3 Berechenbarkeitstheorie

3.5 Rekursiv aufzählbare Sprachen

Definition 3.14

Eine Turingmaschine M **erkennt** eine Sprache $L \subseteq \Sigma^*$, wenn sie jedes Wort $w \in L$ akzeptiert und jedes Wort $w \in \Sigma^* \setminus L$ entweder verwirft oder darauf nicht terminiert.

Definition 3.14

Eine Turingmaschine M **erkennt** eine Sprache $L \subseteq \Sigma^*$, wenn sie jedes Wort $w \in L$ akzeptiert und jedes Wort $w \in \Sigma^* \setminus L$ entweder verwirft oder darauf nicht terminiert.

Eine Sprache $L \subseteq \Sigma^*$ heißt **semi-entscheidbar** oder **rekursiv aufzählbar**, wenn es eine Turingmaschine M gibt, die L erkennt.

Definition 3.14

Eine Turingmaschine M **erkennt** eine Sprache $L \subseteq \Sigma^*$, wenn sie jedes Wort $w \in L$ akzeptiert und jedes Wort $w \in \Sigma^* \setminus L$ entweder verwirft oder darauf nicht terminiert.

Eine Sprache $L \subseteq \Sigma^*$ heißt **semi-entscheidbar** oder **rekursiv aufzählbar**, wenn es eine Turingmaschine M gibt, die L erkennt.

Beispiel:

Die folgende Turingmaschine M_H **erkennt** das Halteproblem:

Bei Eingabe $\langle M \rangle w$ simuliert M_H die TM M auf w.

Terminiert diese Simulation, so akzeptiert M_H die Eingabe $\langle M \rangle w$.

Theorem 3.15 (Abschlusseigenschaften)

Es seien $L_1 \subseteq \Sigma^*$ und $L_2 \subseteq \Sigma^*$ zwei semi-entscheidbare Sprachen. Dann sind auch die Sprachen $L_1 \cup L_2$ und $L_1 \cap L_2$ semi-entscheidbar.

Theorem 3.15 (Abschlusseigenschaften)

Es seien $L_1 \subseteq \Sigma^*$ und $L_2 \subseteq \Sigma^*$ zwei semi-entscheidbare Sprachen. Dann sind auch die Sprachen $L_1 \cup L_2$ und $L_1 \cap L_2$ semi-entscheidbar.

Beweis: Es seien M_1 und M_2 Turingmaschinen, die L_1 bzw. L_2 erkennen.

$\Rightarrow$ Die Turingmaschine M_i akzeptiert jedes Wort $x \in L_i$ und kein Wort $x \notin L_i$.

Theorem 3.15 (Abschlusseigenschaften)

Es seien $L_1 \subseteq \Sigma^*$ und $L_2 \subseteq \Sigma^*$ zwei semi-entscheidbare Sprachen. Dann sind auch die Sprachen $L_1 \cup L_2$ und $L_1 \cap L_2$ semi-entscheidbar.

Beweis: Es seien M_1 und M_2 Turingmaschinen, die L_1 bzw. L_2 erkennen.
$\Rightarrow$ Die Turingmaschine M_i akzeptiert jedes Wort $x \in L_i$ und kein Wort $x \notin L_i$.

Konstruktion einer TM $M_\cap$ für $L_1 \cap L_2$:

$M_\cap(x)$

1 Simuliere M_1 auf x.
2 Simuliere M_2 auf x.
3 Akzeptiere x, wenn M_1 und M_2 die Eingabe x akzeptiert haben.
  Sonst verwirf x.

3.5 Rekursiv aufzählbare Sprachen

Theorem 3.15 (Abschlusseigenschaften)

Es seien $L_1 \subseteq \Sigma^*$ und $L_2 \subseteq \Sigma^*$ zwei semi-entscheidbare Sprachen. Dann sind auch die Sprachen $L_1 \cup L_2$ und $L_1 \cap L_2$ semi-entscheidbar.

Beweis: Es seien M_1 und M_2 Turingmaschinen, die L_1 bzw. L_2 erkennen.
$\Rightarrow$ Die Turingmaschine M_i akzeptiert jedes Wort $x \in L_i$ und kein Wort $x \notin L_i$.

