielle Halteproblem ist nicht entscheidbar (und damit unentscheidbar).

Beweis Durch Widerspruch. Wir nehmen an, dass K entscheidbar ist.

Unentscheidbarkeit des speziellen Halteproblems

Satz

Das spezielle Halteproblem ist nicht entscheidbar (und damit unentscheidbar).

Beweis Durch Widerspruch. Wir nehmen an, dass K entscheidbar ist.

Dann ist χ_K berechenbar, und es gibt eine DTM M, die χ_K berechnet.

Unentscheidbarkeit des speziellen Halteproblems

Satz

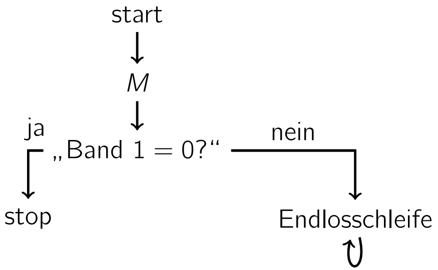
Das spezielle Halteproblem ist nicht entscheidbar (und damit unentscheidbar).

Beweis Durch Widerspruch. Wir nehmen an, dass K entscheidbar ist.

Dann ist χ_K berechenbar, und es gibt eine DTM M, die χ_K berechnet.

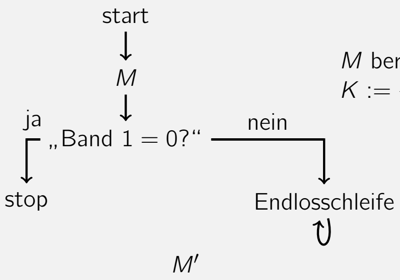
Konstruiere M' wie folgt:

1. M' lässt M ablaufen.
2. Wenn M mit 0 auf dem Band endet, dann akzeptiert M'.
3. Wenn M mit 1 auf dem Band endet, dann läuft M' in eine Endlosschleife.



Unentscheidbarkeit des speziellen Halteproblems

Zur Erinnerung:

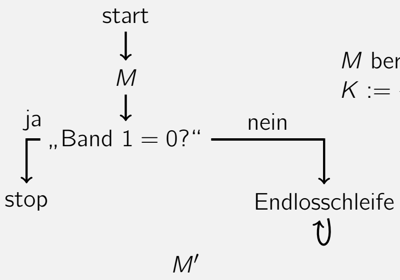


M berechnet χ_K, wobei
$K := \{w \in \{0,1\}^* \mid M_w \text{ hält für Eingabe } w\}$

Beweis (Fortsetzung) Betrachte nun M' auf der Eingabe $w_{M'}$:

Unentscheidbarkeit des speziellen Halteproblems

Zur Erinnerung:



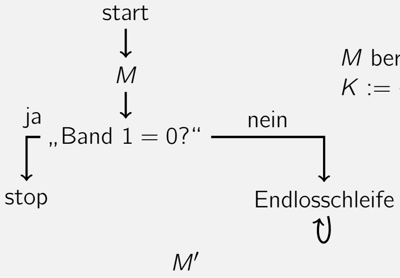
M berechnet χ_K, wobei
$K := \{w \in \{0,1\}^* \mid M_w \text{ hält für Eingabe } w\}$

Beweis (Fortsetzung) Betrachte nun M' auf der Eingabe $w_{M'}$:

M' hält für Eingabe $w_{M'}$

Unentscheidbarkeit des speziellen Halteproblems

Zur Erinnerung:



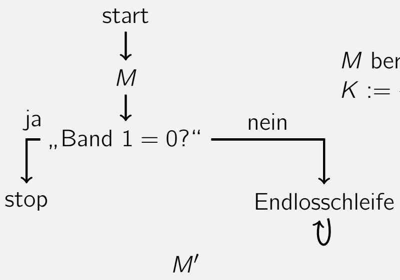
M berechnet χ_K, wobei
$K := \{w \in \{0,1\}^* \mid M_w \text{ hält für Eingabe } w\}$

Beweis (Fortsetzung) Betrachte nun M' auf der Eingabe $w_{M'}$:

M' hält für Eingabe $w_{M'}$
g.d.w. M angesetzt auf $w_{M'}$ gibt 0 aus

Unentscheidbarkeit des speziellen Halteproblems

Zur Erinnerung:



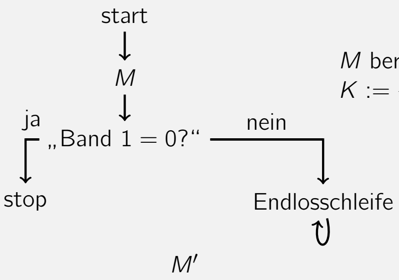
M berechnet χ_K, wobei
$K := \{w \in \{0,1\}^* \mid M_w \text{ hält für Eingabe } w\}$

