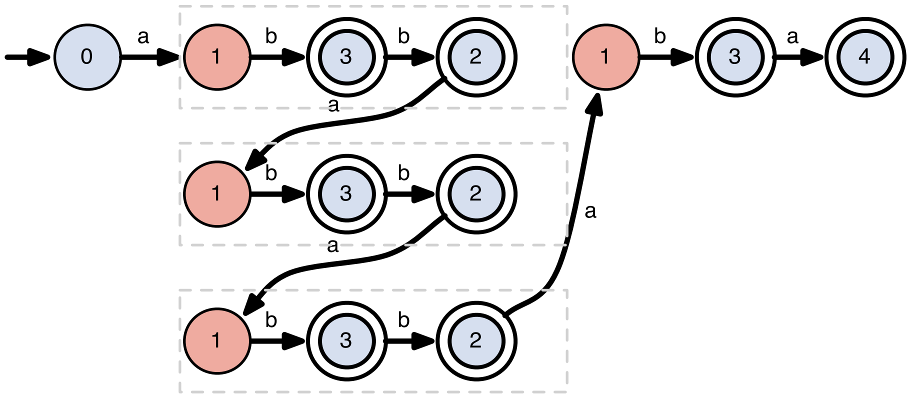
Berechenbarkeitstheorie

13. Vorlesung



Dr. Franziska Jahnke
Institut für Mathematische Logik und Grundlagenforschung
WWU Münster

WS 15/16

Nichtdeterministische TM (NTM)

Nichtdeterministische TM (NTM)

- TM definiert wie bisher, bis auf
  $\delta\colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$

Nichtdeterministische TM (NTM)

- TM definiert wie bisher, bis auf $\delta\colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$
- **Def. Berechnungsbaum:**
  - Baum dessen Knoten mit Konfigurationen beschriftet sind
  - Wurzel $=$ Startkonfiguration
  - Blätter $=$ akz./verwerfende Konfigurationen
  - Kinder eines Knotens mit Beschriftung C sind mit den Folgekonfigurationen von C beschriftet

Nichtdeterministische TM (NTM) [2]

- TM definiert wie bisher, bis auf
  $\delta \colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$
- **Def. Berechnungsbaum:**
  - Baum dessen Knoten mit Konfigurationen beschriftet sind
  - Wurzel $=$ Startkonfiguration
  - Blätter $=$ akz./verwerfende Konfigurationen
  - Kinder eines Knotens mit Beschriftung C sind mit den Folgekonfigurationen von C beschriftet
- Berechnungsbaum nicht notwendiger Weise endlich

Nichtdeterministische TM (NTM) [2]

- TM definiert wie bisher, bis auf
  $\delta\colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$
- **Def. Berechnungsbaum:**
  - Baum dessen Knoten mit Konfigurationen beschriftet sind
  - Wurzel = Startkonfiguration
  - Blätter = akz./verwerfende Konfigurationen
  - Kinder eines Knotens mit Beschriftung C sind mit den Folgekonfigurationen von C beschriftet
- Berechnungsbaum nicht notwendiger Weise endlich
- Berechnungsbaum hängt von TM **und** Eingabe ab

Nichtdeterministische TM (NTM) [2]

- TM definiert wie bisher, bis auf
  $\delta\colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$
- **Def. Berechnungsbaum:**
  - Baum dessen Knoten mit Konfigurationen beschriftet sind
  - Wurzel $=$ Startkonfiguration
  - Blätter $=$ akz./verwerfende Konfigurationen
  - Kinder eines Knotens mit Beschriftung C sind mit den Folgekonfigurationen von C beschriftet
- Berechnungsbaum nicht notwendiger Weise endlich
- Berechnungsbaum hängt von TM **und** Eingabe ab
- Lauf$=$Wurzel-Blatt Pfad im Berechnungsbaum
  bei (det.) TM sprechen wir von einem Berechnungspfad

Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

- Durchsuche den Berechnungsbaum nach akz. Konfiguration

Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

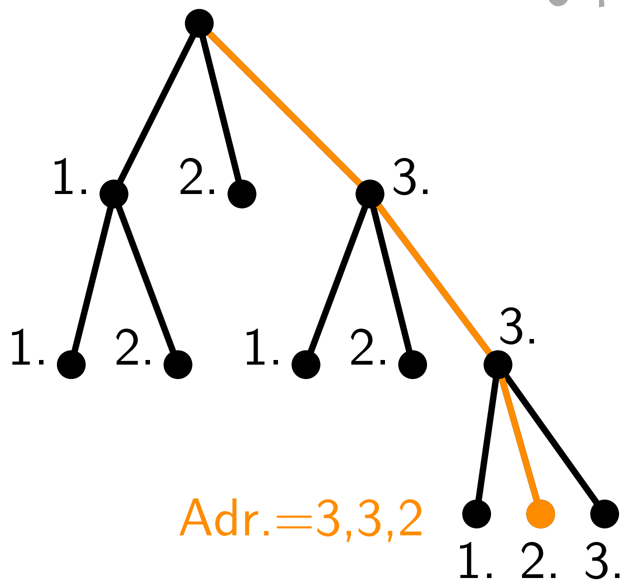
- Durchsuche den Berechnungsbaum nach akz. Konfiguration
- **Breitensuche**, da Baum unendliche Pfade haben kann

Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

- Durchsuche den Berechnungsbaum nach akz. Konfiguration
- **Breitensuche**, da Baum unendliche Pfade haben kann
- Ordne die ausgehenden Kanten (z.B. lexikographisch nach Folgekonfiguration)
- Adresse eines Knotens = Alternativen auf seinem Pfad

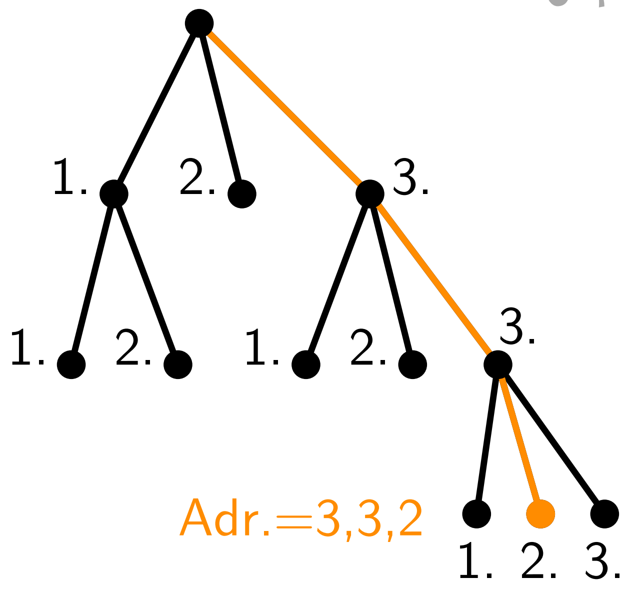


Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

- Durchsuche den Berechnungsbaum nach akz. Konfiguration
- **Breitensuche**, da Baum unendliche Pfade haben kann
- Ordne die ausgehenden Kanten (z.B. lexikographisch nach Folgekonfiguration)
- Adresse eines Knotens = Alternativen auf seinem Pfad
- Zähle auf einem speziellem Band der TM, alle Adressen der Länge nach auf (Details folgen)

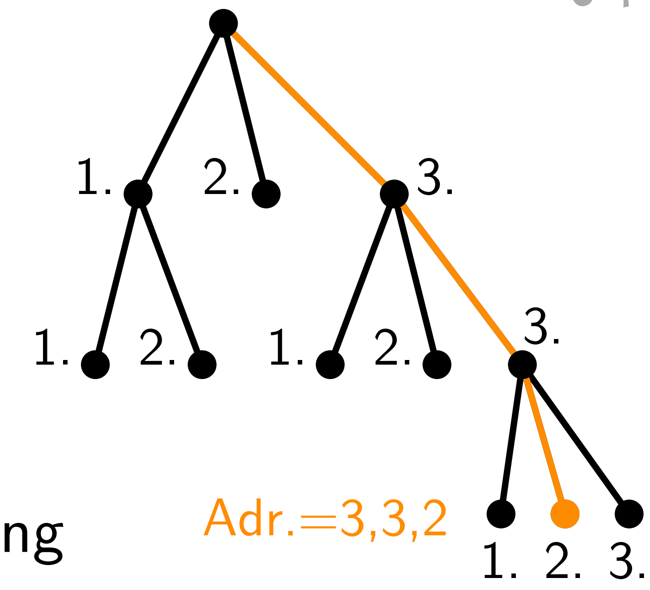


Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

- Durchsuche den Berechnungsbaum nach akz. Konfiguration
- **Breitensuche**, da Baum unendliche Pfade haben kann
- Ordne die ausgehenden Kanten (z.B. lexikographisch nach Folgekonfiguration)
- Adresse eines Knotens = Alternativen auf seinem Pfad
- Zähle auf einem speziellem Band der TM, alle Adressen der Länge nach auf (Details folgen)
- Simuliere auf einem anderem Band den Übergang zur Konfiguration der angegebenen Adresse

