3 Berechenbarkeitstheorie

Halteproblem

Eingabe: Programm P in Ihrer Lieblingsprogrammiersprache
Eingabe w für das Programm

Frage: Terminiert das Programm P nach endlich vielen Schritten auf Eingabe w?

Halteproblem

Eingabe: Programm P in Ihrer Lieblingsprogrammiersprache
Eingabe w für das Programm

Frage: Terminiert das Programm P nach endlich vielen Schritten auf Eingabe w?

Theorem

Das Halteproblem ist **nicht entscheidbar**, d. h. es gibt keinen Algorithmus, der dieses Problem in endlicher Zeit löst.

Ein Algorithmus für das Halteproblem hätte überraschende Konsequenzen:

```
void main()
  int n = 4;
  while (true)
    boolean foundPrimes = false;
    for (int p = 2; p < n; p++)
      if (prime(p) && prime(n-p)) foundPrimes = true;
    if (!foundPrimes) return;
    n = n + 2;
```

Ein Algorithmus für das Halteproblem hätte überraschende Konsequenzen:

```
void main()
  int n = 4;
  while (true)
    boolean foundPrimes = false;
    for (int p = 2; p < n; p++)
      if (prime(p) && prime(n-p)) foundPrimes = true;
    if (!foundPrimes) return;
    n = n + 2;
```

Beobachtung 3.1

Das obige Programm terminiert genau dann, wenn es eine **gerade Zahl größer als zwei gibt, die sich nicht als Summe zweier Primzahlen** schreiben lässt.

Ein Algorithmus für das Halteproblem hätte überraschende Konsequenzen:

```
void main()
  int n = 4;
  while (true)
    boolean foundPrimes = false;
    for (int p = 2; p < n; p++)
      if (prime(p) && prime(n-p)) foundPrimes = true;
    if (!foundPrimes) return;
    n = n + 2;
```

Beobachtung 3.1

Das obige Programm terminiert genau dann, wenn es eine **gerade Zahl größer als zwei gibt, die sich nicht als Summe zweier Primzahlen** schreiben lässt.

$\Rightarrow$ Mithilfe eines Algorithmus für das Halteproblem könnte die **Goldbachsche Vermutung** bewiesen oder widerlegt werden.

3.1 Entwurf einer universellen Turingmaschine

Universelle Turingmaschine (Idee)

Eine **universelle Turingmaschine U** erhält als Eingabe die Codierung einer Turingmaschine M und eine Eingabe w für M. Sie **simuliert das Verhalten von M auf w**.

3.1 Entwurf einer universellen Turingmaschine

Universelle Turingmaschine (Idee)

Eine **universelle Turingmaschine U** erhält als Eingabe die Codierung einer Turingmaschine M und eine Eingabe w für M. Sie **simuliert das Verhalten von M auf w**.

Annahmen: Eingabealphabet $\Sigma = \{0, 1\}$, Bandalphabet $\Gamma = \{0, 1, \square\}$.

Für eine Turingmaschine M sei $\langle M \rangle \in \{0, 1\}^*$ ihre Codierung.

3.1 Entwurf einer universellen Turingmaschine

Universelle Turingmaschine (Idee)

Eine **universelle Turingmaschine U** erhält als Eingabe die Codierung einer Turingmaschine M und eine Eingabe w für M. Sie **simuliert das Verhalten von M auf w**.

Annahmen: Eingabealphabet $\Sigma = \{0, 1\}$, Bandalphabet $\Gamma = \{0, 1, \square\}$.

Für eine Turingmaschine M sei $\langle M \rangle \in \{0, 1\}^*$ ihre Codierung.

Universelle Turingmaschine (formal)

Eingabe für U ist $\langle M \rangle w \in \{0, 1\}^*$ für eine TM M und ein $w \in \{0, 1\}^*$.

3.1 Entwurf einer universellen Turingmaschine

Universelle Turingmaschine (Idee)

Eine **universelle Turingmaschine U** erhält als Eingabe die Codierung einer Turingmaschine M und eine Eingabe w für M. Sie **simuliert das Verhalten von M auf w**.

Annahmen: Eingabealphabet $\Sigma = \{0, 1\}$, Bandalphabet $\Gamma = \{0, 1, \square\}$.

Für eine Turingmaschine M sei $\langle M \rangle \in \{0, 1\}^*$ ihre Codierung.

Universelle Turingmaschine (formal)

Eingabe für U ist $\langle M \rangle w \in \{0, 1\}^*$ für eine TM M und ein $w \in \{0, 1\}^*$.

U simuliert das Verhalten von M auf Eingabe w, d. h.:

3.1 Entwurf einer universellen Turingmaschine

Universelle Turingmaschine (Idee)

Eine **universelle Turingmaschine U** erhält als Eingabe die Codierung einer Turingmaschine M und eine Eingabe w für M. Sie **simuliert das Verhalten von M auf w**.

Annahmen: Eingabealphabet $\Sigma = \{0, 1\}$, Bandalphabet $\Gamma = \{0, 1, \square\}$.

Für eine Turingmaschine M sei $\langle M \rangle \in \{0, 1\}^*$ ihre Codierung.

Universelle Turingmaschine (formal)

Eingabe für U ist $\langle M \rangle w \in \{0, 1\}^*$ für eine TM M und ein $w \in \{0, 1\}^*$.

U simuliert das Verhalten von M auf Eingabe w, d. h.:

U terminiert auf $\langle M \rangle w \iff M$ terminiert auf w

3.1 Entwurf einer universellen Turingmaschine

Universelle Turingmaschine (Idee)

Eine **universelle Turingmaschine U** erhält als Eingabe die Codierung einer Turingmaschine M und eine Eingabe w für M. Sie **simuliert das Verhalten von M auf w**.

Annahmen: Eingabealphabet $\Sigma = \{0, 1\}$, Bandalphabet $\Gamma = \{0, 1, \square\}$.

Für eine Turingmaschine M sei $\langle M \rangle \in \{0, 1\}^*$ ihre Codierung.

Universelle Turingmaschine (formal)

Eingabe für U ist $\langle M \rangle w \in \{0, 1\}^*$ für eine TM M und ein $w \in \{0, 1\}^*$.

U simuliert das Verhalten von M auf Eingabe w, d. h.:

U terminiert auf $\langle M \rangle w \iff M$ terminiert auf w

U akzeptiert (verwirft) $\langle M \rangle w \iff M$ akzeptiert (verwirft) w

Definition 3.2 (Gödelnummern)

Eine injektive Abbildung der Menge aller Turingmaschinen in die Menge $\{0,1\}^*$ heißt **Gödelnummerierung**.

