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

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

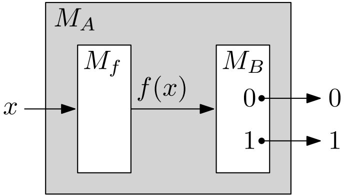
für alle $x \in \Sigma_1^*$ gilt. Existiert eine solche Reduktion, so heißt A auf B **reduzierbar** und wir schreiben $A \leq B$.

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.

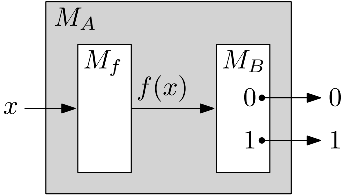
Theorem 3.9

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

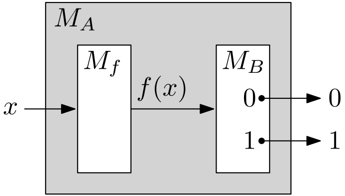
Es seien $A \subseteq \Sigma_1^*$ und $B \subseteq \Sigma_2^*$ zwei Sprachen, für die $A \leq B$ gilt. Ist B entscheidbar, so ist auch A entscheidbar. Ist A nicht entscheidbar, so ist auch B nicht entscheidbar.



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. $\square$

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.

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

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.

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

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

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

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

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

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

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

1. Fall: $x \in H$

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

3.3 Turing- und Many-One-Reduktionen

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

1. Fall: $x \in H$

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

3.3 Turing- und Many-One-Reduktionen

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

1. Fall: $x \in H$

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

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

1. Fall: $x \in H$

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

2. Fall: $x \notin H$

- 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.

3.3 Turing- und Many-One-Reduktionen

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

1. Fall: $x \in H$

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

2. Fall: $x \notin H$

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

1. Fall: $x \in H$

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

2. Fall: $x \notin H$

- 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: $M^\star$ simuliert das Verhalten von M. $\Rightarrow$ $M^\star$ hält nicht auf w. $\Rightarrow$ $M^\star$ akzeptiert w nicht. Dies bedeutet, dass $f(x) = \langle M^\star \rangle w \notin U$ gilt. $\square$

Theorem 3.12

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

Theorem 3.12

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 3.12

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.

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

Theorem 3.12

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.

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

Die TM $M^\star$ löscht die Eingabe und simuliert das Verhalten von M auf w Schritt für Schritt.

Theorem 3.12

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.

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

Die TM $M^\star$ löscht die Eingabe und simuliert das Verhalten von M auf w Schritt für Schritt.

Die Funktion f ist berechenbar, da $\langle M^\star\rangle$ für gegebene $\langle M\rangle$ und w konstruiert werden kann.

3.3 Turing- und Many-One-Reduktionen

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

3.3 Turing- und Many-One-Reduktionen

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

1. Fall: $x \in H$

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

3.3 Turing- und Many-One-Reduktionen

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

1. Fall: $x \in H$

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

3.3 Turing- und Many-One-Reduktionen

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

1. Fall: $x \in H$

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

2. Fall: $x \notin H$

- 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.

3.3 Turing- und Many-One-Reduktionen

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

1. Fall: $x \in H$

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

2. Fall: $x \notin H$

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

1. Fall: $x \in H$

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

2. Fall: $x \notin H$

- 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: $M^\star$ simuliert bei jeder Eingabe das Verhalten von M auf w. $\Rightarrow$ $M^\star$ hält nicht auf ε. Dies bedeutet, dass $f(x) = \langle M^\star \rangle \notin H_\varepsilon$ gilt. $\square$

Sei $\Sigma = \{0, 1\}$. Es bezeichne

$$\mathcal{R} = \{f\colon \Sigma^* \to \Sigma^* \cup \{\bot\} \mid \exists \text{ Turingmaschine } M \text{ mit } f_M = f\}$$

die **Menge aller von Turingmaschinen berechenbaren Funktionen**.