Konstruktion einer TM $M_\cap$ für $L_1 \cap L_2$:

```
M∩(x)
    Simuliere M1 auf x.
    Simuliere M2 auf x.
    Akzeptiere x, wenn M1 und M2 die Eingabe x akzeptiert haben.
    Sonst verwirf x.
```

$M_\cap$ **akzeptiert jedes** $x \in L_1 \cap L_2$, da jedes solche Wort von M_1 und M_2 akzeptiert wird.

Theorem 3.15 (Abschlusseigenschaften)

Es seien $L_1 \subseteq \Sigma^*$ und $L_2 \subseteq \Sigma^*$ zwei semi-entscheidbare Sprachen. Dann sind auch die Sprachen $L_1 \cup L_2$ und $L_1 \cap L_2$ semi-entscheidbar.

Beweis: Es seien M_1 und M_2 Turingmaschinen, die L_1 bzw. L_2 erkennen.

$\Rightarrow$ Die Turingmaschine M_i akzeptiert jedes Wort $x \in L_i$ und kein Wort $x \notin L_i$.

Konstruktion einer TM $M_\cap$ für $L_1 \cap L_2$:

$M_\cap(x)$

1 Simuliere M_1 auf x.
2 Simuliere M_2 auf x.
3 Akzeptiere x, wenn M_1 und M_2 die Eingabe x akzeptiert haben.
Sonst verwirf x.

$M_\cap$ **akzeptiert jedes** $x \in L_1 \cap L_2$, da jedes solche Wort von M_1 und M_2 akzeptiert wird.
$M_\cap$ **akzeptiert kein** $x \notin L_1 \cap L_2$, da jedes solche Wort von M_1 oder M_2 nicht akzeptiert wird.

Konstruktion einer TM $M_\cup$ für $L_1 \cup L_2$:

$M_\cup(x)$

1 Simuliere M_1 auf x.

2 Simuliere M_2 auf x.

3 Akzeptiere x, wenn M_1 **oder** M_2 die Eingabe x akzeptiert haben.
Sonst verwirf x.

Konstruktion einer TM $M_\cup$ für $L_1 \cup L_2$:

$M_\cup(x)$
1 Simuliere M_1 auf x.
2 Simuliere M_2 auf x.
3 Akzeptiere x, wenn M_1 **oder** M_2 die Eingabe x akzeptiert haben.
Sonst verwirf x.

Problem: Gilt $x \notin L_1$ und $x \in L_2$, so gilt $x \in L_1 \cup L_2$.

In diesem Fall besteht die Möglichkeit, dass M_1 nicht terminiert.

Dann würde auch $M_\cup$ nicht terminieren und x damit nicht akzeptieren.

Lösung: Führe die Schritte 1 und 2 **parallel** aus.

Konstruiere $M_\cup$ dazu beispielsweise als 2-Band-TM.

Stoppe, sobald M_1 oder M_2 akzeptiert (oder beide verwerfen).

$M_\cup(x)$

1 Simuliere M_1 auf x.

2 Simuliere M_2 auf x.

3 Akzeptiere x, wenn M_1 **oder** M_2 die Eingabe x akzeptiert haben.
Sonst verwirf x.

Lösung: Führe die Schritte 1 und 2 **parallel** aus.

Konstruiere $M_\cup$ dazu beispielsweise als 2-Band-TM.

Stoppe, sobald M_1 oder M_2 akzeptiert (oder beide verwerfen).

$M_\cup(x)$
1 Simuliere M_1 auf x.
2 Simuliere M_2 auf x.
3 Akzeptiere x, wenn M_1 **oder** M_2 die Eingabe x akzeptiert haben.
Sonst verwirf x.

$M_\cup$ **akzeptiert jedes** $x \in L_1 \cup L_2$, da jedes solche Wort von M_1 oder M_2 akzeptiert wird.

Lösung: Führe die Schritte 1 und 2 **parallel** aus.