Definition 3.2 (Gödelnummern)

Eine injektive Abbildung der Menge aller Turingmaschinen in die Menge $\{0,1\}^*$ heißt **Gödelnummerierung**. Für eine feste Gödelnummerierung bezeichnen wir mit $\langle M \rangle \in \{0,1\}^*$ die **Gödelnummer** der Turingmaschine M, also die Zeichenkette, auf die M gemäß der Gödelnummerierung abgebildet wird.

Definition 3.2 (Gödelnummern)

Eine injektive Abbildung der Menge aller Turingmaschinen in die Menge $\{0,1\}^*$ heißt **Gödelnummerierung**. Für eine feste Gödelnummerierung bezeichnen wir mit $\langle M \rangle \in \{0,1\}^*$ die **Gödelnummer** der Turingmaschine M, also die Zeichenkette, auf die M gemäß der Gödelnummerierung abgebildet wird.

Eine Gödelnummerierung heißt **präfixfrei**, wenn kein echtes Präfix (Anfangsstück) einer Gödelnummer $\langle M \rangle$ selbst eine gültige Gödelnummer ist.

Definition 3.2 (Gödelnummern)

Eine injektive Abbildung der Menge aller Turingmaschinen in die Menge $\{0,1\}^*$ heißt **Gödelnummerierung**. Für eine feste Gödelnummerierung bezeichnen wir mit $\langle M \rangle \in \{0,1\}^*$ die **Gödelnummer** der Turingmaschine M, also die Zeichenkette, auf die M gemäß der Gödelnummerierung abgebildet wird.

Eine Gödelnummerierung heißt **präfixfrei**, wenn kein echtes Präfix (Anfangsstück) einer Gödelnummer $\langle M \rangle$ selbst eine gültige Gödelnummer ist.

Warum präfixfrei? Damit eine Zeichenkette $\langle M \rangle w \in \{0,1\}^*$ **eindeutig** in die Komponenten $\langle M \rangle \in \{0,1\}^*$ und $w \in \{0,1\}^*$ **zerlegt werden kann**.

Eine konkrete präfixfreie Gödelnummerierung:

Es sei eine Turingmaschine $M = (Q, \Sigma, \Gamma, \square, q_1, q_2, \delta)$ wie folgt:

- Es sei $Q = \{q_1, q_2, \ldots, q_t\}$ für ein $t \geq 2$.
- Es sei $\Sigma = \{X_1, X_2\}$ und $\Gamma = \Sigma \cup \{X_3\}$ für $X_1 = 0$, $X_2 = 1$ und $X_3 = \square$.
- Es sei q_1 der Startzustand und q_2 der Endzustand.

Eine konkrete präfixfreie Gödelnummerierung:

Es sei eine Turingmaschine $M = (Q, \Sigma, \Gamma, \square, q_1, q_2, \delta)$ wie folgt:

- Es sei $Q = \{q_1, q_2, \ldots, q_t\}$ für ein $t \geq 2$.
- Es sei $\Sigma = \{X_1, X_2\}$ und $\Gamma = \Sigma \cup \{X_3\}$ für $X_1 = 0$, $X_2 = 1$ und $X_3 = \square$.
- Es sei q_1 der Startzustand und q_2 der Endzustand.

Es sei $D_1 = L$, $D_2 = N$ und $D_3 = R$.
Einen Übergang $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ codieren wir als $0^i 1 0^j 1 0^k 1 0^\ell 1 0^m$.

Eine konkrete präfixfreie Gödelnummerierung:

Es sei eine Turingmaschine $M = (Q, \Sigma, \Gamma, \square, q_1, q_2, \delta)$ wie folgt:

- Es sei $Q = \{q_1, q_2, \ldots, q_t\}$ für ein $t \geq 2$.
- Es sei $\Sigma = \{X_1, X_2\}$ und $\Gamma = \Sigma \cup \{X_3\}$ für $X_1 = 0$, $X_2 = 1$ und $X_3 = \square$.
- Es sei q_1 der Startzustand und q_2 der Endzustand.

Es sei $D_1 = L$, $D_2 = N$ und $D_3 = R$.
Einen Übergang $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ codieren wir als $0^i10^j10^k10^\ell10^m$.
Die Zustandsüberführungsfunktion beschreibt $3(t-1)$ viele Übergänge, deren Codierungen wir mit $\mathrm{code}(1), \ldots, \mathrm{code}(3(t-1))$ bezeichnen.

Eine konkrete präfixfreie Gödelnummerierung:

Es sei eine Turingmaschine $M = (Q, \Sigma, \Gamma, \square, q_1, q_2, \delta)$ wie folgt:

- Es sei $Q = \{q_1, q_2, \ldots, q_t\}$ für ein $t \geq 2$.
- Es sei $\Sigma = \{X_1, X_2\}$ und $\Gamma = \Sigma \cup \{X_3\}$ für $X_1 = 0$, $X_2 = 1$ und $X_3 = \square$.
- Es sei q_1 der Startzustand und q_2 der Endzustand.

Es sei $D_1 = L$, $D_2 = N$ und $D_3 = R$.
Einen Übergang $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ codieren wir als $0^i 1 0^j 1 0^k 1 0^\ell 1 0^m$.
Die Zustandsüberführungsfunktion beschreibt $3(t-1)$ viele Übergänge, deren Codierungen wir mit $\mathrm{code}(1), \ldots, \mathrm{code}(3(t-1))$ bezeichnen.

Die Gödelnummer von M lautet

$$\langle M \rangle = 11\,\mathrm{code}(1)\,11\,\mathrm{code}(2)\,11\ \ldots\ 11\,\mathrm{code}(3(t-1))\,111.$$

Eine konkrete präfixfreie Gödelnummerierung:

Es sei eine Turingmaschine $M = (Q, \Sigma, \Gamma, \square, q_1, q_2, \delta)$ wie folgt:

- Es sei $Q = \{q_1, q_2, \ldots, q_t\}$ für ein $t \geq 2$.
- Es sei $\Sigma = \{X_1, X_2\}$ und $\Gamma = \Sigma \cup \{X_3\}$ für $X_1 = 0$, $X_2 = 1$ und $X_3 = \square$.
- Es sei q_1 der Startzustand und q_2 der Endzustand.

Es sei $D_1 = L$, $D_2 = N$ und $D_3 = R$.
Einen Übergang $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ codieren wir als $0^i 1 0^j 1 0^k 1 0^\ell 1 0^m$.
Die Zustandsüberführungsfunktion beschreibt $3(t-1)$ viele Übergänge, deren Codierungen wir mit $\mathrm{code}(1), \ldots, \mathrm{code}(3(t-1))$ bezeichnen.