3.4 Der Satz von Rice

Sei $\Sigma = \{0, 1\}$. Es bezeichne

$$\mathcal{R} = \{f\colon \Sigma^* \to \Sigma^* \cup \{\perp\} \mid \exists \text{ Turingmaschine } M \text{ mit } f_M = f\}$$

die **Menge aller von Turingmaschinen berechenbaren Funktionen**.

Für $S \subseteq \mathcal{R}$ sei

$$L(S) = \{\langle M \rangle \mid f_M \in S\}$$

die Menge der Gödelnummern der **Turingmaschinen, die eine Funktion aus S berechnen**.

Sei $\Sigma = \{0, 1\}$. Es bezeichne

$$\mathcal{R} = \{f\colon \Sigma^* \to \Sigma^* \cup \{\perp\} \mid \exists \text{ Turingmaschine } M \text{ mit } f_M = f\}$$

die **Menge aller von Turingmaschinen berechenbaren Funktionen**.

Für $S \subseteq \mathcal{R}$ sei

$$L(S) = \{\langle M \rangle \mid f_M \in S\}$$

die Menge der Gödelnummern der **Turingmaschinen, die eine Funktion aus S berechnen**.

Theorem 3.13 (Satz von Rice)

Es sei $S \subseteq \mathcal{R}$ mit $S \neq \emptyset$ und $S \neq \mathcal{R}$ eine Teilmenge der berechenbaren Funktionen.
Dann ist die Sprache L(S) nicht entscheidbar.

Theorem 3.13 (Satz von Rice)

Es sei $S \subseteq \mathcal{R}$ mit $S \neq \emptyset$ und $S \neq \mathcal{R}$ eine Teilmenge der berechenbaren Funktionen.

Dann ist die Sprache L(S) nicht entscheidbar.

Beispiele:

Theorem 3.13 (Satz von Rice)

Es sei $S \subseteq \mathcal{R}$ mit $S \neq \emptyset$ und $S \neq \mathcal{R}$ eine Teilmenge der berechenbaren Funktionen.

Dann ist die Sprache L(S) nicht entscheidbar.

Beispiele:

- Sei $S = \{f\}$ für

$$f(x) = \begin{cases} 1 & \mathrm{val}(x) \text{ ist eine Primzahl} \\ 0 & \text{sonst} \end{cases}$$

Theorem 3.13 (Satz von Rice)

Es sei $S \subseteq \mathcal{R}$ mit $S \neq \emptyset$ und $S \neq \mathcal{R}$ eine Teilmenge der berechenbaren Funktionen.

Dann ist die Sprache $L(S)$ nicht entscheidbar.

Beispiele:

- Sei $S = \{f\}$ für

$$f(x) = \begin{cases} 1 & \mathrm{val}(x) \text{ ist eine Primzahl} \\ 0 & \text{sonst} \end{cases}$$

$\Rightarrow L(S) =$ Menge der Gödelnummern von Turingmaschinen, die korrekt entscheiden, ob die Eingabe eine Primzahl ist.

Theorem 3.13 (Satz von Rice)

Es sei $S \subseteq \mathcal{R}$ mit $S \neq \emptyset$ und $S \neq \mathcal{R}$ eine Teilmenge der berechenbaren Funktionen.

Dann ist die Sprache L(S) nicht entscheidbar.

Beispiele:

- Sei $S = \{f\}$ für

$$f(x) = \begin{cases} 1 & \mathrm{val}(x) \text{ ist eine Primzahl} \\ 0 & \text{sonst} \end{cases}$$

$\Rightarrow L(S) =$ Menge der Gödelnummern von Turingmaschinen, die korrekt entscheiden, ob die Eingabe eine Primzahl ist.

$\Rightarrow$ Wir können nicht entscheiden, ob eine gegebene Turingmaschine korrekt entscheidet, ob die Eingabe eine Primzahl ist.

Theorem 3.13 (Satz von Rice)

Es sei $S \subseteq \mathcal{R}$ mit $S \neq \emptyset$ und $S \neq \mathcal{R}$ eine Teilmenge der berechenbaren Funktionen.