Konstruiere $M_\cup$ dazu beispielsweise als 2-Band-TM.

Stoppe, sobald M_1 oder M_2 akzeptiert (oder beide verwerfen).

$M_\cup(x)$
1. Simuliere M_1 auf x.
2. Simuliere M_2 auf x.
3. Akzeptiere x, wenn M_1 **oder** M_2 die Eingabe x akzeptiert haben.
   Sonst verwirf x.

$M_\cup$ **akzeptiert jedes** $x \in L_1 \cup L_2$, da jedes solche Wort von M_1 oder M_2 akzeptiert wird.
$M_\cup$ **akzeptiert kein** $x \notin L_1 \cup L_2$, da jedes solche Wort weder von M_1 noch von M_2 akzeptiert wird. $\square$

Theorem 3.16

Sind $L \subseteq \Sigma^*$ und das **Komplement** $\overline{L} = \Sigma^* \setminus L$ semi-entscheidbar, so ist L entscheidbar.

Theorem 3.16

Sind $L \subseteq \Sigma^*$ und das **Komplement** $\overline{L} = \Sigma^* \setminus L$ semi-entscheidbar, so ist L entscheidbar.

Beweis: Seien M_L und $M_{\overline{L}}$ Turingmaschinen, die L bzw. $\overline{L}$ erkennen.

Wir konstruieren eine TM M, die L entscheidet:

$M(x)$

Führe die folgenden Schritte parallel aus.

1 Simuliere M_L auf x. Akzeptiere x, wenn M_L Eingabe x akzeptiert.

2 Simuliere $M_{\overline{L}}$ auf x. Verwirf x, wenn $M_{\overline{L}}$ Eingabe x akzeptiert.

Theorem 3.16

Sind $L \subseteq \Sigma^*$ und das **Komplement** $\overline{L} = \Sigma^* \setminus L$ semi-entscheidbar, so ist L entscheidbar.

Beweis: Seien M_L und $M_{\overline{L}}$ Turingmaschinen, die L bzw. $\overline{L}$ erkennen.

Wir konstruieren eine TM M, die L entscheidet:

$M(x)$

Führe die folgenden Schritte parallel aus.

1. Simuliere M_L auf x. Akzeptiere x, wenn M_L Eingabe x akzeptiert.
2. Simuliere $M_{\overline{L}}$ auf x. Verwirf x, wenn $M_{\overline{L}}$ Eingabe x akzeptiert.

M **akzeptiert jedes** $x \in L$, da jedes solche Wort von M_L akzeptiert wird.

Theorem 3.16

Sind $L \subseteq \Sigma^*$ und das **Komplement** $\overline{L} = \Sigma^* \setminus L$ semi-entscheidbar, so ist L entscheidbar.

Beweis: Seien M_L und $M_{\overline{L}}$ Turingmaschinen, die L bzw. $\overline{L}$ erkennen.

Wir konstruieren eine TM M, die L entscheidet:

$M(x)$

Führe die folgenden Schritte parallel aus.

1. Simuliere M_L auf x. Akzeptiere x, wenn M_L Eingabe x akzeptiert.
2. Simuliere $M_{\overline{L}}$ auf x. Verwirf x, wenn $M_{\overline{L}}$ Eingabe x akzeptiert.

M **akzeptiert jedes** $x \in L$, da jedes solche Wort von M_L akzeptiert wird.

M **verwirft jedes** $x \notin L$, da jedes solche Wort von $M_{\overline{L}}$ akzeptiert wird. $\square$

Theorem 3.17

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

Theorem 3.17

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

Theorem 3.18

Weder das vollständige Halteproblem

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

noch sein Komplement $\overline{H_{\text{all}}}$ sind semi-entscheidbar.

Lemma

$H \leq H_{\text{all}}$

Beweis: Wir konstruieren eine Funktion $f\colon \{0,1\}^* \to \{0,1\}^*$, die Eingaben für das Halteproblem H auf Eingaben für das allgemeine Halteproblem H_{all} 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.