Die Gödelnummer von M lautet

$$\langle M \rangle = 11\,\mathrm{code}(1)\,11\,\mathrm{code}(2)\,11\ \ldots\ 11\,\mathrm{code}(3(t-1))\,111.$$

111 am Ende stellt die **Präfixfreiheit** sicher.

Theorem 3.3

Es existiert eine universelle Turingmaschine U, die auf jeder Eingabe der Form $\langle M \rangle w \in \{0,1\}^*$ das Verhalten der Turingmaschine M auf der Eingabe $w \in \{0,1\}^*$ simuliert.

Theorem 3.3

Es existiert eine universelle Turingmaschine U, die auf jeder Eingabe der Form $\langle M \rangle w \in \{0,1\}^*$ das Verhalten der Turingmaschine M auf der Eingabe $w \in \{0,1\}^*$ simuliert.

Die **Rechenzeit** von U auf der Eingabe $\langle M \rangle w$ ist nur **um einen konstanten Faktor** (der nur von M, nicht aber von w abhängt) größer als die Rechenzeit von M auf der Eingabe w.

Beweis: Konstruktion von U als 3-Band-Turingmaschine.

Beweis: Konstruktion von U als 3-Band-Turingmaschine.

Initialisierung:

- Zu Beginn: $\langle M \rangle w$ auf Band 1.
- U schreibt $\langle M \rangle$ auf Band 2 und löscht es von Band 1.
- U schreibt auf Band 3 eine Codierung des Startzustandes.
  Wir codieren dabei den Zustand q_i als 0^i.

Beweis: Konstruktion von U als 3-Band-Turingmaschine.

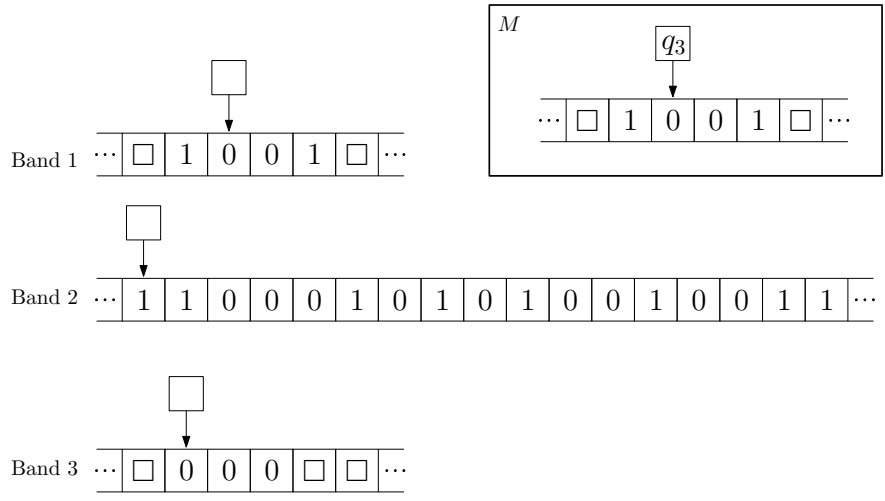
Initialisierung:

- Zu Beginn: $\langle M \rangle w$ auf Band 1.
- U schreibt $\langle M \rangle$ auf Band 2 und löscht es von Band 1.
- U schreibt auf Band 3 eine Codierung des Startzustandes.
  Wir codieren dabei den Zustand q_i als 0^i.

Invariante:

- Band 1 enthält den Bandinhalt von M nach den bereits simulierten Schritten.
  Die Kopfposition auf Band 1 stimmt ebenfalls mit der von M überein.
- Band 2 enthält die Gödelnummer $\langle M \rangle$.
- Band 3 codiert den Zustand von M nach den bereits simulieren Schritten.
  Ist dies q_i, so enthält Band 3 die Zeichenkette 0^i.

3.1 Entwurf einer universellen Turingmaschine



Simulation von M auf w Schritt für Schritt:

- **U sucht auf Band 2 nach einem passenden Übergang:**
  U sucht nach der Zeichenkette 110^i10^j1, wobei q_i der aktuell auf Band 3 codierte Zustand sei und X_j das Zeichen, das sich auf Band 1 an der Kopfposition befindet.

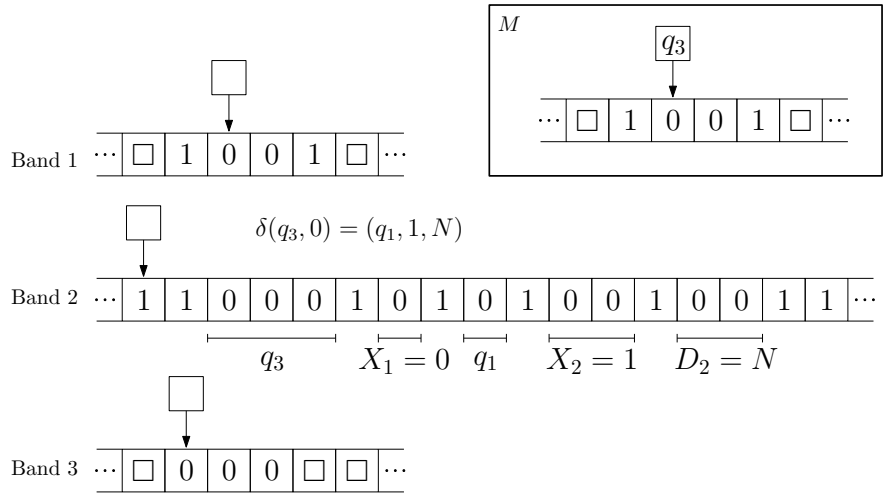
Simulation von M auf w Schritt für Schritt:

- **U sucht auf Band 2 nach einem passenden Übergang:**
  U sucht nach der Zeichenkette 110^i10^j1, wobei q_i der aktuell auf Band 3 codierte Zustand sei und X_j das Zeichen, das sich auf Band 1 an der Kopfposition befindet.
- Findet U eine entsprechende Zeichenkette, so kann sie entsprechend der oben beschriebenen Codierung den **Übergang** $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ **rekonstruieren**.

Simulation von M auf w Schritt für Schritt:

- **U sucht auf Band 2 nach einem passenden Übergang:**
  U sucht nach der Zeichenkette 110^i10^j1, wobei q_i der aktuell auf Band 3 codierte Zustand sei und X_j das Zeichen, das sich auf Band 1 an der Kopfposition befindet.
- Findet U eine entsprechende Zeichenkette, so kann sie entsprechend der oben beschriebenen Codierung den **Übergang** $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ **rekonstruieren**.
- U ersetzt dann das Zeichen an der Kopfposition auf Band 1 durch X_ℓ, bewegt den Kopf von Band 1 gemäß D_m und ersetzt den Inhalt von Band 3 durch 0^k.