Beweis (Fortsetzung) Betrachte nun M' auf der Eingabe $w_{M'}$:

$$
\begin{aligned}
&M' \text{ hält für Eingabe } w_{M'}\\
\text{g.d.w. } &M \text{ angesetzt auf } w_{M'} \text{ gibt } 0 \text{ aus}\\
\text{g.d.w. } &\chi_K(w_{M'}) = 0
\end{aligned}
$$

Unentscheidbarkeit des speziellen Halteproblems

Zur Erinnerung:



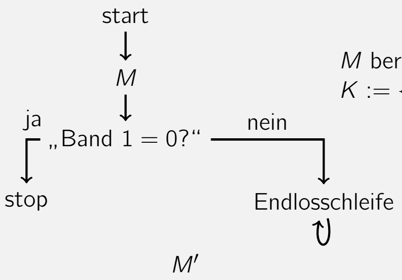
M berechnet χ_K, wobei
$K := \{w \in \{0,1\}^* \mid M_w \text{ hält für Eingabe } w\}$

Beweis (Fortsetzung) Betrachte nun M' auf der Eingabe $w_{M'}$:

$$\begin{aligned} & M' \text{ hält für Eingabe } w_{M'} \\ \text{g.d.w. } & M \text{ angesetzt auf } w_{M'} \text{ gibt } 0 \text{ aus} \\ \text{g.d.w. } & \chi_K(w_{M'}) = 0 \\ \text{g.d.w. } & w_{M'} \notin K \end{aligned}$$

Unentscheidbarkeit des speziellen Halteproblems

Zur Erinnerung:



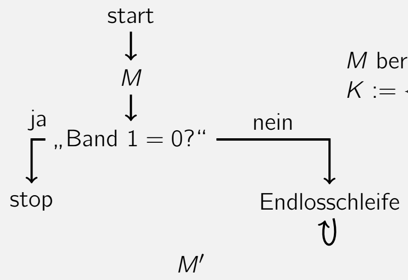
M berechnet χ_K, wobei
$K := \{w \in \{0,1\}^* \mid M_w \text{ hält für Eingabe } w\}$

Beweis (Fortsetzung) Betrachte nun M' auf der Eingabe $w_{M'}$:

$$\begin{array}{ll} & M' \text{ hält für Eingabe } w_{M'} \\ \text{g.d.w.} & M \text{ angesetzt auf } w_{M'} \text{ gibt 0 aus} \\ \text{g.d.w.} & \chi_K(w_{M'}) = 0 \\ \text{g.d.w.} & w_{M'} \notin K \\ \text{g.d.w.} & M' \text{ hält nicht für Eingabe } w_{M'} \end{array}$$

Unentscheidbarkeit des speziellen Halteproblems

Zur Erinnerung:



M berechnet χ_K, wobei
$K := \{w \in \{0,1\}^* \mid M_w \text{ hält für Eingabe } w\}$

Beweis (Fortsetzung) Betrachte nun M' auf der Eingabe $w_{M'}$:

M' hält für Eingabe $w_{M'}$
g.d.w. M angesetzt auf $w_{M'}$ gibt 0 aus
g.d.w. $\chi_K(w_{M'}) = 0$
g.d.w. $w_{M'} \notin K$
g.d.w. M' hält nicht für Eingabe $w_{M'}$

Unentscheidbarkeit des speziellen Halteproblems

Beweis (Fortsetzung) Es gilt also:
M' hält für Eingabe $w_{M'}$ g.d.w. M' hält nicht für Eingabe $w_{M'}$.
Widerspruch. K ist daher nicht entscheidbar, sondern unentscheidbar. $\square$

Unentscheidbarkeit des speziellen Halteproblems

Beweis (Fortsetzung) Es gilt also:
M' hält für Eingabe $w_{M'}$ g.d.w. M' hält nicht für Eingabe $w_{M'}$.
Widerspruch. K ist daher nicht entscheidbar, sondern unentscheidbar. $\square$

Der Beweis, dass K unentscheidbar ist, verwendet ein Diagonalisierungsargument, das wir gleich genauer erläutern.

Unentscheidbarkeit des speziellen Halteproblems

Alternativer Beweis

	w_1	w_2	w_3	...
M_{w_1}	ja	nein	ja	...
M_{w_2}	nein	nein	ja	...
M_{w_3}	ja	nein	ja	...
...	...	...	...	...

Unentscheidbarkeit des speziellen Halteproblems

Alternativer Beweis

	w_1	w_2	w_3	...
M_{w_1}	ja	nein	ja	...
M_{w_2}	nein	nein	ja	...
M_{w_3}	ja	nein	ja	...
...	...	...	...	...
M_{w_D}				

Eintrag in Zeile i und Spalte j: ja, wenn M_{w_i} bei Eingabe w_j akzeptiert, nein sonst.