Analog zum Beweis von Theorem 3.12 gilt auch hier: $x \in H \iff f(x) \in H_{\text{all}}$. $\square$

Theorem 3.18

Weder das vollständige Halteproblem

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

noch sein Komplement $\overline{H_{\text{all}}}$ sind semi-entscheidbar.

Beweis von Theorem 3.18:

Teil 1: $\overline{H_{\text{all}}}$ ist nicht semi-entscheidbar

Theorem 3.18

Weder das vollständige Halteproblem

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

noch sein Komplement $\overline{H_{\text{all}}}$ sind semi-entscheidbar.

Beweis von Theorem 3.18:

Teil 1: $\overline{H_{\text{all}}}$ ist nicht semi-entscheidbar

Es gilt $H \le H_{\text{all}}$ (analog zu Beweis von Theorem 3.12).

Theorem 3.18

Weder das vollständige Halteproblem

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

noch sein Komplement $\overline{H_{\text{all}}}$ sind semi-entscheidbar.

Beweis von Theorem 3.18:

Teil 1: $\overline{H_{\text{all}}}$ ist nicht semi-entscheidbar

Es gilt $H \leq H_{\text{all}}$ (analog zu Beweis von Theorem 3.12).

$\Rightarrow \overline{H} \leq \overline{H_{\text{all}}}$ (Allgemein folgt aus $A \leq B$ auch $\overline{A} \leq \overline{B}$.)

Theorem 3.18

Weder das vollständige Halteproblem

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

noch sein Komplement $\overline{H_{\text{all}}}$ sind semi-entscheidbar.

Beweis von Theorem 3.18:

Teil 1: $\overline{H_{\text{all}}}$ ist nicht semi-entscheidbar

Es gilt $H \leq H_{\text{all}}$ (analog zu Beweis von Theorem 3.12).
$\Rightarrow \overline{H} \leq \overline{H_{\text{all}}}$ (Allgemein folgt aus $A \leq B$ auch $\overline{A} \leq \overline{B}$.)

Ausserdem ist $\overline{H}$ nicht semi-entscheidbar, denn H ist semi-entscheidbar und gemäß Theorem 3.16 wäre H sonst entscheidbar.

Theorem 3.18

Weder das vollständige Halteproblem

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

noch sein Komplement $\overline{H_{\text{all}}}$ sind semi-entscheidbar.

Beweis von Theorem 3.18:

Teil 1: $\overline{H_{\text{all}}}$ ist nicht semi-entscheidbar

Es gilt $H \leq H_{\text{all}}$ (analog zu Beweis von Theorem 3.12).
$\Rightarrow \overline{H} \leq \overline{H_{\text{all}}}$ (Allgemein folgt aus $A \leq B$ auch $\overline{A} \leq \overline{B}$.)

Ausserdem ist $\overline{H}$ nicht semi-entscheidbar, denn H ist semi-entscheidbar und gemäß Theorem 3.16 wäre H sonst entscheidbar.

Zusammen impliziert dies mit Theorem 3.17, dass $\overline{H_{\text{all}}}$ nicht semi-entscheidbar ist.

Teil 2: H_{all} ist nicht semi-entscheidbar

Konstruktion einer Reduktion $\overline{H_\varepsilon} \leq H_{\text{all}}$:

$$f(x) = \begin{cases} \langle M_1 \rangle & \text{falls } x \text{ keine Gödelnummer ist} \\ \langle M^\star \rangle & \text{falls } x = \langle M \rangle \text{ für eine TM } M \end{cases}$$

Teil 2: H_{all} ist nicht semi-entscheidbar

Konstruktion einer Reduktion $\overline{H_\varepsilon} \leq H_{\text{all}}$:

$$f(x) = \begin{cases} \langle M_1 \rangle & \text{falls } x \text{ keine Gödelnummer ist} \\ \langle M^\star \rangle & \text{falls } x = \langle M \rangle \text{ für eine TM } M \end{cases}$$

M_1 ist beliebige TM mit $\langle M_1 \rangle \in H_{\text{all}}$.