3.1 Entwurf einer universellen Turingmaschine



- Für Simulation eines Schrittes von M benötigt U **konstant viele Schritte**.

3.1 Entwurf einer universellen Turingmaschine

- Für Simulation eines Schrittes von M benötigt U **konstant viele Schritte**.
- Bei der Simulation der 3-Band-Turingmaschine durch eine 1-Band-Turingmaschine gemäß Theorem 2.5 handeln wir uns einen **quadratischen Zeitverlust** ein.

- Für Simulation eines Schrittes von M benötigt U **konstant viele Schritte**.
- Bei der Simulation der 3-Band-Turingmaschine durch eine 1-Band-Turingmaschine gemäß Theorem 2.5 handeln wir uns einen **quadratischen Zeitverlust** ein.
- Dieser kann vermieden werden, indem die universelle Turingmaschine von **vornherein als 3-Spur-Turingmaschine** konstruiert wird.
  (Die Details hierzu besprechen wir nicht.) □

Definition: Lexikographische Ordnung

Für zwei Wörter $x = x_1 \dots x_\ell \in \{0,1\}^\ell$ und $y = y_1 \dots y_\ell \in \{0,1\}^\ell$ derselben Länge ℓ heißt x **lexikographisch kleiner** als y, wenn ein Index i existiert, für den $x_1 \dots x_{i-1} = y_1 \dots y_{i-1}$ sowie $x_i = 0$ und $y_i = 1$ gilt.

Definition: Lexikographische Ordnung

Für zwei Wörter $x = x_1 \dots x_\ell \in \{0,1\}^\ell$ und $y = y_1 \dots y_\ell \in \{0,1\}^\ell$ derselben Länge ℓ heißt x **lexikographisch kleiner** als y, wenn ein Index i existiert, für den $x_1 \dots x_{i-1} = y_1 \dots y_{i-1}$ sowie $x_i = 0$ und $y_i = 1$ gilt.

Nun können wir die **kanonische Ordnung** auf der Menge $\{0,1\}^*$ definieren. In dieser Ordnung kommt ein Wort $x \in \{0,1\}^*$ vor einem Wort $y \in \{0,1\}^*$, wenn $|x| < |y|$ gilt oder wenn $|x| = |y|$ gilt und x lexikographisch kleiner als y ist.

Definition: Lexikographische Ordnung

Für zwei Wörter $x = x_1 \dots x_\ell \in \{0,1\}^\ell$ und $y = y_1 \dots y_\ell \in \{0,1\}^\ell$ derselben Länge ℓ heißt x **lexikographisch kleiner** als y, wenn ein Index i existiert, für den $x_1 \dots x_{i-1} = y_1 \dots y_{i-1}$ sowie $x_i = 0$ und $y_i = 1$ gilt.

Nun können wir die **kanonische Ordnung** auf der Menge $\{0,1\}^*$ definieren. In dieser Ordnung kommt ein Wort $x \in \{0,1\}^*$ vor einem Wort $y \in \{0,1\}^*$, wenn $|x| < |y|$ gilt oder wenn $|x| = |y|$ gilt und x lexikographisch kleiner als y ist.

Die ersten Wörter in der kanonischen Ordnung von $\{0,1\}^*$ sehen demnach wie folgt aus:

$$\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, \dots$$

Definition: Lexikographische Ordnung

Für zwei Wörter $x = x_1 \dots x_\ell \in \{0,1\}^\ell$ und $y = y_1 \dots y_\ell \in \{0,1\}^\ell$ derselben Länge ℓ heißt x **lexikographisch kleiner** als y, wenn ein Index i existiert, für den $x_1 \dots x_{i-1} = y_1 \dots y_{i-1}$ sowie $x_i = 0$ und $y_i = 1$ gilt.

Nun können wir die **kanonische Ordnung** auf der Menge $\{0,1\}^*$ definieren. In dieser Ordnung kommt ein Wort $x \in \{0,1\}^*$ vor einem Wort $y \in \{0,1\}^*$, wenn $|x| < |y|$ gilt oder wenn $|x| = |y|$ gilt und x lexikographisch kleiner als y ist.

Die ersten Wörter in der kanonischen Ordnung von $\{0,1\}^*$ sehen demnach wie folgt aus:

$$\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, \dots$$

Für $i \in \mathbb{N}$ bezeichnen wir mit w_i das Wort aus $\{0,1\}^*$, das **in der kanonischen Ordnung an der i-ten Stelle** steht. Es gilt also beispielsweise $w_1 = \varepsilon$ und $w_5 = 01$.

3.2 Die Unentscheidbarkeit des Halteproblems

Sei $\mathcal{G} \subseteq \{0,1\}^*$ die **Menge aller Gödelnummern**.

Sei $\mathbf{M_i}$ für $i \in \mathbb{N}$ die Turingmaschine, deren Gödelnummer $\langle M_i \rangle$ in der kanonischen Ordnung der Menge $\mathcal{G}$ **an der i-ten Stelle steht**.

Sei $\mathcal{G} \subseteq \{0,1\}^*$ die **Menge aller Gödelnummern**.

Sei **M_i** für $i \in \mathbb{N}$ die Turingmaschine, deren Gödelnummer $\langle M_i \rangle$ in der kanonischen Ordnung der Menge $\mathcal{G}$ **an der i-ten Stelle steht**.

Beobachtung

Für ein gegebenes $i \in \mathbb{N}$ können w_i und M_i berechnet werden.

Sei $\mathcal{G} \subseteq \{0,1\}^*$ die **Menge aller Gödelnummern**.

Sei **M_i** für $i \in \mathbb{N}$ die Turingmaschine, deren Gödelnummer $\langle M_i \rangle$ in der kanonischen Ordnung der Menge $\mathcal{G}$ **an der i-ten Stelle steht**.

Beobachtung

Für ein gegebenes $i \in \mathbb{N}$ können w_i und M_i berechnet werden. Außerdem ist es möglich, für ein gegebenes Wort $w \in \{0,1\}^*$ den Index i mit $w_i = w$ zu berechnen und für eine gegebene Gödelnummer $\langle M \rangle \in \mathcal{G}$ den Index $i \in \mathbb{N}$ mit $M_i = M$.