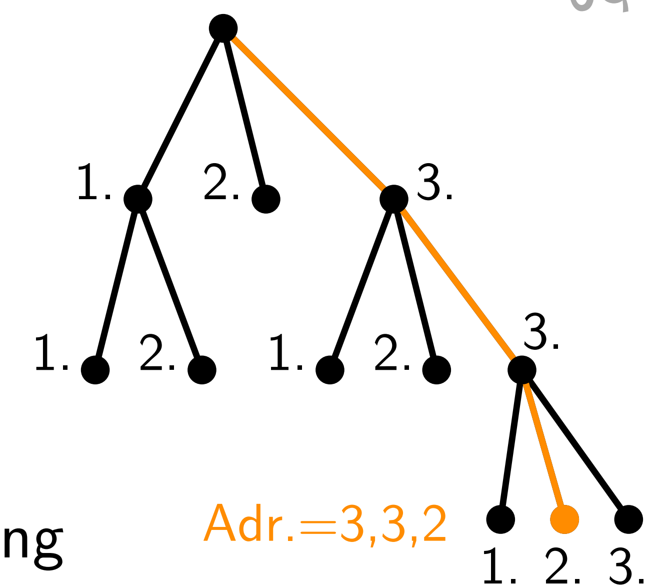


Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

- Durchsuche den Berechnungsbaum nach akz. Konfiguration
- **Breitensuche**, da Baum unendliche Pfade haben kann
- Ordne die ausgehenden Kanten (z.B. lexikographisch nach Folgekonfiguration)
- Adresse eines Knotens = Alternativen auf seinem Pfad
- Zähle auf einem speziellem Band der TM, alle Adressen der Länge nach auf (Details folgen)
- Simuliere auf einem anderem Band den Übergang zur Konfiguration der angegebenen Adresse



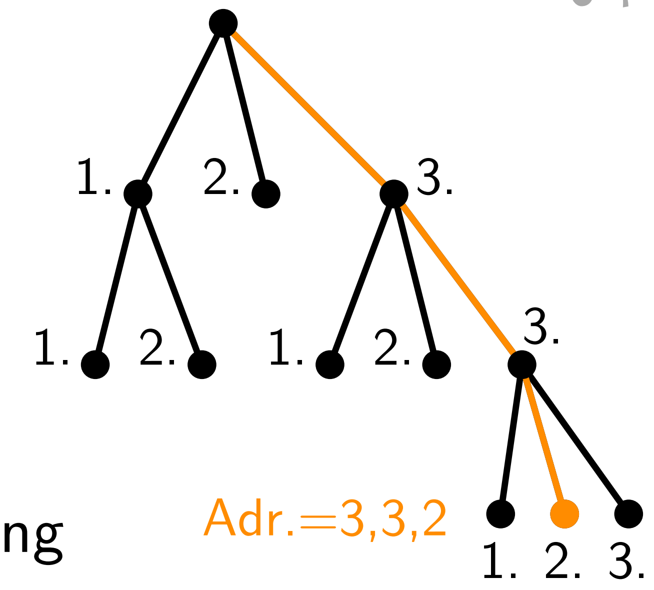
- Akzeptiere, wenn die berechnete Konf. akzeptierend, sonst weiter mit nächster Adresse

Satz 16

Jede NTM kann durch eine herkömmliche TM simuliert werden.

Beweis

- Durchsuche den Berechnungsbaum nach akz. Konfiguration
- **Breitensuche**, da Baum unendliche Pfade haben kann
- Ordne die ausgehenden Kanten (z.B. lexikographisch nach Folgekonfiguration)
- Adresse eines Knotens = Alternativen auf seinem Pfad
- Zähle auf einem speziellem Band der TM, alle Adressen der Länge nach auf (Details folgen)
- Simuliere auf einem anderem Band den Übergang zur Konfiguration der angegebenen Adresse



- Akzeptiere, wenn die berechnete Konf. akzeptierend, sonst weiter mit nächster Adresse

□

TM mit Ausgabe

TM mit Ausgabe

- TM M wie bisher mit einem besonderem Band, dem Ausgabeband

TM mit Ausgabe

- TM M wie bisher mit einem besonderem Band, dem Ausgabeband
- Ausgabeband ist nur beschriebbar

TM mit Ausgabe

- TM M wie bisher mit einem besonderem Band, dem Ausgabeband
- Ausgabeband ist nur beschriebbar
- Kopf des Ausgabebandes bewegt sich nur nach einem Schreibevorgang einen Schritt nach rechts, bleibt ansonsten stehen

TM mit Ausgabe

- TM M wie bisher mit einem besonderem Band, dem Ausgabeband
- Ausgabeband ist nur beschreibbar
- Kopf des Ausgabebandes bewegt sich nur nach einem Schreibevorgang einen Schritt nach rechts, bleibt ansonsten stehen
- M beschreibt folgende Funktion:

$$\boxed{\begin{aligned} f_M(x) := y, \ &\text{mit } M(\langle x\rangle) \text{ stoppt akzeptierend} \\ &\text{und auf dem Ausgabeband steht } \langle y\rangle \end{aligned}}$$

TM mit Ausgabe

- TM M wie bisher mit einem besonderem Band, dem Ausgabeband
- Ausgabeband ist nur beschriebbar
- Kopf des Ausgabebandes bewegt sich nur nach einem Schreibevorgang einen Schritt nach rechts, bleibt ansonsten stehen
- M beschreibt folgende Funktion:

$$\boxed{\begin{aligned} f_M(x) := y, \text{ mit } & M(\langle x \rangle) \text{ stoppt akzeptierend} \\ & \text{und auf dem Ausgabeband steht } \langle y \rangle \end{aligned}}$$

- verwirft M oder stoppt M nicht, dann ist die Funktion f_M für diese Eingabe undefiniert

Berechenbare Funktionen

Berechenbare Funktionen

Definition

Eine Funktion f heißt **berechenbar**, gdw. es eine TM M gibt mit $f_M \equiv f$.

Berechenbare Funktionen

Definition

Eine Funktion f heißt **berechenbar**, gdw. es eine TM M gibt mit $f_M \equiv f$.

- Beispiel einer berechenbaren Funktion:
  $f\colon \mathbf{N} \to \mathbf{N}$ und $f(x) = 2^x$
  (Turingmaschine die $\{\mathtt{a}^{2^n} \mid n \geq 0\}$ erkennt kann entsprechend abgewandelt werden)

Berechenbare Funktionen

Definition

Eine Funktion f heißt **berechenbar**, gdw. es eine TM M gibt mit $f_M \equiv f$.

- Beispiel einer berechenbaren Funktion:
  $f : \mathbf{N} \to \mathbf{N}$ und $f(x) = 2^x$
  (Turingmaschine die $\{\mathtt{a}^{2^n} \mid n \geq 0\}$ erkennt kann entsprechend abgewandelt werden)
- Rechtseindeutige Relationen heißen auch **partielle** Funktionen.

Berechenbare Funktionen

Definition

Eine Funktion f heißt **berechenbar**, gdw. es eine TM M gibt mit $f_M \equiv f$.

- Beispiel einer berechenbaren Funktion:
  $f : \mathbf{N} \to \mathbf{N}$ und $f(x) = 2^x$
  (Turingmaschine die $\{\mathsf{a}^{2^n} \mid n \geq 0\}$ erkennt kann entsprechend abgewandelt werden)
- Rechtseindeutige Relationen heißen auch **partielle** Funktionen.
- In diesem Sinne beschreibt jede TM eine partielle Funktion.