$M^\star$ simuliert bei Eingabe w, die TM M auf ε, solange bis sie entweder hält oder $|w|$ viele Schritte gemacht hat. Falls M innerhalb dieser $|w|$ Schritte hält, so geht $M^\star$ in eine Endlosschleife. Ansonsten terminiert $M^\star$.

M_1 ist beliebige TM mit $\langle M_1 \rangle \in H_{\text{all}}$.

$M^\star$ simuliert bei Eingabe w, die TM M auf ε, solange bis sie entweder hält oder $|w|$ viele Schritte gemacht hat. Falls M innerhalb dieser $|w|$ Schritte hält, so geht $M^\star$ in eine Endlosschleife. Ansonsten terminiert $M^\star$.

Zu zeigen: $x \in \overline{H_\varepsilon} \iff f(x) \in H_{\textbf{all}}$

M_1 ist beliebige TM mit $\langle M_1 \rangle \in H_{\text{all}}$.

$M^\star$ simuliert bei Eingabe w, die TM M auf ε, solange bis sie entweder hält oder $|w|$ viele Schritte gemacht hat. Falls M innerhalb dieser $|w|$ Schritte hält, so geht $M^\star$ in eine Endlosschleife. Ansonsten terminiert $M^\star$.

Zu zeigen: $x \in \overline{H_\varepsilon} \iff f(x) \in H_{\text{all}}$

Fall 1: $x \in \overline{H_\varepsilon}$ (d.h. $x \notin H_\varepsilon$)

- **Entweder** x ist keine gültige Gödelnummer, dann gilt $f(x) = \langle M_1 \rangle \in H_{\text{all}}$,

M_1 ist beliebige TM mit $\langle M_1 \rangle \in H_{\text{all}}$.

$M^\star$ simuliert bei Eingabe w, die TM M auf ε, solange bis sie entweder hält oder $|w|$ viele Schritte gemacht hat. Falls M innerhalb dieser $|w|$ Schritte hält, so geht $M^\star$ in eine Endlosschleife. Ansonsten terminiert $M^\star$.

Zu zeigen: $x \in \overline{H_\varepsilon} \iff f(x) \in H_{\text{all}}$

Fall 1: $x \in \overline{H_\varepsilon}$ (d.h. $x \notin H_\varepsilon$)

- **Entweder** x ist keine gültige Gödelnummer, dann gilt $f(x) = \langle M_1 \rangle \in H_{\text{all}}$,
- **oder** es gilt $x = \langle M \rangle$ für eine TM, die nicht auf ε hält

M_1 ist beliebige TM mit $\langle M_1 \rangle \in H_{\text{all}}$.

$M^\star$ simuliert bei Eingabe w, die TM M auf ε, solange bis sie entweder hält oder $|w|$ viele Schritte gemacht hat. Falls M innerhalb dieser $|w|$ Schritte hält, so geht $M^\star$ in eine Endlosschleife. Ansonsten terminiert $M^\star$.

Zu zeigen: $x \in \overline{H_\varepsilon} \iff f(x) \in H_{\text{all}}$

Fall 1: $x \in \overline{H_\varepsilon}$ (d.h. $x \notin H_\varepsilon$)

- **Entweder** x ist keine gültige Gödelnummer, dann gilt $f(x) = \langle M_1 \rangle \in H_{\text{all}}$,
- **oder** es gilt $x = \langle M \rangle$ für eine TM, die nicht auf ε hält $\Rightarrow M^\star$ terminiert auf jeder Eingabe $w \Rightarrow f(x) = \langle M^\star \rangle \in H_{\text{all}}$

M_1 ist beliebige TM mit $\langle M_1 \rangle \in H_{\text{all}}$.

$M^\star$ simuliert bei Eingabe w, die TM M auf ε, solange bis sie entweder hält oder $|w|$ viele Schritte gemacht hat. Falls M innerhalb dieser $|w|$ Schritte hält, so geht $M^\star$ in eine Endlosschleife. Ansonsten terminiert $M^\star$.

Zu zeigen: $x \in \overline{H_\varepsilon} \iff f(x) \in H_{\textbf{all}}$

Fall 2: $x \notin \overline{H_\varepsilon}$ (d.h. $x \in H_\varepsilon$)

- Es gilt $x = \langle M \rangle$ für eine TM M, die auf ε hält.