Definition 3.4

Die **Diagonalsprache** ist definiert als

$$D = \{ w_i \in \{0,1\}^* \mid M_i \text{ akzeptiert } w_i \text{ nicht} \}.$$

Definition 3.4

Die **Diagonalsprache** ist definiert als

$$D = \{w_i \in \{0,1\}^* \mid M_i \text{ akzeptiert } w_i \text{ nicht}\}.$$

In Zelle (M_j, w_i) steht eine Eins, wenn M_j das Wort w_i akzeptiert, und sonst eine Null.

	w_1	w_2	w_3	w_4	w_5	$\ldots$
M_1	**0**	1	1	0	1	$\ldots$
M_2	1	**1**	0	0	0	$\ldots$
M_3	1	0	**0**	1	1	$\ldots$
M_4	1	0	0	**1**	0	$\ldots$
M_5	0	1	1	1	**1**	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Definition 3.4

Die **Diagonalsprache** ist definiert als

$$D = \{w_i \in \{0,1\}^* \mid M_i \text{ akzeptiert } w_i \text{ nicht}\}.$$

In Zelle (M_j, w_i) steht eine Eins, wenn M_j das Wort w_i akzeptiert, und sonst eine Null.

	w_1	w_2	w_3	w_4	w_5	$\ldots$
M_1	**0**	1	1	0	1	$\ldots$
M_2	1	**1**	0	0	0	$\ldots$
M_3	1	0	**0**	1	1	$\ldots$
M_4	1	0	0	**1**	0	$\ldots$
M_5	0	1	1	1	**1**	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Im Beispiel gilt $w_1, w_3 \in D$, aber $w_2, w_4, w_5 \notin D$.

Theorem 3.5

Die Diagonalsprache D ist nicht entscheidbar.

Beweis: Wir führen einen Widerspruchsbeweis.

Sei M eine Turingmaschine, die D entscheidet.

Theorem 3.5

Die Diagonalsprache D ist nicht entscheidbar.

Beweis: Wir führen einen Widerspruchsbeweis.

Sei M eine Turingmaschine, die D entscheidet.

Es gibt einen Index $i \in \mathbb{N}$ mit $M = M_i$.

Theorem 3.5

Die Diagonalsprache D ist nicht entscheidbar.

Beweis: Wir führen einen Widerspruchsbeweis.

Sei M eine Turingmaschine, die D entscheidet.

Es gibt einen Index $i \in \mathbb{N}$ mit $M = M_i$.

Wie verhält sich $M = M_i$ auf dem Wort w_i?

Theorem 3.5

Die Diagonalsprache D ist nicht entscheidbar.

Beweis: Wir führen einen Widerspruchsbeweis.

Sei M eine Turingmaschine, die D entscheidet.

Es gibt einen Index $i \in \mathbb{N}$ mit $M = M_i$.

Wie verhält sich $M = M_i$ auf dem Wort w_i?

- Gilt $w_i \in D$, so akzeptiert M_i die Eingabe w_i, da M_i die Sprache D entscheidet.

Theorem 3.5

Die Diagonalsprache D ist nicht entscheidbar.

Beweis: Wir führen einen Widerspruchsbeweis.

Sei M eine Turingmaschine, die D entscheidet.

Es gibt einen Index $i \in \mathbb{N}$ mit $M = M_i$.

Wie verhält sich $M = M_i$ auf dem Wort w_i?

- Gilt $w_i \in D$, so akzeptiert M_i die Eingabe w_i, da M_i die Sprache D entscheidet.
  Aus der Definition von D folgt dann aber, dass $w_i \notin D$ gilt.

Theorem 3.5

Die Diagonalsprache D ist nicht entscheidbar.

Beweis: Wir führen einen Widerspruchsbeweis.

Sei M eine Turingmaschine, die D entscheidet.

Es gibt einen Index $i \in \mathbb{N}$ mit $M = M_i$.

Wie verhält sich $M = M_i$ auf dem Wort w_i?

- Gilt $w_i \in D$, so akzeptiert M_i die Eingabe w_i, da M_i die Sprache D entscheidet.
  Aus der Definition von D folgt dann aber, dass $w_i \notin D$ gilt.
- Gilt $w_i \notin D$, so verwirft M_i die Eingabe w_i, da M_i die Sprache D entscheidet.

Theorem 3.5

Die Diagonalsprache D ist nicht entscheidbar.

Beweis: Wir führen einen Widerspruchsbeweis.

Sei M eine Turingmaschine, die D entscheidet.

Es gibt einen Index $i \in \mathbb{N}$ mit $M = M_i$.

Wie verhält sich $M = M_i$ auf dem Wort w_i?

- Gilt $w_i \in D$, so akzeptiert M_i die Eingabe w_i, da M_i die Sprache D entscheidet.
  Aus der Definition von D folgt dann aber, dass $w_i \notin D$ gilt.
- Gilt $w_i \notin D$, so verwirft M_i die Eingabe w_i, da M_i die Sprache D entscheidet.
  Aus der Definition von D folgt dann aber, dass $w_i \in D$ gilt.

Widerspruch! □

Definition 3.6

Das **Halteproblem** ist definiert als

$$H = \{\langle M \rangle w \mid M \text{ hält auf } w\} \subseteq \{0,1\}^*.$$

Definition 3.6

Das **Halteproblem** ist definiert als

$$H = \{\langle M \rangle w \mid M \text{ hält auf } w\} \subseteq \{0,1\}^*.$$

Theorem 3.7

Das Halteproblem H ist nicht entscheidbar.

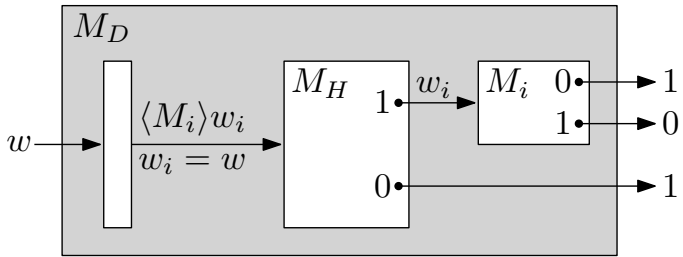
Beweis durch Reduktion: Eine TM M_H, die das Halteproblem entscheidet, kann genutzt werden, um eine TM M_D zu konstruieren, die die Diagonalsprache D entscheidet.

Beweis durch Reduktion: Eine TM M_H, die das Halteproblem entscheidet, kann genutzt werden, um eine TM M_D zu konstruieren, die die Diagonalsprache D entscheidet.