Dann ist die Sprache $L(S)$ nicht entscheidbar.

Beispiele:

- Quadrieren als Relation:

$$R = \{(x, y) \mid \mathrm{val}(y) = \mathrm{val}(x)^2\} \subseteq \{0, 1\}^* \times \{0, 1\}^*.$$

Theorem 3.13 (Satz von Rice)

Es sei $S \subseteq \mathcal{R}$ mit $S \neq \emptyset$ und $S \neq \mathcal{R}$ eine Teilmenge der berechenbaren Funktionen.

Dann ist die Sprache $L(S)$ nicht entscheidbar.

Beispiele:

- Quadrieren als Relation:

$$R = \{(x, y) \mid \mathrm{val}(y) = \mathrm{val}(x)^2\} \subseteq \{0, 1\}^* \times \{0, 1\}^*.$$

  Wir definieren S als die Menge aller berechenbaren Funktionen, die bei jeder Eingabe $x \in \{0, 1\}^*$ eine Ausgabe $y \in \{0, 1\}^*$ mit $(x, y) \in R$ produzieren:

$$S = \{f \in \mathcal{R} \mid \forall x \in \{0, 1\}^* : (x, f(x)) \in R\}.$$

Theorem 3.13 (Satz von Rice)

Es sei $S \subseteq \mathcal{R}$ mit $S \neq \emptyset$ und $S \neq \mathcal{R}$ eine Teilmenge der berechenbaren Funktionen.

Dann ist die Sprache $L(S)$ nicht entscheidbar.

Beispiele:

- Quadrieren als Relation:

$$R = \{(x, y) \mid \mathrm{val}(y) = \mathrm{val}(x)^2\} \subseteq \{0, 1\}^* \times \{0, 1\}^*.$$

  Wir definieren S als die Menge aller berechenbaren Funktionen, die bei jeder Eingabe $x \in \{0, 1\}^*$ eine Ausgabe $y \in \{0, 1\}^*$ mit $(x, y) \in R$ produzieren:

$$S = \{f \in \mathcal{R} \mid \forall x \in \{0, 1\}^* : (x, f(x)) \in R\}.$$

  $\Rightarrow$ Wir können nicht entscheiden, ob eine gegebene Turingmaschine die Eingabe quadriert.

Beweis mittels Turing-Reduktion von H_ε auf $L(S)$:

Dazu konstruieren wir TM M_{H_ε}, die H_ε mithilfe einer TM $M_{L(S)}$ für $L(S)$ entscheidet.

Beweis mittels Turing-Reduktion von H_ε auf $L(S)$:

Dazu konstruieren wir TM M_{H_ε}, die H_ε mithilfe einer TM $M_{L(S)}$ für $L(S)$ entscheidet.

Es sei $u\colon \Sigma^* \to \Sigma^* \cup \{\perp\}$ mit $u(x) = \perp$ für alle $x \in \Sigma^*$. Es gilt $u \in \mathcal{R}$.

Beweis mittels Turing-Reduktion von H_ε auf $L(S)$:

Dazu konstruieren wir TM M_{H_ε}, die H_ε mithilfe einer TM $M_{L(S)}$ für $L(S)$ entscheidet.

Es sei $u\colon \Sigma^* \to \Sigma^* \cup \{\perp\}$ mit $u(x) = \perp$ für alle $x \in \Sigma^*$. Es gilt $u \in \mathcal{R}$.

1. Fall: $u \notin S$.

Es sei $f \in S$ beliebig (dann gilt $f \neq u$) und sei M_f eine TM, die f berechnet.

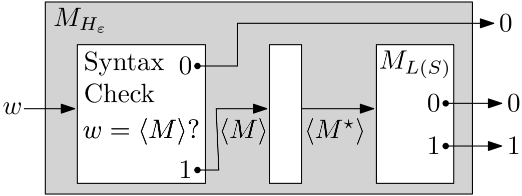
Beweis mittels Turing-Reduktion von H_ε auf $L(S)$:

Dazu konstruieren wir TM M_{H_ε}, die H_ε mithilfe einer TM $M_{L(S)}$ für $L(S)$ entscheidet.