Berechenbare Funktionen

Definition

Eine Funktion f heißt **berechenbar**, gdw. es eine TM M gibt mit $f_M \equiv f$.

- Beispiel einer berechenbaren Funktion:
  $f : \mathbf{N} \to \mathbf{N}$ und $f(x) = 2^x$
  (Turingmaschine die $\{\mathsf{a}^{2^n} \mid n \geq 0\}$ erkennt kann entsprechend abgewandelt werden)
- Rechtseindeutige Relationen heißen auch **partielle** Funktionen.
- In diesem Sinne beschreibt jede TM eine partielle Funktion.
- Um Hervorzuheben, dass eine partielle Funktion überall definiert ist, nennen wir solche Funktionen **total**.

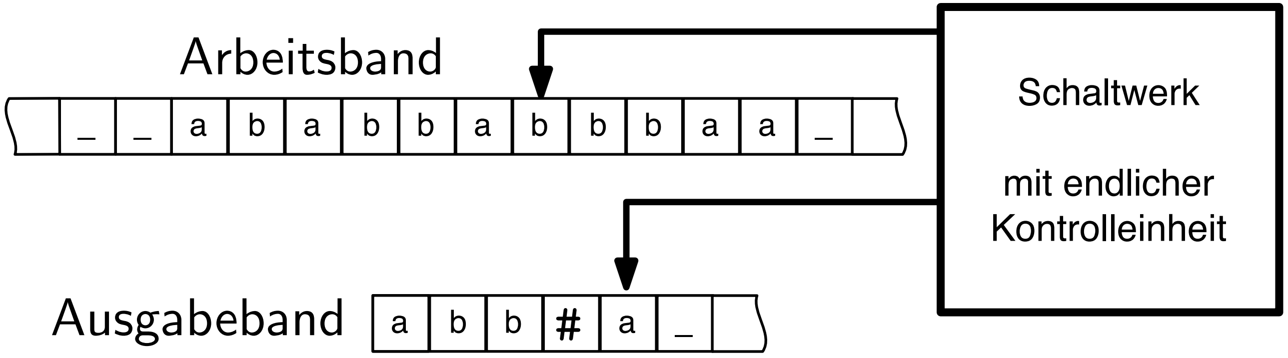
TM als Aufzähler

TM als Aufzähler

- ein **Aufzähler für $L \subseteq \Sigma^*$** ist eine 2-Band TM mit einem Ausgabeband und einem Arbeitsband

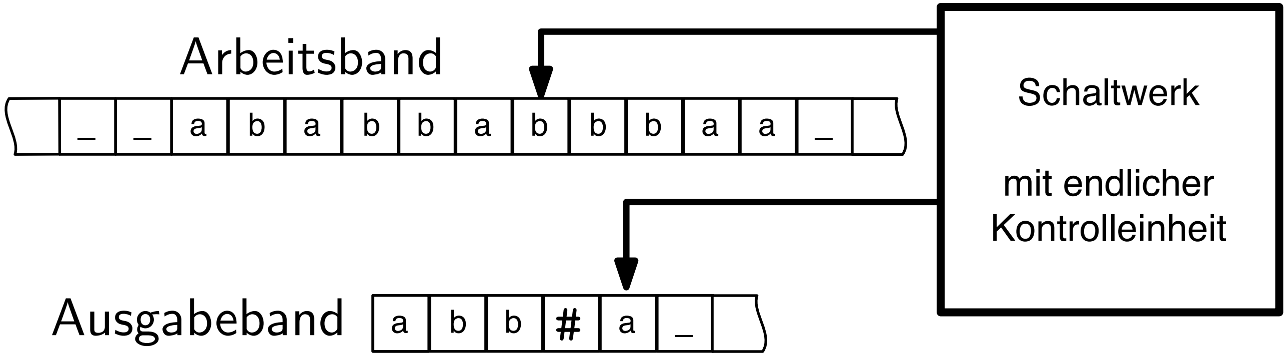
TM als Aufzähler

- ein **Aufzähler für $L \subseteq \Sigma^*$** ist eine 2-Band TM mit einem Ausgabeband und einem Arbeitsband



TM als Aufzähler

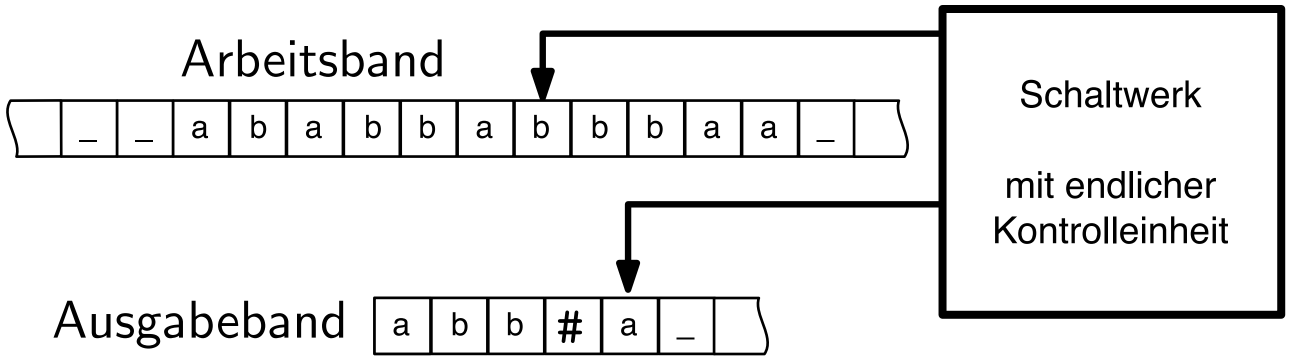
- ein **Aufzähler für $L \subseteq \Sigma^*$** ist eine 2-Band TM mit einem Ausgabeband und einem Arbeitsband



- Ausgabeband nicht lesbar, nach jedem Zugriff bewegt sich Kopf einen Schritt nach rechts, bleibt sonst stehen

TM als Aufzähler

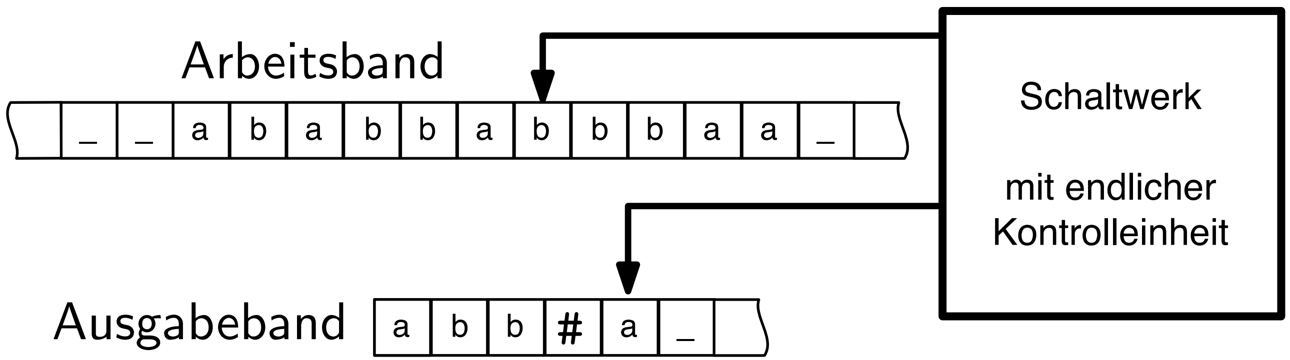
- ein **Aufzähler für $L \subseteq \Sigma^*$** ist eine 2-Band TM mit einem Ausgabeband und einem Arbeitsband



- Ausgabeband nicht lesbar, nach jedem Zugriff bewegt sich Kopf einen Schritt nach rechts, bleibt sonst stehen
- TM schreibt auf das Ausgabeband $w_1\#w_2\#w_3\cdots$, wobei $\# \notin \Sigma$ und $L = \bigcup_i w_i$