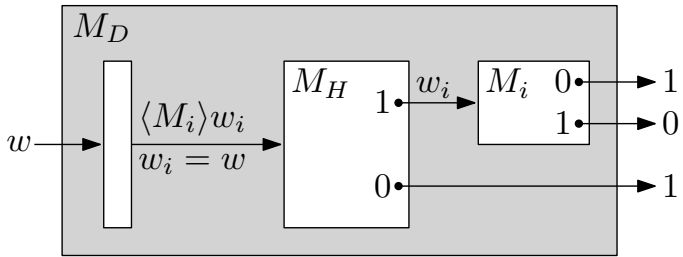
$M_D(w)$
1 Berechne $i \in \mathbb{N}$ mit $w_i = w$. Berechne die Gödelnummer $\langle M_i \rangle$.
2 Simuliere das Verhalten von M_H auf der Eingabe $\langle M_i \rangle w_i$.
3 **if** (M_H akzeptiert $\langle M_i \rangle w_i$ nicht)
4 akzeptiere w;
5 **else**
6 Simuliere das Verhalten von M_i auf der Eingabe w_i.
7 **if** (M_i akzeptiert w_i) verwirf w;
8 **else** akzeptiere w;

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

$$H = \{ \langle M \rangle w \mid M \text{ hält auf } w \} \subseteq \{0,1\}^*$$



$$D = \{ w_i \in \{0,1\}^* \mid M_i \text{ akzeptiert } w_i \text{ nicht} \}$$
$$H = \{ \langle M \rangle w \mid M \text{ hält auf } w \} \subseteq \{0,1\}^*$$



M_D akzeptiert $w = w_i$ genau dann, wenn M_i die Eingabe w_i nicht akzeptiert. □

Definition (Turing-Reduktion)

Eine **Turing-Reduktion** einer Sprache A auf eine Sprache B ist eine Turingmaschine, die die Sprache A mithilfe eines (hypothetischen) Unterprogramms für die Sprache B löst.

Turing-Reduktionen werden auch als **Unterprogrammtechnik** bezeichnet.

3.3 Turing- und Many-One-Reduktionen

Definition (Turing-Reduktion)

Eine **Turing-Reduktion** einer Sprache A auf eine Sprache B ist eine Turingmaschine, die die Sprache A mithilfe eines (hypothetischen) Unterprogramms für die Sprache B löst. Turing-Reduktionen werden auch als **Unterprogrammtechnik** bezeichnet.

Definition 3.8 (Many-One-Reduktion)

Eine **Many-One-Reduktion** einer Sprache $A \subseteq \Sigma_1^*$ auf eine Sprache $B \subseteq \Sigma_2^*$ ist eine **berechenbare** Funktion $f\colon \Sigma_1^* \to \Sigma_2^*$ mit der Eigenschaft, dass

$$x \in A \iff f(x) \in B$$

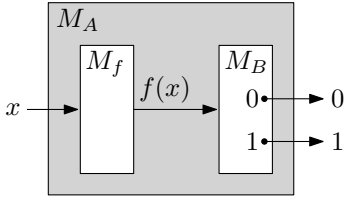
für alle $x \in \Sigma_1^*$ gilt. Existiert eine solche Reduktion, so heißt A auf B **reduzierbar** und wir schreiben $A \leq B$.

Theorem 3.9

Es seien $A \subseteq \Sigma_1^*$ und $B \subseteq \Sigma_2^*$ zwei Sprachen, für die $A \leq B$ gilt. Ist B entscheidbar, so ist auch A entscheidbar. Ist A nicht entscheidbar, so ist auch B nicht entscheidbar.

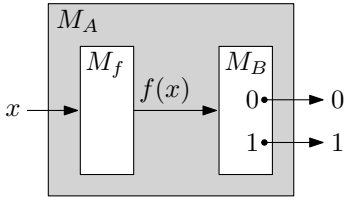
Theorem 3.9

Es seien $A \subseteq \Sigma_1^*$ und $B \subseteq \Sigma_2^*$ zwei Sprachen, für die $A \leq B$ gilt. Ist B entscheidbar, so ist auch A entscheidbar. Ist A nicht entscheidbar, so ist auch B nicht entscheidbar.



Theorem 3.9

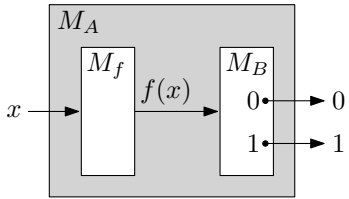
Es seien $A \subseteq \Sigma_1^*$ und $B \subseteq \Sigma_2^*$ zwei Sprachen, für die $A \leq B$ gilt. Ist B entscheidbar, so ist auch A entscheidbar. Ist A nicht entscheidbar, so ist auch B nicht entscheidbar.



Beweis: Sei $A \leq B$ und sei B entscheidbar. Dann gibt es TM M_B, die B entscheidet, und TM M_f, die die Reduktion f berechnet.

Theorem 3.9

Es seien $A \subseteq \Sigma_1^*$ und $B \subseteq \Sigma_2^*$ zwei Sprachen, für die $A \leq B$ gilt. Ist B entscheidbar, so ist auch A entscheidbar. Ist A nicht entscheidbar, so ist auch B nicht entscheidbar.



Beweis: Sei $A \leq B$ und sei B entscheidbar. Dann gibt es TM M_B, die B entscheidet, und TM M_f, die die Reduktion f berechnet.

Konstruiere TM M_A für A wie folgt:

(1) Berechne bei Eingabe x mit M_f die Zeichenkette $f(x)$.

(2) Entscheide mit M_B, ob $f(x) \in B$ gilt. □

Definition 3.10

Das **spezielle Halteproblem** H_ε sei definiert durch

$$H_\varepsilon = \{\langle M \rangle \mid M \text{ hält auf } \varepsilon\} \subseteq \{0,1\}^*.$$

Definition 3.10

Das **spezielle Halteproblem** H_ε sei definiert durch

$$H_\varepsilon = \{\langle M \rangle \mid M \text{ hält auf } \varepsilon\} \subseteq \{0,1\}^*.$$

Das **vollständige Halteproblem** H_{all} sei definiert durch

$$H_{\text{all}} = \{\langle M \rangle \mid M \text{ hält auf jeder Eingabe aus } \{0,1\}^*\} \subseteq \{0,1\}^*.$$

Definition 3.10

Das **spezielle Halteproblem** H_ε sei definiert durch

$$H_\varepsilon = \{\langle M \rangle \mid M \text{ hält auf } \varepsilon\} \subseteq \{0,1\}^*.$$

Das **vollständige Halteproblem** H_{all} sei definiert durch

$$H_{\text{all}} = \{\langle M \rangle \mid M \text{ hält auf jeder Eingabe aus } \{0,1\}^*\} \subseteq \{0,1\}^*.$$

Die **universelle Sprache** U sei definiert durch

$$U = \{\langle M \rangle w \mid M \text{ akzeptiert } w\} \subseteq \{0,1\}^*.$$

Theorem 3.11

Die universelle Sprache $U = \{\langle M \rangle w \mid M \text{ akzeptiert } w\} \subseteq \{0,1\}^*$ ist nicht entscheidbar.