M_1 ist beliebige TM mit $\langle M_1 \rangle \in H_{\text{all}}$.

$M^\star$ simuliert bei Eingabe w, die TM M auf ε, solange bis sie entweder hält oder $|w|$ viele Schritte gemacht hat. Falls M innerhalb dieser $|w|$ Schritte hält, so geht $M^\star$ in eine Endlosschleife. Ansonsten terminiert $M^\star$.

Zu zeigen: $x \in \overline{H_\varepsilon} \iff f(x) \in H_{\textbf{all}}$

Fall 2: $x \notin \overline{H_\varepsilon}$ (d.h. $x \in H_\varepsilon$)

- Es gilt $x = \langle M \rangle$ für eine TM M, die auf ε hält.
- Sei $t \in \mathbb{N}$ Anzahl an Schritten, die M auf ε benötigt.

M_1 ist beliebige TM mit $\langle M_1 \rangle \in H_{\text{all}}$.

$M^\star$ simuliert bei Eingabe w, die TM M auf ε, solange bis sie entweder hält oder $|w|$ viele Schritte gemacht hat. Falls M innerhalb dieser $|w|$ Schritte hält, so geht $M^\star$ in eine Endlosschleife. Ansonsten terminiert $M^\star$.

Zu zeigen: $x \in \overline{H_\varepsilon} \iff f(x) \in H_{\text{all}}$

Fall 2: $x \notin \overline{H_\varepsilon}$ (d.h. $x \in H_\varepsilon$)

- Es gilt $x = \langle M \rangle$ für eine TM M, die auf ε hält.
- Sei $t \in \mathbb{N}$ Anzahl an Schritten, die M auf ε benötigt.
- Die TM $M^\star$ gerät für jede Eingabe w mit $|w| > t$ in eine Endlosschleife.

M_1 ist beliebige TM mit $\langle M_1 \rangle \in H_{\text{all}}$.
$M^\star$ simuliert bei Eingabe w, die TM M auf ε, solange bis sie entweder hält oder $|w|$ viele Schritte gemacht hat. Falls M innerhalb dieser $|w|$ Schritte hält, so geht $M^\star$ in eine Endlosschleife. Ansonsten terminiert $M^\star$.

Zu zeigen: $x \in \overline{H_\varepsilon} \iff f(x) \in H_{\text{all}}$

Fall 2: $x \notin \overline{H_\varepsilon}$ (d.h. $x \in H_\varepsilon$)

- Es gilt $x = \langle M \rangle$ für eine TM M, die auf ε hält.
- Sei $t \in \mathbb{N}$ Anzahl an Schritten, die M auf ε benötigt.
- Die TM $M^\star$ gerät für jede Eingabe w mit $|w| > t$ in eine Endlosschleife.
- Somit terminiert $M^\star$ nicht auf jeder Eingabe und es gilt $f(x) = \langle M^\star \rangle \notin H_{\text{all}}$. $\square$

Definition 3.19

Ein **Aufzähler** für eine Sprache $L \subseteq \Sigma^*$ ist eine TM mit einem zusätzlichen Ausgabeband, das zu Beginn leer ist.

Definition 3.19

Ein **Aufzähler** für eine Sprache $L \subseteq \Sigma^*$ ist eine TM mit einem zusätzlichen Ausgabeband, das zu Beginn leer ist.

Ein Aufzähler erhält keine Eingabe und er schreibt nach und nach Wörter aus L (durch Leerzeichen getrennt) auf das Ausgabeband.

Definition 3.19

Ein **Aufzähler** für eine Sprache $L \subseteq \Sigma^*$ ist eine TM mit einem zusätzlichen Ausgabeband, das zu Beginn leer ist.

Ein Aufzähler erhält keine Eingabe und er schreibt nach und nach Wörter aus L (durch Leerzeichen getrennt) auf das Ausgabeband.