TM als Aufzähler

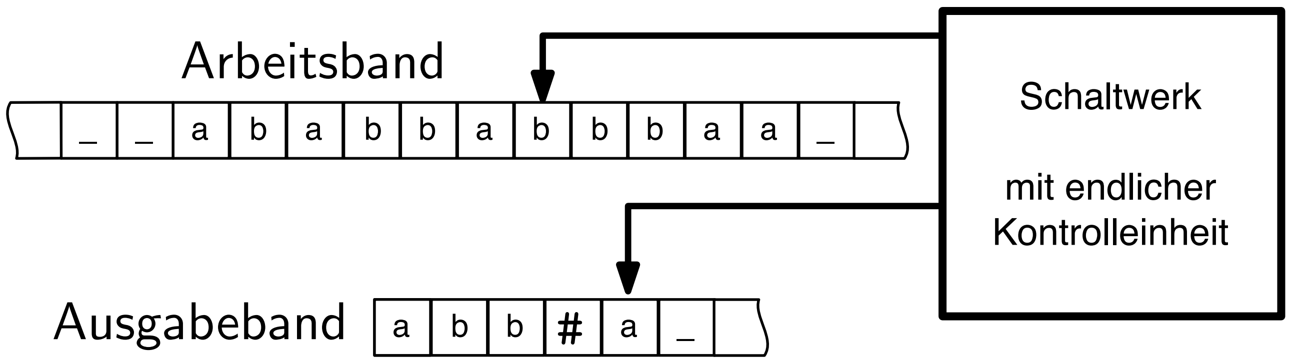
- ein **Aufzähler für $L \subseteq \Sigma^*$** ist eine 2-Band TM mit einem Ausgabeband und einem Arbeitsband



- Ausgabeband nicht lesbar, nach jedem Zugriff bewegt sich Kopf einen Schritt nach rechts, bleibt sonst stehen
- TM schreibt auf das Ausgabeband $w_1 \# w_2 \# w_3 \cdots$, wobei $\# \notin \Sigma$ und $L = \bigcup_i w_i$
- Achtung: Wiederholungen der Wörter aus L erlaubt

TM als Aufzähler

- ein **Aufzähler für $L \subseteq \Sigma^*$** ist eine 2-Band TM mit einem Ausgabeband und einem Arbeitsband



- Ausgabeband nicht lesbar, nach jedem Zugriff bewegt sich Kopf einen Schritt nach rechts, bleibt sonst stehen
- TM schreibt auf das Ausgabeband $w_1\#w_2\#w_3\cdots$, wobei $\# \notin \Sigma$ und $L = \bigcup_i w_i$
- Achtung: Wiederholungen der Wörter aus L erlaubt
- Sprachen, für die es einen Aufzähler gibt, heißen **aufzählbar**

Beispiel $L = \Sigma^* = \{0, 1\}^*$

Beispiel $L = \Sigma^* = \{0, 1\}^*$

- Ausgabeband soll $\#0\#1\#00\#01\#10\#\ldots$ enthalten

Beispiel $L = \Sigma^* = \{0, 1\}^*$

- Ausgabeband soll $\#0\#1\#00\#01\#10\#\ldots$ enthalten
- Ablauf: Zähle zuerst Σ^0 auf, dann Σ^1, Σ^2, usw.

Beispiel $L = \Sigma^* = \{0, 1\}^*$

- Ausgabeband soll $\#0\#1\#00\#01\#10\#\ldots$ enthalten
- Ablauf: Zähle zuerst Σ^0 auf, dann Σ^1, Σ^2, usw.
- Modul A: Zähle Σ^i auf (Eingabe Arbeitsband 0^i)
  1. Kopiere Arbeitsband$+$ $\#$ auf Ausgabe
  2. Wenn auf Arbeitsband 1^* steht, verlasse Modul
  3. Addiere$+1$ auf Binärzahl auf Arbeitsband
  4. Gehe zu 1.

Beispiel $L = \Sigma^* = \{0, 1\}^*$

- Ausgabeband soll $\#0\#1\#00\#01\#10\# \ldots$ enthalten
- Ablauf: Zähle zuerst Σ^0 auf, dann Σ^1, Σ^2, usw.
- Modul A: Zähle Σ^i auf (Eingabe Arbeitsband 0^i)
  1. Kopiere Arbeitsband$+$ $\#$ auf Ausgabe
  2. Wenn auf Arbeitsband 1^* steht, verlasse Modul
  3. Addiere$+1$ auf Binärzahl auf Arbeitsband
  4. Gehe zu 1.
- Modul: Addiere $+1$
  1. Gehe nach rechts
  2. Wenn Kopf auf 1 ersetze 1 durch 0, gehe einen Schritt nach links und wiederhole 2.
  3. Wenn 0 ersetze 0 durch 1 und stoppe Modul

Beispiel $L = \Sigma^* = \{0, 1\}^*$

- Ausgabeband soll $\#0\#1\#00\#01\#10\#\ldots$ enthalten
- Ablauf: Zähle zuerst Σ^0 auf, dann Σ^1, Σ^2, usw.
- Modul A: Zähle Σ^i auf (Eingabe Arbeitsband 0^i)
    1. Kopiere Arbeitsband$+\ \#$ auf Ausgabe
    2. Wenn auf Arbeitsband 1^* steht, verlasse Modul
    3. Addiere$+1$ auf Binärzahl auf Arbeitsband
    4. Gehe zu 1.
- Modul: Addiere $+1$
    1. Gehe nach rechts
    2. Wenn Kopf auf 1 ersetze 1 durch 0, gehe einen Schritt nach links und wiederhole 2.
    3. Wenn 0 ersetze 0 durch 1 und stoppe Modul
- Hauptmodul:
    1. Schreibe $\#0\#1\#$ auf das Ausgabe- und 1 aufs Arbeitsband
    2. Ersetze alle 1en durch 0en plus Extra-0
    3. Führe Modul A aus, dann wiederhole ab 2.

Satz 17

L ist aufzählbar $\iff$ L ist erkennbar

Satz 17

L ist aufzählbar $\iff$ L ist erkennbar

Beweis

1. Teil L ist aufzählbar $\Rightarrow$ L ist erkennbar

Satz 17

L ist aufzählbar $\iff$ L ist erkennbar

Beweis

1. Teil L ist aufzählbar $\Rightarrow$ L ist erkennbar

- Sein A ein Aufzähler für L

Satz 17

L ist aufzählbar $\iff$ L ist erkennbar

Beweis

1. Teil L ist aufzählbar $\Rightarrow$ L ist erkennbar

- Sein A ein Aufzähler für L
- Wir konstruieren TM M, welche L erkennt (d.h. alle $w \in L$ werden akzeptiert)

Satz 17

L ist aufzählbar $\iff$ L ist erkennbar

Beweis

1. Teil L ist aufzählbar $\Rightarrow$ L ist erkennbar

- Sein A ein Aufzähler für L
- Wir konstruieren TM M, welche L erkennt (d.h. alle $w \in L$ werden akzeptiert)
- M simuliert A auf zwei Extra-Bändern (Unterprogramm)

Satz 17

L ist aufzählbar $\iff$ L ist erkennbar

Beweis

1. Teil L ist aufzählbar $\Rightarrow$ L ist erkennbar

- Sein A ein Aufzähler für L
- Wir konstruieren TM M, welche L erkennt (d.h. alle $w \in L$ werden akzeptiert)
- M simuliert A auf zwei Extra-Bändern (Unterprogramm)
- Falls ein $\#$ aufs Ausgabeband geschrieben wurde, vergleiche Wort zwischen den letzten beiden $\#$ mit der Eingabe

Satz 17

L ist aufzählbar $\iff$ L ist erkennbar

Beweis

1. Teil L ist aufzählbar $\Rightarrow$ L ist erkennbar

- Sein A ein Aufzähler für L
- Wir konstruieren TM M, welche L erkennt (d.h. alle $w \in L$ werden akzeptiert)
- M simuliert A auf zwei Extra-Bändern (Unterprogramm)
- Falls ein $\#$ aufs Ausgabeband geschrieben wurde, vergleiche Wort zwischen den letzten beiden $\#$ mit der Eingabe
- Falls beide Wörter identisch sind, folgt dass $w \in L$, dann akzeptiere

Satz 17

$$L \text{ ist aufzählbar} \iff L \text{ ist erkennbar}$$

Beweis

1. Teil L ist aufzählbar $\Rightarrow L$ ist erkennbar

- Sein A ein Aufzähler für L
- Wir konstruieren TM M, welche L erkennt (d.h. alle $w \in L$ werden akzeptiert)
- M simuliert A auf zwei Extra-Bändern (Unterprogramm)
- Falls ein $\#$ aufs Ausgabeband geschrieben wurde, vergleiche Wort zwischen den letzten beiden $\#$ mit der Eingabe
- Falls beide Wörter identisch sind, folgt dass $w \in L$, dann akzeptiere
- Ansonsten fahre mit Simulation von A fort