Theorem 3.11

Die universelle Sprache $U = \{\langle M \rangle w \mid M \text{ akzeptiert } w\} \subseteq \{0,1\}^*$ ist nicht entscheidbar.

Beweis: Wir zeigen $H \leq U$. Dazu konstruieren wir eine Funktion $f\colon \{0,1\}^* \to \{0,1\}^*$, die Eingaben für das Halteproblem H auf Eingaben für die universelle Sprache U abbildet.

Theorem 3.11

Die universelle Sprache $U = \{\langle M \rangle w \mid M \text{ akzeptiert } w\} \subseteq \{0,1\}^*$ ist nicht entscheidbar.

Beweis: Wir zeigen $H \leq U$. Dazu konstruieren wir eine Funktion $f\colon \{0,1\}^* \to \{0,1\}^*$, die Eingaben für das Halteproblem H auf Eingaben für die universelle Sprache U abbildet.

$$f(x) = \begin{cases} x & \text{falls } x \text{ nicht mit Gödelnummer beginnt} \\ \langle M^\star_M \rangle w & \text{falls } x = \langle M \rangle w \text{ für eine TM } M \end{cases}$$

Theorem 3.11

Die universelle Sprache $U = \{\langle M \rangle w \mid M \text{ akzeptiert } w\} \subseteq \{0,1\}^*$ ist nicht entscheidbar.

Beweis: Wir zeigen $H \leq U$. Dazu konstruieren wir eine Funktion $f\colon \{0,1\}^* \to \{0,1\}^*$, die Eingaben für das Halteproblem H auf Eingaben für die universelle Sprache U abbildet.

$$f(x) = \begin{cases} x & \text{falls } x \text{ nicht mit Gödelnummer beginnt} \\ \langle M_M^\star \rangle w & \text{falls } x = \langle M \rangle w \text{ für eine TM } M \end{cases}$$

Die TM $M_M^\star$ simuliert das Verhalten von M auf der gegebenen Eingabe Schritt für Schritt, solange bis M terminiert. Anschließend akzeptiert $M_M^\star$ die Eingabe unabhängig von der Ausgabe von M.

Theorem 3.11

Die universelle Sprache $U = \{\langle M \rangle w \mid M \text{ akzeptiert } w\} \subseteq \{0,1\}^*$ ist nicht entscheidbar.

Beweis: Wir zeigen $H \leq U$. Dazu konstruieren wir eine Funktion $f\colon \{0,1\}^* \to \{0,1\}^*$, die Eingaben für das Halteproblem H auf Eingaben für die universelle Sprache U abbildet.

$$f(x) = \begin{cases} x & \text{falls } x \text{ nicht mit Gödelnummer beginnt} \\ \langle M_M^\star \rangle w & \text{falls } x = \langle M \rangle w \text{ für eine TM } M \end{cases}$$

Die TM $M_M^\star$ simuliert das Verhalten von M auf der gegebenen Eingabe Schritt für Schritt, solange bis M terminiert. Anschließend akzeptiert $M_M^\star$ die Eingabe unabhängig von der Ausgabe von M.

Die Funktion f ist berechenbar, da $\langle M_M^\star \rangle$ für gegebenes $\langle M \rangle$ konstruiert werden kann.

Zu zeigen: $x \in H \iff f(x) \in U$.

Zu zeigen: $x \in H \iff f(x) \in U$.

„$\Rightarrow$“:

- $x \in H \Rightarrow x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w hält.

Zu zeigen: $x \in H \iff f(x) \in U$.

„$\Rightarrow$“:

- $x \in H \Rightarrow x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w hält.
- $M_M^\star$ simuliert das Verhalten von M. $\Rightarrow$ $M_M^\star$ hält auf w. $\Rightarrow$ $M_M^\star$ akzeptiert w.

Zu zeigen: $x \in H \iff f(x) \in U$.

„$\Rightarrow$“:

- $x \in H \Rightarrow x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w hält.
- $M_M^\star$ simuliert das Verhalten von M. $\Rightarrow$ $M_M^\star$ hält auf w. $\Rightarrow$ $M_M^\star$ akzeptiert w.
- Dies bedeutet, dass $f(x) = \langle M_M^\star \rangle w \in U$ gilt.

Zu zeigen: $x \in H \iff f(x) \in U$.

„$\Rightarrow$“:

- $x \in H \Rightarrow x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w hält.
- $M_M^\star$ simuliert das Verhalten von M. $\Rightarrow$ $M_M^\star$ hält auf w. $\Rightarrow$ $M_M^\star$ akzeptiert w.
- Dies bedeutet, dass $f(x) = \langle M_M^\star \rangle w \in U$ gilt.

„$\Leftarrow$“:

- $x \notin H \Rightarrow$ entweder x beginnt nicht mit Gödelnummer oder $x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w nicht hält.

Zu zeigen: $x \in H \iff f(x) \in U$.

„$\Rightarrow$“:

- $x \in H \Rightarrow x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w hält.
- $M_M^\star$ simuliert das Verhalten von M. $\Rightarrow$ $M_M^\star$ hält auf w. $\Rightarrow$ $M_M^\star$ akzeptiert w.
- Dies bedeutet, dass $f(x) = \langle M_M^\star \rangle w \in U$ gilt.

„$\Leftarrow$“:

- $x \notin H \Rightarrow$ entweder x beginnt nicht mit Gödelnummer oder $x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w nicht hält.
- Beginnt x nicht mit Gödelnummer, so gilt $f(x) = x \notin U$.

Zu zeigen: $x \in H \iff f(x) \in U$.

„$\Rightarrow$“:

- $x \in H \Rightarrow x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w hält.
- $M_M^\star$ simuliert das Verhalten von M. $\Rightarrow$ $M_M^\star$ hält auf w. $\Rightarrow$ $M_M^\star$ akzeptiert w.
- Dies bedeutet, dass $f(x) = \langle M_M^\star \rangle w \in U$ gilt.

„$\Leftarrow$“:

- $x \notin H \Rightarrow$ entweder x beginnt nicht mit Gödelnummer oder $x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w nicht hält.
- Beginnt x nicht mit Gödelnummer, so gilt $f(x) = x \notin U$.
- Sonst: $M_M^\star$ simuliert das Verhalten von M. $\Rightarrow$ $M_M^\star$ hält nicht auf w. $\Rightarrow$ $M_M^\star$ akzeptiert w nicht. Dies bedeutet, dass $f(x) = \langle M_M^\star \rangle w \notin U$ gilt. $\square$

Theorem 3.11

Das spezielle Halteproblem $H_\varepsilon = \{\langle M \rangle \mid M \text{ hält auf } \varepsilon\} \subseteq \{0,1\}^*$ ist nicht entscheidbar.