Er schreibt keine Wörter auf das Ausgabeband, die nicht zu L gehören, und zu jedem Wort $w \in L$ existiert ein Index $i_w \in \mathbb{N}$, sodass das Wort w nach i_w Schritten des Aufzählers auf dem Ausgabeband steht.

3.5 Rekursiv aufzählbare Sprachen

Definition 3.19

Ein **Aufzähler** für eine Sprache $L \subseteq \Sigma^*$ ist eine TM mit einem zusätzlichen Ausgabeband, das zu Beginn leer ist.

Ein Aufzähler erhält keine Eingabe und er schreibt nach und nach Wörter aus L (durch Leerzeichen getrennt) auf das Ausgabeband.

Er schreibt keine Wörter auf das Ausgabeband, die nicht zu L gehören, und zu jedem Wort $w \in L$ existiert ein Index $i_w \in \mathbb{N}$, sodass das Wort w nach i_w Schritten des Aufzählers auf dem Ausgabeband steht.

Theorem 3.20

Eine Sprache L ist genau dann semi-entscheidbar, wenn ein Aufzähler für L existiert.

Beweis: „$\Leftarrow$“: Sei A ein Aufzähler für L.

Konstruktion einer TM M, die L erkennt:

Beweis: „$\Leftarrow$“: Sei A ein Aufzähler für L.

Konstruktion einer TM M, die L erkennt:

Bei Eingabe w simuliert M den Aufzähler A. M terminiert und akzeptiert die Eingabe w, sobald A das Wort w auf das Ausgabeband schreibt.

Beweis: „$\Leftarrow$“: Sei A ein Aufzähler für L.

Konstruktion einer TM M, die L erkennt:

Bei Eingabe w simuliert M den Aufzähler A. M terminiert und akzeptiert die Eingabe w, sobald A das Wort w auf das Ausgabeband schreibt.

Sei $w \in L$. Dann schreibt A das Wort w nach endlich vielen Schritten auf das Ausgabeband. $\Rightarrow$ M akzeptiert w.

Beweis: „$\Leftarrow$“: Sei A ein Aufzähler für L.

Konstruktion einer TM M, die L erkennt:

Bei Eingabe w simuliert M den Aufzähler A. M terminiert und akzeptiert die Eingabe w, sobald A das Wort w auf das Ausgabeband schreibt.

Sei $w \in L$. Dann schreibt A das Wort w nach endlich vielen Schritten auf das Ausgabeband. $\Rightarrow$ M akzeptiert w.

Seit $w \notin L$. Dann schreibt A das Wort w nie auf das Ausgabeband. $\Rightarrow$ M terminiert nicht auf w.

„$\Rightarrow$“: Sei eine TM M gegeben, die die Sprache L erkennt.

Konstruktion eines Aufzählers A für L:

3.5 Rekursiv aufzählbare Sprachen

„$\Rightarrow$“: Sei eine TM M gegeben, die die Sprache L erkennt.

Konstruktion eines Aufzählers A für L:

Erster Versuch:

Aufzähler A für L

for $i = 1, 2, 3, \ldots$
    Simuliere M auf w_i.
    Wird w_i von M akzeptiert, so schreibe es auf das Ausgabeband.

3.5 Rekursiv aufzählbare Sprachen

„$\Rightarrow$“: Sei eine TM M gegeben, die die Sprache L erkennt.

Konstruktion eines Aufzählers A für L:

Erster Versuch:

Aufzähler A für L

```
for i = 1, 2, 3, ...
    Simuliere M auf w_i.
    Wird w_i von M akzeptiert, so schreibe es auf das Ausgabeband.
```

Problem: Gilt $w_i \notin L$, so terminiert M nicht notwendigerweise auf w_i und Wörter w_j mit $j > i$ werden nicht mehr erreicht.

Zweiter Versuch:

Aufzähler A **für** L

for $i = 1, 2, 3, \ldots$
    Simuliere jeweils i Schritte von M auf den Eingaben $w_1, \ldots, w_i$.
    Wird bei einer dieser Simulationen ein Wort w akzeptiert,
      so schreibe es auf das Ausgabeband.