Es sei $u\colon \Sigma^* \to \Sigma^* \cup \{\perp\}$ mit $u(x) = \perp$ für alle $x \in \Sigma^*$. Es gilt $u \in \mathcal{R}$.

1. Fall: $u \notin S$.

Es sei $f \in S$ beliebig (dann gilt $f \neq u$) und sei M_f eine TM, die f berechnet.



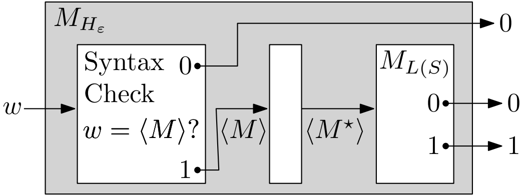
Beweis mittels Turing-Reduktion von H_ε auf $L(S)$:

Dazu konstruieren wir TM M_{H_ε}, die H_ε mithilfe einer TM $M_{L(S)}$ für $L(S)$ entscheidet.

Es sei $u\colon \Sigma^* \to \Sigma^* \cup \{\perp\}$ mit $u(x) = \perp$ für alle $x \in \Sigma^*$. Es gilt $u \in \mathcal{R}$.

1. Fall: $u \notin S$.

Es sei $f \in S$ beliebig (dann gilt $f \neq u$) und sei M_f eine TM, die f berechnet.



Verhalten der TM M* bei Eingabe x:

Schritt 1: $M^\star$ simuliert das Verhalten von M auf ε.

Schritt 2: $M^\star$ simuliert M_f auf Eingabe x und übernimmt die Ausgabe.

Verhalten der TM $M^\star$ bei Eingabe x:

Schritt 1: $M^\star$ simuliert das Verhalten von M auf ε.

Schritt 2: $M^\star$ simuliert M_f auf Eingabe x und übernimmt die Ausgabe.

Verhalten der TM $M^\star$ bei Eingabe x:

Schritt 1: $M^\star$ simuliert das Verhalten von M auf ε.

Schritt 2: $M^\star$ simuliert M_f auf Eingabe x und übernimmt die Ausgabe.

Beobachtung:

M hält nicht auf ε. $\Rightarrow$ $M^\star$ hält auf keiner Eingabe. $\Rightarrow$ $M^\star$ berechnet u.

Verhalten der TM M* bei Eingabe x:

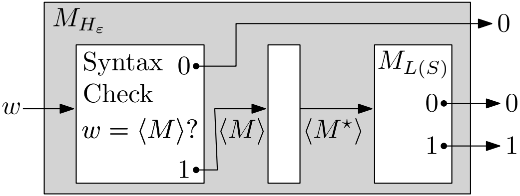
Schritt 1: $M^\star$ simuliert das Verhalten von M auf ε.

Schritt 2: $M^\star$ simuliert M_f auf Eingabe x und übernimmt die Ausgabe.

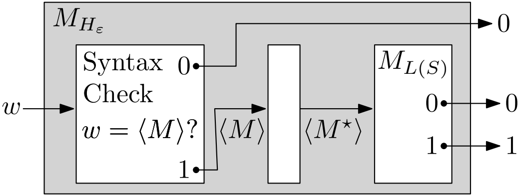
Beobachtung:

M hält nicht auf ε. $\Rightarrow$ $M^\star$ hält auf keiner Eingabe. $\Rightarrow$ $M^\star$ berechnet u.

M hält auf ε. $\Rightarrow$ $M^\star$ erreicht Schritt 2. $\Rightarrow$ $M^\star$ berechnet die Funktion f.