Satz 17

L ist aufzählbar $\iff$ L ist erkennbar

Beweis

1. Teil L ist aufzählbar $\Rightarrow$ L ist erkennbar

- Sein A ein Aufzähler für L
- Wir konstruieren TM M, welche L erkennt (d.h. alle $w \in L$ werden akzeptiert)
- M simuliert A auf zwei Extra-Bändern (Unterprogramm)
- Falls ein $\#$ aufs Ausgabeband geschrieben wurde, vergleiche Wort zwischen den letzten beiden $\#$ mit der Eingabe
- Falls beide Wörter identisch sind, folgt dass $w \in L$, dann akzeptiere
- Ansonsten fahre mit Simulation von A fort
- $M(w)$ zykelt für alle $w \notin L$, aber das ist erlaubt

2. Teil L ist erkennbar $\Rightarrow$ L ist aufzählbar

2. Teil L ist erkennbar $\Rightarrow$ L ist aufzählbar

- Sei M TM die L erkennt, wir konstruieren Aufzähler A für L

2. Teil L ist erkennbar $\Rightarrow$ L ist aufzählbar

- Sei M TM die L erkennt, wir konstruieren Aufzähler A für L
- Zähle Σ^* als $s_1\#s_2\#s_3\#\ldots$ auf Extra-Band auf (mittels Aufzähler, so weit wie es benötigt wird)

2. Teil L ist erkennbar $\Rightarrow$ L ist aufzählbar

- Sei M TM die L erkennt, wir konstruieren Aufzähler A für L
- Zähle Σ^* als $s_1\#s_2\#s_3\#\ldots$ auf Extra-Band auf (mittels Aufzähler, so weit wie es benötigt wird)

Algorithm 3: Aufzähler für L

```
for i = 1 to ∞ do
    for j = 1 to i do
        Simuliere M(s_j) i Schritte lang;
        Bei Erfolg schreibe s_j# aufs Ausgabeband;
    end
end
```

2. Teil L ist erkennbar $\Rightarrow$ L ist aufzählbar

- Sei M TM die L erkennt, wir konstruieren Aufzähler A für L
- Zähle Σ^* als $s_1\#s_2\#s_3\#\ldots$ auf Extra-Band auf (mittels Aufzähler, so weit wie es benötigt wird)

Algorithm 3: Aufzähler für L

```
for i = 1 to ∞ do
    for j = 1 to i do
        Simuliere M(s_j) i Schritte lang;
        Bei Erfolg schreibe s_j# aufs Ausgabeband;
    end
end
```

- Algorithmus kann mittels Mehrspurtechnik auf einem Arbeits- und Ausgabeband realisiert werden

2. Teil L ist erkennbar $\Rightarrow$ L ist aufzählbar

- Sei M TM die L erkennt, wir konstruieren Aufzähler A für L
- Zähle Σ^* als $s_1\#s_2\#s_3\#\ldots$ auf Extra-Band auf (mittels Aufzähler, so weit wie es benötigt wird)

Algorithm 3: Aufzähler für L

for $i = 1$ **to** ∞ **do**
    for $j = 1$ **to** i **do**
        Simuliere $M(s_j)$ i Schritte lang;
        Bei Erfolg schreibe $s_j\#$ aufs Ausgabeband;
    end
end

- Algorithmus kann mittels Mehrspurtechnik auf einem Arbeits- und Ausgabeband realisiert werden
- alle $s_j \in L$ werden ab einem bestimmten i erkannt, und erscheinen dann auf dem Ausgabeband

2. Teil L ist erkennbar $\Rightarrow$ L ist aufzählbar

- Sei M TM die L erkennt, wir konstruieren Aufzähler A für L
- Zähle Σ^* als $s_1\#s_2\#s_3\#\ldots$ auf Extra-Band auf (mittels Aufzähler, so weit wie es benötigt wird)

Algorithm 3: Aufzähler für L

```
for i = 1 to ∞ do
    for j = 1 to i do
        Simuliere M(s_j) i Schritte lang;
        Bei Erfolg schreibe s_j# aufs Ausgabeband;
    end
end
```

- Algorithmus kann mittels Mehrspurtechnik auf einem Arbeits- und Ausgabeband realisiert werden
- alle $s_j \in L$ werden ab einem bestimmten i erkannt, und erscheinen dann auf dem Ausgabeband
- $s_j \notin L$ wird nie ausgegeben $\square$

Die Church-Turing-These

Die Church-Turing-These

- **Frage:** Was ist eine sinnvolle Definition für den Begriff *berechenbar*

Die Church-Turing-These

- **Frage:** Was ist eine sinnvolle Definition für den Begriff *berechenbar*
- 1900 formulierte D. Hilbert auf dem 2. Mathematischen Kongress in Paris 23 offene Probleme

Die Church-Turing-These

- **Frage:** Was ist eine sinnvolle Definition für den Begriff *berechenbar*
- 1900 formulierte D. Hilbert auf dem 2. Mathematischen Kongress in Paris 23 offene Probleme
- **10. Problem:** Finde ein Verfahren das feststellt ob ein Polynom eine ganzzahlige Nullstelle hat? (Lösung von Diophantischen Gleichungen)

Die Church-Turing-These

- **Frage:** Was ist eine sinnvolle Definition für den Begriff *berechenbar*
- 1900 formulierte D. Hilbert auf dem 2. Mathematischen Kongress in Paris 23 offene Probleme
- **10. Problem:** Finde ein Verfahren das feststellt ob ein Polynom eine ganzzahlige Nullstelle hat? (Lösung von Diophantischen Gleichungen)
- konkret fragte Hilbert:

> *... Verfahren angeben, nach welchem sich mittelst einer endlichen Anzahl von Operationen entscheiden läßt, ob die Gleichung in ganzen rationalen Zahlen lösbar ist.*

Die Church-Turing-These

- **Frage:** Was ist eine sinnvolle Definition für den Begriff *berechenbar*
- 1900 formulierte D. Hilbert auf dem 2. Mathematischen Kongress in Paris 23 offene Probleme
- **10. Problem:** Finde ein Verfahren das feststellt ob ein Polynom eine ganzzahlige Nullstelle hat? (Lösung von Diophantischen Gleichungen)
- konkret fragte Hilbert:

> *... Verfahren angeben, nach welchem sich mittelst einer endlichen Anzahl von Operationen entscheiden läßt, ob die Gleichung in ganzen rationalen Zahlen lösbar ist.*

Die Church-Turing-These

- **Frage:** Was ist eine sinnvolle Definition für den Begriff *berechenbar*
- 1900 formulierte D. Hilbert auf dem 2. Mathematischen Kongress in Paris 23 offene Probleme
- **10. Problem:** Finde ein Verfahren das feststellt ob ein Polynom eine ganzzahlige Nullstelle hat? (Lösung von Diophantischen Gleichungen)
- konkret fragte Hilbert:

  > *... Verfahren angeben, nach welchem sich mittelst einer endlichen Anzahl von Operationen entscheiden läßt, ob die Gleichung in ganzen rationalen Zahlen lösbar ist.*

- In der heutigen Terminologie fragt Hilbert nach der Existenz eines Algorithmus (endliche Folge von elementaren Befehlen).

Die Church-Turing-These

- **Frage:** Was ist eine sinnvolle Definition für den Begriff *berechenbar*
- 1900 formulierte D. Hilbert auf dem 2. Mathematischen Kongress in Paris 23 offene Probleme
- **10. Problem:** Finde ein Verfahren das feststellt ob ein Polynom eine ganzzahlige Nullstelle hat? (Lösung von Diophantischen Gleichungen)
- konkret fragte Hilbert:

  > ... *Verfahren angeben, nach welchem sich mittelst einer endlichen Anzahl von Operationen entscheiden läßt, ob die Gleichung in ganzen rationalen Zahlen lösbar ist.*

- In der heutigen Terminologie fragt Hilbert nach der Existenz eines Algorithmus (endliche Folge von elementaren Befehlen).
- Was heißt "Folge von elementaren Befehlen" genau?