Theorem 3.11

Das spezielle Halteproblem $H_\varepsilon = \{\langle M \rangle \mid M \text{ hält auf } \varepsilon\} \subseteq \{0,1\}^*$ ist nicht entscheidbar.

Beweis: Wir zeigen $H \leq H_\varepsilon$. Dazu konstruieren wir eine Funktion $f\colon \{0,1\}^* \to \{0,1\}^*$, die Eingaben für das Halteproblem H auf Eingaben für das spezielle Halteproblem H_ε abbildet.

Theorem 3.11

Das spezielle Halteproblem $H_\varepsilon = \{\langle M \rangle \mid M \text{ hält auf } \varepsilon\} \subseteq \{0,1\}^*$ ist nicht entscheidbar.

Beweis: Wir zeigen $H \leq H_\varepsilon$. Dazu konstruieren wir eine Funktion $f\colon \{0,1\}^* \to \{0,1\}^*$, die Eingaben für das Halteproblem H auf Eingaben für das spezielle Halteproblem H_ε abbildet.

$$f(x) = \begin{cases} x & \text{falls } x \text{ nicht mit Gödelnummer beginnt} \\ \langle M^\star_{M,w} \rangle & \text{falls } x = \langle M \rangle w \text{ für eine TM } M \end{cases}$$

Theorem 3.11

Das spezielle Halteproblem $H_\varepsilon = \{\langle M \rangle \mid M \text{ hält auf } \varepsilon\} \subseteq \{0,1\}^*$ ist nicht entscheidbar.

Beweis: Wir zeigen $H \leq H_\varepsilon$. Dazu konstruieren wir eine Funktion $f\colon \{0,1\}^* \to \{0,1\}^*$, die Eingaben für das Halteproblem H auf Eingaben für das spezielle Halteproblem H_ε abbildet.

$$f(x) = \begin{cases} x & \text{falls } x \text{ nicht mit Gödelnummer beginnt} \\ \langle M^\star_{M,w} \rangle & \text{falls } x = \langle M \rangle w \text{ für eine TM } M \end{cases}$$

Die TM $M^\star_{M,w}$ löscht die Eingabe und simuliert das Verhalten von M auf w Schritt für Schritt.

Theorem 3.11

Das spezielle Halteproblem $H_\varepsilon = \{\langle M\rangle \mid M \text{ hält auf } \varepsilon\} \subseteq \{0,1\}^*$ ist nicht entscheidbar.

Beweis: Wir zeigen $H \leq H_\varepsilon$. Dazu konstruieren wir eine Funktion $f\colon \{0,1\}^* \to \{0,1\}^*$, die Eingaben für das Halteproblem H auf Eingaben für das spezielle Halteproblem H_ε abbildet.

$$f(x) = \begin{cases} x & \text{falls } x \text{ nicht mit Gödelnummer beginnt} \\ \langle M^\star_{M,w}\rangle & \text{falls } x = \langle M\rangle w \text{ für eine TM } M \end{cases}$$

Die TM $M^\star_{M,w}$ löscht die Eingabe und simuliert das Verhalten von M auf w Schritt für Schritt.

Die Funktion f ist berechenbar, da $\langle M^\star_{M,w}\rangle$ für gegebenes $\langle M\rangle$ konstruiert werden kann.

Zu zeigen: $x \in H \iff f(x) \in H_\varepsilon$.

Zu zeigen: $x \in H \iff f(x) \in H_\varepsilon$.

„$\Rightarrow$“:

- $x \in H \Rightarrow x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w hält.

Zu zeigen: $x \in H \iff f(x) \in H_\varepsilon$.

„$\Rightarrow$“:

- $x \in H \Rightarrow x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w hält.
- $M^\star_{M,w}$ simuliert für jede Eingabe das Verhalten von M auf w. $\Rightarrow$ $M^\star_{M,w}$ hält auf ε.
- Dies bedeutet, dass $f(x) = \langle M^\star_{M,w} \rangle \in H_\varepsilon$ gilt.

Zu zeigen: $x \in H \iff f(x) \in H_\varepsilon$.

„$\Rightarrow$“:

- $x \in H \Rightarrow x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w hält.
- $M^\star_{M,w}$ simuliert für jede Eingabe das Verhalten von M auf w. $\Rightarrow$ $M^\star_{M,w}$ hält auf ε.
- Dies bedeutet, dass $f(x) = \langle M^\star_{M,w} \rangle \in H_\varepsilon$ gilt.

„$\Leftarrow$“:

- $x \notin H \Rightarrow$ entweder x beginnt nicht mit Gödelnummer oder $x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w nicht hält.

Zu zeigen: $x \in H \iff f(x) \in H_\varepsilon$.

„$\Rightarrow$“:

- $x \in H \Rightarrow x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w hält.
- $M^\star_{M,w}$ simuliert für jede Eingabe das Verhalten von M auf w. $\Rightarrow$ $M^\star_{M,w}$ hält auf ε.
- Dies bedeutet, dass $f(x) = \langle M^\star_{M,w} \rangle \in H_\varepsilon$ gilt.

„$\Leftarrow$“:

- $x \notin H \Rightarrow$ entweder x beginnt nicht mit Gödelnummer oder $x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w nicht hält.
- Beginnt x nicht mit Gödelnummer, so gilt $f(x) = x \notin H_\varepsilon$.

Zu zeigen: $x \in H \iff f(x) \in H_\varepsilon$.

„$\Rightarrow$“:

- $x \in H \Rightarrow x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w hält.
- $M^\star_{M,w}$ simuliert für jede Eingabe das Verhalten von M auf w. $\Rightarrow$ $M^\star_{M,w}$ hält auf ε.
- Dies bedeutet, dass $f(x) = \langle M^\star_{M,w} \rangle \in H_\varepsilon$ gilt.

„$\Leftarrow$“:

- $x \notin H \Rightarrow$ entweder x beginnt nicht mit Gödelnummer oder $x = \langle M \rangle w$ für eine TM M und ein $w \in \{0,1\}^*$, sodass M auf w nicht hält.
- Beginnt x nicht mit Gödelnummer, so gilt $f(x) = x \notin H_\varepsilon$.
- Sonst: $M^\star_{M,w}$ simuliert bei jeder Eingabe das Verhalten von M auf w. $\Rightarrow$ $M^\star_{M,w}$ hält nicht auf ε. Dies bedeutet, dass $f(x) = \langle M^\star_{M,w} \rangle \notin H_\varepsilon$ gilt. $\square$