3.5 Rekursiv aufzählbare Sprachen

Zweiter Versuch:

Aufzähler A für L

for $i = 1, 2, 3, \ldots$
    Simuliere jeweils i Schritte von M auf den Eingaben $w_1, \ldots, w_i$.
    Wird bei einer dieser Simulationen ein Wort w akzeptiert,
        so schreibe es auf das Ausgabeband.

Diese TM schreibt nur Wörter, die von M akzeptiert werden, auf das Ausgabeband.

Alle diese Wörter gehören zu L.

Zweiter Versuch:

Aufzähler A für L

for $i = 1, 2, 3, \ldots$

  Simuliere jeweils i Schritte von M auf den Eingaben $w_1, \ldots, w_i$.

  Wird bei einer dieser Simulationen ein Wort w akzeptiert, so schreibe es auf das Ausgabeband.

Diese TM schreibt nur Wörter, die von M akzeptiert werden, auf das Ausgabeband. Alle diese Wörter gehören zu L.

Für jedes Wort $w \in L$ gibt es ein t_w, sodass M die Eingabe w nach t_w vielen Schritten akzeptiert. Somit gibt A jedes Wort $w = w_j \in L$ für $i = \max\{t_w, j\}$ (also nach endlich vielen Schritten) aus. $\square$

Hilberts zehntes Problem

Eingabe: multivariates Polynom

z. B. $xy + x^2 + 10xy^2 - 2x^2y^3z - 7$

Frage: Besitzt das Polynom eine ganzzahlige Nullstelle?

Hilberts zehntes Problem

Eingabe: multivariates Polynom
z. B. $xy + x^2 + 10xy^2 - 2x^2y^3z - 7$

Frage: Besitzt das Polynom eine ganzzahlige Nullstelle?

Theorem 3.21

Hilberts zehntes Problem ist nicht entscheidbar.

Postsches Korrespondenzproblem (PKP)

Eingabe: endliche Menge Σ

endliche Menge $K = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ mit $x_i, y_i \in \Sigma^*$

3.6 Weitere nicht entscheidbare Probleme

Postsches Korrespondenzproblem (PKP)

Eingabe: endliche Menge Σ

endliche Menge $K = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ mit $x_i, y_i \in \Sigma^*$

Frage: Existieren $n \geq 1$ und Indizes $i_1, \ldots, i_n \in \{1, \ldots, k\}$,

sodass $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

Postsches Korrespondenzproblem (PKP)

Eingabe: endliche Menge Σ

endliche Menge $K = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ mit $x_i, y_i \in \Sigma^*$

Frage: Existieren $n \geq 1$ und Indizes $i_1, \ldots, i_n \in \{1, \ldots, k\}$,

sodass $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

Beispiel:

$$K = \left\{ \binom{110}{1}, \binom{1}{10}, \binom{0}{110}, \binom{0}{00} \right\}$$

Postsches Korrespondenzproblem (PKP)

Eingabe: endliche Menge Σ

endliche Menge $K = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ mit $x_i, y_i \in \Sigma^*$

Frage: Existieren $n \geq 1$ und Indizes $i_1, \ldots, i_n \in \{1, \ldots, k\}$,

sodass $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

Beispiel:

$$K = \left\{ \binom{110}{1}, \binom{1}{10}, \binom{0}{110}, \binom{0}{00} \right\}$$

Lösung: $n = 5$, $i_1 = 1$, $i_2 = 2$, $i_3 = 1$, $i_4 = 3$, $i_5 = 4$

$$\binom{110}{1}\binom{1}{10}\binom{110}{1}\binom{0}{110}\binom{0}{00} = \binom{110111000}{110111000}$$

3.6 Weitere nicht entscheidbare Probleme

Postsches Korrespondenzproblem (PKP)

Eingabe: endliche Menge Σ
endliche Menge $K = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ mit $x_i, y_i \in \Sigma^*$

Frage: Existieren $n \geq 1$ und Indizes $i_1, \ldots, i_n \in \{1, \ldots, k\}$,
sodass $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

Theorem 3.22

Das Postsche Korrespondenzproblem ist nicht entscheidbar.