Church-Turing-These

Wir verstehen unter (intuitiv) berechenbar alle die Funktionen die eine Turingmaschine berechnen kann.

Church-Turing-These

Wir verstehen unter (intuitiv) berechenbar alle die Funktionen die eine Turingmaschine berechnen kann.

Rechtfertigung

Church-Turing-These

Wir verstehen unter (intuitiv) berechenbar alle die Funktionen die eine Turingmaschine berechnen kann.

Rechtfertigung

- es gibt sehr viele alternative Berechnungsmodelle die zur TM äquivalent sind (Registermaschine, λ-Kalkül, μ-rekursive Funktionen, ...)

Church-Turing-These

Wir verstehen unter (intuitiv) berechenbar alle die Funktionen die eine Turingmaschine berechnen kann.

Rechtfertigung

- es gibt sehr viele alternative Berechnungsmodelle die zur TM äquivalent sind (Registermaschine, λ-Kalkül, μ-rekursive Funktionen, ...)
- jede Art von sinnvollen Pseudocode kann in ein TM-Programm überführt werden

Church-Turing-These

Wir verstehen unter (intuitiv) berechenbar alle die Funktionen die eine Turingmaschine berechnen kann.

Rechtfertigung

- es gibt sehr viele alternative Berechnungsmodelle die zur TM äquivalent sind (Registermaschine, λ-Kalkül, μ-rekursive Funktionen, ...)
- jede Art von sinnvollen Pseudocode kann in ein TM-Programm überführt werden
- TM äquivalente Berechnungsmodelle brauchen nur sehr einfache Annahmen: ihre endlichen Operationen können zuverlässig, wiederholbar und verifizierbar durchgeführt werden

Church-Turing-These

Wir verstehen unter (intuitiv) berechenbar alle die Funktionen die eine Turingmaschine berechnen kann.

Rechtfertigung

- es gibt sehr viele alternative Berechnungsmodelle die zur TM äquivalent sind (Registermaschine, λ-Kalkül, μ-rekursive Funktionen, ...)
- jede Art von sinnvollen Pseudocode kann in ein TM-Programm überführt werden
- TM äquivalente Berechnungsmodelle brauchen nur sehr einfache Annahmen: ihre endlichen Operationen können zuverlässig, wiederholbar und verifizierbar durchgeführt werden

  gestützt durch die physikalische Wahrnehmung von maschinellem Rechnen

Church-Turing-These

Wir verstehen unter (intuitiv) berechenbar alle die Funktionen die eine Turingmaschine berechnen kann.

Rechtfertigung

- es gibt sehr viele alternative Berechnungsmodelle die zur TM äquivalent sind (Registermaschine, λ-Kalkül, μ-rekursive Funktionen, ...)
- jede Art von sinnvollen Pseudocode kann in ein TM-Programm überführt werden
- TM äquivalente Berechnungsmodelle brauchen nur sehr einfache Annahmen: ihre endlichen Operationen können zuverlässig, wiederholbar und verifizierbar durchgeführt werden

  gestützt durch die physikalische Wahrnehmung von maschinellem Rechnen
- Anmerkung: Hilberts 10. Problem wurde 1970 von Matiyasevich negativ beantwortet

Abzählbarkeit

Abzählbarkeit

- zwei Mengen heißen **gleichmächtig**, wenn es eine Bijektion zwischen ihnen gibt

Abzählbarkeit

- zwei Mengen heißen **gleichmächtig**, wenn es eine Bijektion zwischen ihnen gibt

Definition

Eine Menge X heißt **abzählbar**, gdw.

1. X ist endlich, oder
2. X und die natürlichen Zahlen sind gleichmächtig.

Abzählbarkeit

- zwei Mengen heißen **gleichmächtig**, wenn es eine Bijektion zwischen ihnen gibt

Definition

Eine Menge X heißt **abzählbar**, gdw.

1. X ist endlich, oder
2. X und die natürlichen Zahlen sind gleichmächtig.

Beispiel 1: $\mathbf{Z}$ ist abzählbar

Abzählbarkeit

- zwei Mengen heißen **gleichmächtig**, wenn es eine Bijektion zwischen ihnen gibt

Definition

Eine Menge X heißt **abzählbar**, gdw.

1. X ist endlich, oder
2. X und die natürlichen Zahlen sind gleichmächtig.

Beispiel 1: $\mathbf{Z}$ ist abzählbar

- wir müssen eine Bijektion zwischen $\mathbf{N}$ und $\mathbf{Z}$ finden:

Abzählbarkeit

- zwei Mengen heißen **gleichmächtig**, wenn es eine Bijektion zwischen ihnen gibt

Definition

Eine Menge X heißt **abzählbar**, gdw.

1. X ist endlich, oder
2. X und die natürlichen Zahlen sind gleichmächtig.

Beispiel 1: $\mathbf{Z}$ ist abzählbar

- wir müssen eine Bijektion zwischen $\mathbf{N}$ und $\mathbf{Z}$ finden:

$$f(x) := \begin{cases} 2x & x \geq 0 \\ |2x| - 1 & x < 0 \end{cases}$$

Abzählbarkeit

- zwei Mengen heißen **gleichmächtig**, wenn es eine Bijektion zwischen ihnen gibt

Definition

Eine Menge X heißt **abzählbar**, gdw.

1. X ist endlich, oder
2. X und die natürlichen Zahlen sind gleichmächtig.

Beispiel 1: $\mathbf{Z}$ ist abzählbar

- wir müssen eine Bijektion zwischen $\mathbf{N}$ und $\mathbf{Z}$ finden:

$$f(x) := \begin{cases} 2x & x \geq 0 \\ |2x| - 1 & x < 0 \end{cases}$$

- $f(0) = 0, f(-1) = 1, f(1) = 2, f(-2) = 3, \ldots$

Abzählbarkeit

- zwei Mengen heißen **gleichmächtig**, wenn es eine Bijektion zwischen ihnen gibt

Definition

Eine Menge X heißt **abzählbar**, gdw.

1. X ist endlich, oder
2. X und die natürlichen Zahlen sind gleichmächtig.

Beispiel 1: $\mathbf{Z}$ ist abzählbar

- wir müssen eine Bijektion zwischen $\mathbf{N}$ und $\mathbf{Z}$ finden:

$$f(x) := \begin{cases} 2x & x \geq 0 \\ |2x| - 1 & x < 0 \end{cases}$$

- $f(0) = 0, f(-1) = 1, f(1) = 2, f(-2) = 3, \ldots$
- f ist Bijektion

Beispiel 2: $\mathbf{N} \setminus \{0\}$ ist abzählbar

Beispiel 2: $\mathbf{N} \setminus \{0\}$ ist abzählbar

$f(x) := x - 1$ ist Bijektion zwischen $\mathbf{N} \setminus \{0\}$ und $\mathbf{N}$

Beispiel 2: $\mathbf{N} \setminus \{0\}$ ist abzählbar

$f(x) := x - 1$ ist Bijektion zwischen $\mathbf{N} \setminus \{0\}$ und $\mathbf{N}$

Beispiel 3: $\{0, 1\}^*$ ist abzählbar

Beispiel 2: $\mathbf{N} \setminus \{0\}$ ist abzählbar

$f(x) := x - 1$ ist Bijektion zwischen $\mathbf{N} \setminus \{0\}$ und $\mathbf{N}$

Beispiel 3: $\{0, 1\}^*$ ist abzählbar

Gleichmächtigkeit ist transitiv, deshalb genügt es Bijektion $f \colon \{0, 1\}^* \to \mathbf{N} \setminus \{0\}$ zu finden

Beispiel 2: $\mathbf{N} \setminus \{0\}$ ist abzählbar

$f(x) := x - 1$ ist Bijektion zwischen $\mathbf{N} \setminus \{0\}$ und $\mathbf{N}$

Beispiel 3: $\{0, 1\}^*$ ist abzählbar

Gleichmächtigkeit ist transitiv, deshalb genügt es Bijektion $f\colon \{0,1\}^* \to \mathbf{N} \setminus \{0\}$ zu finden

$f(w) := \mathrm{bin}(1 \circ w)$

Beispiel 2: $\mathbf{N} \setminus \{0\}$ ist abzählbar

$f(x) := x - 1$ ist Bijektion zwischen $\mathbf{N} \setminus \{0\}$ und $\mathbf{N}$