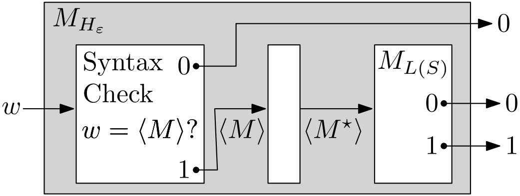


M_{H_ε} löst das spezielle Halteproblem H_ε:



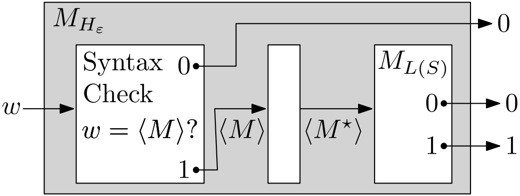
M_{H_ε} löst das spezielle Halteproblem H_ε:

- Sei $w \in H_\varepsilon$. Dann gilt $w = \langle M \rangle$ für eine TM M, die auf ε hält.



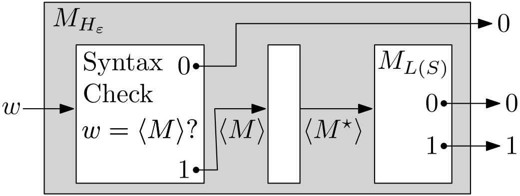
M_{H_ε} löst das spezielle Halteproblem H_ε:

- Sei $w \in H_\varepsilon$. Dann gilt $w = \langle M \rangle$ für eine TM M, die auf ε hält.
  $\Rightarrow M^\star$ berechnet f.



M_{H_ε} löst das spezielle Halteproblem H_ε:

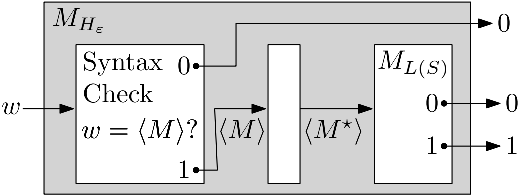
- Sei $w \in H_\varepsilon$. Dann gilt $w = \langle M \rangle$ für eine TM M, die auf ε hält.
  $\Rightarrow M^\star$ berechnet f.
  $\Rightarrow \langle M^\star \rangle \in L(S)$ wegen $f \in S$.



M_{H_ε} löst das spezielle Halteproblem H_ε:

- Sei $w \in H_\varepsilon$. Dann gilt $w = \langle M \rangle$ für eine TM M, die auf ε hält.
  $\Rightarrow$ $M^\star$ berechnet f.
  $\Rightarrow$ $\langle M^\star \rangle \in L(S)$ wegen $f \in S$.
  $\Rightarrow$ $M_{L(S)}$ akzeptiert die Eingabe $\langle M^\star \rangle$.

3.4 Der Satz von Rice



M_{H_ε} löst das spezielle Halteproblem H_ε:

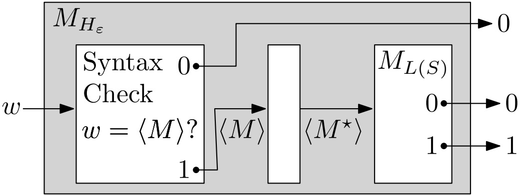
- Sei $w \in H_\varepsilon$. Dann gilt $w = \langle M \rangle$ für eine TM M, die auf ε hält.

  $\Rightarrow$ $M^\star$ berechnet f.

  $\Rightarrow$ $\langle M^\star \rangle \in L(S)$ wegen $f \in S$.

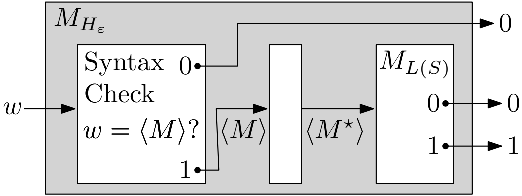
  $\Rightarrow$ $M_{L(S)}$ akzeptiert die Eingabe $\langle M^\star \rangle$.

  $\Rightarrow$ M_{H_ε} akzeptiert die Eingabe w.



M_{H_ε} löst das spezielle Halteproblem H_ε:

- Sei $w \notin H_\varepsilon$.