Unentscheidbarkeit des speziellen Halteproblems

Alternativer Beweis

	w_1	w_2	w_3	$\dots$
M_{w_1}	ja	nein	ja	$\dots$
M_{w_2}	nein	nein	ja	$\dots$
M_{w_3}	ja	nein	ja	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
M_{w_D}	nein			

Eintrag in Zeile i und Spalte j: ja, wenn M_{w_i} bei Eingabe w_j akzeptiert, nein sonst.

Sei $D = \overline{K} = \{w_i \mid M_{w_i}$ hält nicht für Eingabe $w_i\}$.

Unentscheidbarkeit des speziellen Halteproblems

Alternativer Beweis

	w_1	w_2	w_3	...
M_{w_1}	ja	nein	ja	...
M_{w_2}	nein	nein	ja	...
M_{w_3}	ja	nein	ja	...
...	...	...	...	...
M_{w_D}	nein	ja		

Eintrag in Zeile i und Spalte j: ja, wenn M_{w_i} bei Eingabe w_j akzeptiert, nein sonst.

Sei $D = \overline{K} = \{w_i \mid M_{w_i} \text{ hält nicht für Eingabe } w_i\}$.

Wir nehmen an, dass w_D die Beschreibung von DTM M_{w_D} ist, die χ_D berechnet.

Unentscheidbarkeit des speziellen Halteproblems

Alternativer Beweis

	w_1	w_2	w_3	...
M_{w_1}	ja	nein	ja	...
M_{w_2}	nein	nein	ja	...
M_{w_3}	ja	nein	ja	...
...	...	...	...	...
M_{w_D}	nein	ja	nein	

Eintrag in Zeile i und Spalte j: ja, wenn M_{w_i} bei Eingabe w_j akzeptiert, nein sonst.

Sei $D = \overline{K} = \{w_i \mid M_{w_i}$ hält nicht für Eingabe $w_i\}$.

Wir nehmen an, dass w_D die Beschreibung von DTM M_{w_D} ist, die χ_D berechnet.

Der Eintrag in Zeile M_{w_D} und Spalte w_D ist ja g.d.w. der Eintrag nein ist.
Widerspruch. $D = \overline{K}$ und K sind daher nicht entscheidbar. $\square$

Alternativer Beweis

	w_1	w_2	w_3	$\dots$	w_D
M_{w_1}	ja	nein	ja	$\dots$	$\dots$
M_{w_2}	nein	nein	ja	$\dots$	$\dots$
M_{w_3}	ja	nein	ja	$\dots$	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
M_{w_D}	nein	ja	nein	$\dots$	

Eintrag in Zeile i und Spalte j: ja, wenn M_{w_i} bei Eingabe w_j akzeptiert, nein sonst.

Sei $D = \overline{K} = \{w_i \mid M_{w_i}$ hält nicht für Eingabe $w_i\}$.

Wir nehmen an, dass w_D die Beschreibung von DTM M_{w_D} ist, die χ_D berechnet.

Der Eintrag in Zeile M_{w_D} und Spalte w_D ist ja g.d.w. der Eintrag nein ist.
Widerspruch. $D = \overline{K}$ und K sind daher nicht entscheidbar. $\square$

Unentscheidbarkeit des speziellen Halteproblems

Alternativer Beweis

	w_1	w_2	w_3	$\ldots$	w_D
M_{w_1}	ja	nein	ja	$\ldots$	$\ldots$
M_{w_2}	nein	nein	ja	$\ldots$	$\ldots$
M_{w_3}	ja	nein	ja	$\ldots$	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
M_{w_D}	nein	ja	nein	$\ldots$	**??**

Eintrag in Zeile i und Spalte j: ja, wenn M_{w_i} bei Eingabe w_j akzeptiert, nein sonst.

Sei $D = \overline{K} = \{w_i \mid M_{w_i} \text{ hält nicht für Eingabe } w_i\}$.

Wir nehmen an, dass w_D die Beschreibung von DTM M_{w_D} ist, die χ_D berechnet.

Der Eintrag in Zeile M_{w_D} und Spalte w_D ist ja g.d.w. der Eintrag nein ist.
Widerspruch. $D = \overline{K}$ und K sind daher nicht entscheidbar. $\square$

Semi-Entscheidbarkeit des speziellen Halteproblems

K ist semi-entscheidbar:

> **wiederhole für** $i = 0, 1, \ldots$:
> Simuliere i Schritte von M_w auf Eingabe w.
> Wenn M_w akzeptiert, dann akzeptiere und gib 1 aus.

Da K unentscheidbar aber semi-entscheidbar ist, heißt $\overline{K}$ ist nicht semi-entscheidbar.