Beispiel 3: $\{0, 1\}^*$ ist abzählbar

Gleichmächtigkeit ist transitiv, deshalb genügt es Bijektion $f\colon \{0,1\}^* \to \mathbf{N} \setminus \{0\}$ zu finden

$f(w) := \text{bin}(1 \circ w)$ Wert des $\{0,1\}^*$ Wortes als Binärzahl

Beispiel 2: $\mathbf{N} \setminus \{0\}$ ist abzählbar

$f(x) := x - 1$ ist Bijektion zwischen $\mathbf{N} \setminus \{0\}$ und $\mathbf{N}$

Beispiel 3: $\{0, 1\}^*$ ist abzählbar

Gleichmächtigkeit ist transitiv, deshalb genügt es Bijektion $f\colon \{0,1\}^* \to \mathbf{N} \setminus \{0\}$ zu finden

$f(w) := \text{bin}(1 \circ w)$ Wert des $\{0, 1\}^*$ Wortes als Binärzahl

Beispiel 4: $\mathbf{Q}_+$ ist abzählbar

Beispiel 2: $\mathbf{N} \setminus \{0\}$ ist abzählbar

$f(x) := x - 1$ ist Bijektion zwischen $\mathbf{N} \setminus \{0\}$ und $\mathbf{N}$

Beispiel 3: $\{0, 1\}^*$ ist abzählbar

Gleichmächtigkeit ist transitiv, deshalb genügt es Bijektion $f\colon \{0,1\}^* \to \mathbf{N} \setminus \{0\}$ zu finden

$f(w) := \mathrm{bin}(1 \circ w)$ Wert des $\{0,1\}^*$ Wortes als Binärzahl

Beispiel 4: $\mathbf{Q}_+$ ist abzählbar

Erster Schritt: Ordne alle Brüche in Tabelle T an

	1	2	3	4	
1	1/1	1/2	1/3	1/4	
2	2/1	2/2	2/3	2/4	...
3	3/1	3/2	3/3	3/4	
		⋮			

Gegendiagonale k: alle Zellen $T[i, j]$ mit $i + j = k$

Beispiel 2: $\mathbf{N} \setminus \{0\}$ ist abzählbar

$f(x) := x - 1$ ist Bijektion zwischen $\mathbf{N} \setminus \{0\}$ und $\mathbf{N}$

Beispiel 3: $\{0,1\}^*$ ist abzählbar

Gleichmächtigkeit ist transitiv, deshalb genügt es Bijektion $f\colon \{0,1\}^* \to \mathbf{N} \setminus \{0\}$ zu finden

$f(w) := \mathrm{bin}(1 \circ w)$ Wert des $\{0,1\}^*$ Wortes als Binärzahl

Beispiel 4: $\mathbf{Q}_+$ ist abzählbar

Erster Schritt: Ordne alle Brüche in Tabelle T an

	1	2	3	4	
1	1/1	1/2	1/3	1/4	
2	2/1	2/2	2/3	2/4	$\cdots$
3	3/1	3/2	3/3	3/4	
			$\vdots$		

Gegendiagonale 4

Gegendiagonale k: alle Zellen $T[i,j]$ mit $i + j = k$

Beispiel 2: $\mathbf{N} \setminus \{0\}$ ist abzählbar

$f(x) := x - 1$ ist Bijektion zwischen $\mathbf{N} \setminus \{0\}$ und $\mathbf{N}$

Beispiel 3: $\{0, 1\}^*$ ist abzählbar

Gleichmächtigkeit ist transitiv, deshalb genügt es Bijektion $f\colon \{0,1\}^* \to \mathbf{N} \setminus \{0\}$ zu finden

$f(w) := \mathrm{bin}(1 \circ w)$ Wert des $\{0, 1\}^*$ Wortes als Binärzahl

Beispiel 4: $\mathbf{Q}_+$ ist abzählbar

Erster Schritt: Ordne alle Brüche in Tabelle T an

	1	2	3	4	
1	1/1	1/2	1/3	1/4	
2	2/1	2/2	2/3	2/4	...
3	3/1	3/2	3/3	3/4	
			⋮		

Gegendiagonale 4

Gegendiagonale k: alle Zellen $T[i, j]$ mit $i + j = k$

Idee: Nummeriere alle Gegendiagonalen in aufsteigender Reihenfolge

Beispiel 2: $\mathbf{N} \setminus \{0\}$ ist abzählbar

$f(x) := x - 1$ ist Bijektion zwischen $\mathbf{N} \setminus \{0\}$ und $\mathbf{N}$

Beispiel 3: $\{0, 1\}^*$ ist abzählbar

Gleichmächtigkeit ist transitiv, deshalb genügt es Bijektion $f\colon \{0, 1\}^* \to \mathbf{N} \setminus \{0\}$ zu finden

$f(w) := \mathrm{bin}(1 \circ w)$ Wert des $\{0, 1\}^*$ Wortes als Binärzahl

Beispiel 4: $\mathbf{Q}_+$ ist abzählbar

Erster Schritt: Ordne alle Brüche in Tabelle T an

	1	2	3	4	
1	1/1	1/2	1/3	1/4	
2	2/1	2/2	2/3	2/4	. . .
3	3/1	3/2	3/3	3/4	

$\vdots$

Gegendiagonale k: alle Zellen $T[i, j]$ mit $i + j = k$

Idee: Nummeriere alle Gegendiagonalen in aufsteigender Reihenfolge

- Konstruiere Liste L wie folgt

- Konstruiere Liste L wie folgt

  $L = \emptyset$
  for $i = 2$ **to** ∞
  Füge Gegendiagonale i zu L hinzu
  endfor

- Konstruiere Liste L wie folgt

  $L = \emptyset$
  for $i = 2$ **to** ∞
   Füge Gegendiagonale i zu L hinzu
   endfor

- $L = \{\frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{1}{3}, \frac{2}{2}, \frac{3}{1}, \frac{1}{4}, \frac{2}{3}, \frac{3}{2}, \frac{4}{1}, \frac{1}{5}, \frac{2}{4}, \frac{3}{3}, \frac{4}{2}, \frac{5}{1}, \ldots$

- Konstruiere Liste L wie folgt

  $L = \emptyset$
  for $i = 2$ **to** ∞
  Füge Gegendiagonale i zu L hinzu
  endfor

- $L = \{\frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{1}{3}, \frac{2}{2}, \frac{3}{1}, \frac{1}{4}, \frac{2}{3}, \frac{3}{2}, \frac{4}{1}, \frac{1}{5}, \frac{2}{4}, \frac{3}{3}, \frac{4}{2}, \frac{5}{1}, \dots$
- Konstruiere L' von L durch das Streichen doppelter Einträge

- Konstruiere Liste L wie folgt

  $L = \emptyset$
  for $i = 2$ **to** ∞
  Füge Gegendiagonale i zu L hinzu
  endfor

- $L = \{\frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{1}{3}, \cancel{\frac{2}{2}}, \frac{3}{1}, \frac{1}{4}, \frac{2}{3}, \frac{3}{2}, \frac{4}{1}, \frac{1}{5}, \cancel{\frac{2}{4}}, \cancel{\frac{3}{3}}, \cancel{\frac{4}{2}}, \frac{5}{1}, \ldots$
- Konstruiere L' von L durch das Streichen doppelter Einträge
- $L' = \{1, \frac{1}{2}, 2, \frac{1}{3}, 3, \frac{1}{4}, \frac{2}{3}, \frac{3}{2}, 4, \frac{1}{5}, 5, \ldots$

- Konstruiere Liste L wie folgt

  $L = \emptyset$
  for $i = 2$ **to** ∞
    Füge Gegendiagonale i zu L hinzu
  endfor

- $L = \{\frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{1}{3}, \cancel{\frac{2}{2}}, \frac{3}{1}, \frac{1}{4}, \frac{2}{3}, \frac{3}{2}, \frac{4}{1}, \frac{1}{5}, \cancel{\frac{2}{4}}, \cancel{\frac{3}{3}}, \cancel{\frac{4}{2}}, \frac{5}{1}, \ldots$
- Konstruiere L' von L durch das Streichen doppelter Einträge
- $L' = \{1, \frac{1}{2}, 2, \frac{1}{3}, 3, \frac{1}{4}, \frac{2}{3}, \frac{3}{2}, 4, \frac{1}{5}, 5, \ldots$
- Bijektion: $f : \mathbf{N} \setminus \{0\} \to \mathbf{Q}_+$ sei definiert als:
  $f(i) :=$ $i$ter Eintrag in L'