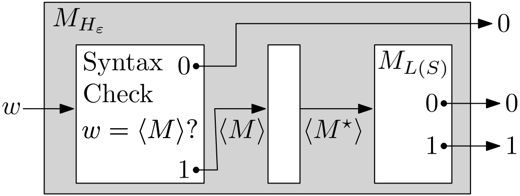


M_{H_ε} löst das spezielle Halteproblem H_ε:

- Sei $w \notin H_\varepsilon$.

  Entweder: w ist keine Gödelnummer. $\Rightarrow$ M_{H_ε} verwirft w.

  Oder: $w = \langle M \rangle$ für eine TM M, die nicht auf ε hält.



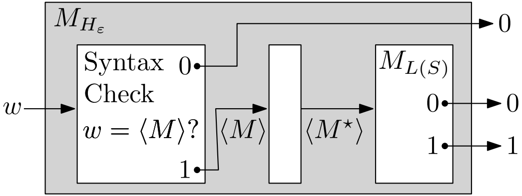
M_{H_ε} löst das spezielle Halteproblem H_ε:

- Sei $w \notin H_\varepsilon$.

  Entweder: w ist keine Gödelnummer. $\Rightarrow$ M_{H_ε} verwirft w.

  Oder: $w = \langle M \rangle$ für eine TM M, die nicht auf ε hält.

  $\Rightarrow$ $M^\star$ berechnet u.



M_{H_ε} löst das spezielle Halteproblem H_ε:

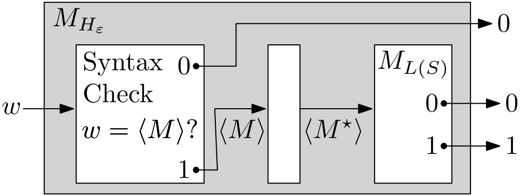
- Sei $w \notin H_\varepsilon$.

  Entweder: w ist keine Gödelnummer. $\Rightarrow$ M_{H_ε} verwirft w.

  Oder: $w = \langle M \rangle$ für eine TM M, die nicht auf ε hält.

  $\Rightarrow$ $M^\star$ berechnet u.

  $\Rightarrow$ $\langle M^\star \rangle \notin L(S)$ wegen $u \notin S$.



M_{H_ε} löst das spezielle Halteproblem H_ε:

- Sei $w \notin H_\varepsilon$.

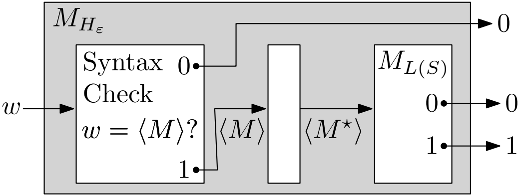
  Entweder: w ist keine Gödelnummer. $\Rightarrow$ M_{H_ε} verwirft w.

  Oder: $w = \langle M \rangle$ für eine TM M, die nicht auf ε hält.

  $\Rightarrow$ $M^\star$ berechnet u.

  $\Rightarrow$ $\langle M^\star \rangle \notin L(S)$ wegen $u \notin S$.

  $\Rightarrow$ $M_{L(S)}$ verwirft die Eingabe $\langle M^\star \rangle$.



M_{H_ε} löst das spezielle Halteproblem H_ε:

- Sei $w \notin H_\varepsilon$.

  Entweder: w ist keine Gödelnummer. $\Rightarrow M_{H_\varepsilon}$ verwirft w.

  Oder: $w = \langle M \rangle$ für eine TM M, die nicht auf ε hält.

  $\Rightarrow M^\star$ berechnet u.

  $\Rightarrow \langle M^\star \rangle \notin L(S)$ wegen $u \notin S$.

  $\Rightarrow M_{L(S)}$ verwirft die Eingabe $\langle M^\star \rangle$.

  $\Rightarrow M_{H_\varepsilon}$ verwirft die Eingabe w.

2. Fall: $u \in S$.

Es sei $f \in \mathcal{R}$ mit $f \notin S$ beliebig (dann gilt $f \neq u$) und sei M_f eine TM, die f berechnet.

Vorgehen analog zu 1. Fall. $\square$