- Konstruiere Liste L wie folgt

  $L = \emptyset$
  for $i = 2$ **to** ∞
  Füge Gegendiagonale i zu L hinzu
  endfor

- $L = \{\frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{1}{3}, \cancel{\frac{2}{2}}, \frac{3}{1}, \frac{1}{4}, \frac{2}{3}, \frac{3}{2}, \frac{4}{1}, \frac{1}{5}, \cancel{\frac{2}{4}}, \cancel{\frac{3}{3}}, \cancel{\frac{4}{2}}, \frac{5}{1}, \ldots$
- Konstruiere L' von L durch das Streichen doppelter Einträge
- $L' = \{1, \frac{1}{2}, 2, \frac{1}{3}, 3, \frac{1}{4}, \frac{2}{3}, \frac{3}{2}, 4, \frac{1}{5}, 5, \ldots$
- Bijektion: $f : \mathbf{N} \setminus \{0\} \to \mathbf{Q}_+$ sei definiert als:

  $f(i) :=$ $i$ter Eintrag in L'
- Die Idee dieser Nummerierung stammt von Georg Cantor, und heißt deshalb **Cantor-Nummerierung**

Beispiel einer nicht-abzählbaren Menge

Beispiel einer nicht-abzählbaren Menge

- wir bezeichnen die Menge der Funktionen $f\colon X \to Y$ als Y^X

Beispiel einer nicht-abzählbaren Menge

- wir bezeichnen die Menge der Funktionen $f\colon X \to Y$ als Y^X

Satz 18

Die Menge $\{0,1\}^{\mathbf{N}}$ ist nicht abzählbar.

Beispiel einer nicht-abzählbaren Menge

- wir bezeichnen die Menge der Funktionen $f\colon X \to Y$ als Y^X

Satz 18

Die Menge $\{0,1\}^{\mathbf{N}}$ ist nicht abzählbar.

Beweis

- **Technik:** Diagonalisierung

Beispiel einer nicht-abzählbaren Menge

- wir bezeichnen die Menge der Funktionen $f\colon X \to Y$ als Y^X

Satz 18

Die Menge $\{0,1\}^{\mathbf{N}}$ ist nicht abzählbar.

Beweis

- **Technik:** Diagonalisierung
- wir verstehen $f \in \{0,1\}^{\mathbf{N}}$ als abzählbare Folge von 0en und 1en

Beispiel einer nicht-abzählbaren Menge

- wir bezeichnen die Menge der Funktionen $f\colon X \to Y$ als Y^X

Satz 18

Die Menge $\{0,1\}^{\mathbf{N}}$ ist nicht abzählbar.

Beweis

- **Technik:** Diagonalisierung
- wir verstehen $f \in \{0,1\}^{\mathbf{N}}$ als abzählbare Folge von 0en und 1en

$$f \mathrel{\widehat{=}} (0,0,1,1,0,0,1,0,1,\ldots$$

Beispiel einer nicht-abzählbaren Menge

- wir bezeichnen die Menge der Funktionen $f\colon X \to Y$ als Y^X

Satz 18

Die Menge $\{0,1\}^{\mathbf{N}}$ ist nicht abzählbar.

Beweis

- **Technik:** Diagonalisierung
- wir verstehen $f \in \{0,1\}^{\mathbf{N}}$ als abzählbare Folge von 0en und 1en
$$f \mathrel{\widehat{=}} (0, 0, 1, 1, 0, 0, 1, 0, 1, \dots$$
- wir nehmen an, dass $\{0,1\}^{\mathbf{N}}$ abzählbar ist, das heißt, es gibt für jedes $i \in \mathbf{N}$ genau eine Funktion $f_i \in \{0,1\}^{\mathbf{N}}$

Beispiel einer nicht-abzählbaren Menge

- wir bezeichnen die Menge der Funktionen $f\colon X \to Y$ als Y^X

Satz 18

Die Menge $\{0,1\}^{\mathbf{N}}$ ist nicht abzählbar.

Beweis

- **Technik:** Diagonalisierung
- wir verstehen $f \in \{0,1\}^{\mathbf{N}}$ als abzählbare Folge von 0en und 1en
$$f \mathrel{\widehat{=}} (0,0,1,1,0,0,1,0,1,\ldots$$
- wir nehmen an, dass $\{0,1\}^{\mathbf{N}}$ abzählbar ist, das heißt, es gibt für jedes $i \in \mathbf{N}$ genau eine Funktion $f_i \in \{0,1\}^{\mathbf{N}}$
- **wir wollen zeigen:** egal wie diese Bijektion aussieht wir finden immer ein $g \in \{0,1\}^{\mathbf{N}}$, welches in der Aufzählung fehlt

Beispiel einer nicht-abzählbaren Menge

- wir bezeichnen die Menge der Funktionen $f\colon X \to Y$ als Y^X

Satz 18

Die Menge $\{0, 1\}^{\mathbf{N}}$ ist nicht abzählbar.

Beweis

- **Technik:** Diagonalisierung
- wir verstehen $f \in \{0, 1\}^{\mathbf{N}}$ als abzählbare Folge von 0en und 1en
  $$f \mathrel{\widehat{=}} (0, 0, 1, 1, 0, 0, 1, 0, 1, \ldots$$
- wir nehmen an, dass $\{0, 1\}^{\mathbf{N}}$ abzählbar ist, das heißt, es gibt für jedes $i \in \mathbf{N}$ genau eine Funktion $f_i \in \{0, 1\}^{\mathbf{N}}$
- **wir wollen zeigen:** egal wie diese Bijektion aussieht wir finden immer ein $g \in \{0, 1\}^{\mathbf{N}}$, welches in der Aufzählung fehlt
- **Idee:** für jede Aufzählung, konstruiere ein solches g, dass sich von den f_i unterscheidet

Beispiel einer nicht-abzählbaren Menge

- wir bezeichnen die Menge der Funktionen $f\colon X \to Y$ als Y^X

Satz 18

Die Menge $\{0,1\}^{\mathbf{N}}$ ist nicht abzählbar.

Beweis

- **Technik:** Diagonalisierung
- wir verstehen $f \in \{0,1\}^{\mathbf{N}}$ als abzählbare Folge von 0en und 1en
$$f \mathrel{\widehat{=}} (0,0,1,1,0,0,1,0,1,\dots$$
- wir nehmen an, dass $\{0,1\}^{\mathbf{N}}$ abzählbar ist, das heißt, es gibt für jedes $i \in \mathbf{N}$ genau eine Funktion $f_i \in \{0,1\}^{\mathbf{N}}$
- **wir wollen zeigen:** egal wie diese Bijektion aussieht wir finden immer ein $g \in \{0,1\}^{\mathbf{N}}$, welches in der Aufzählung fehlt
- **Idee:** für jede Aufzählung, konstruiere ein solches g, dass sich von den f_i unterscheidet
- für $f_i \not\equiv g$ reicht es, dass f und g sich an einer Stelle unterscheiden

- Zeichne alle f_i in eine Tabelle ein

- Zeichne alle f_i in eine Tabelle ein

	0	1	2	3	4	5	6	7	
f_0	0	1	0	0	0	1	1	1	
f_1	0	0	1	1	0	0	1	1	$\dots$
f_2	0	0	0	1	0	0	1	1	
f_3	1	1	0	1	0	0	0	1	
$\vdots$					$\vdots$				

- Zeichne alle f_i in eine Tabelle ein

	0	1	2	3	4	5	6	7	
f_0	0	1	0	0	0	1	1	1	
f_1	0	0	1	1	0	0	1	1	$\cdots$
f_2	0	0	0	1	0	0	1	1	
f_3	1	1	0	1	0	0	0	1	
$\vdots$					$\vdots$				

- Konstruiere für $(f_i)_{i\in\mathbf{N}}$ folgende Funktion

$$g(x) := \begin{cases} 1, \text{falls } f_x(x) = 0 \\ 0, \text{falls } f_x(x) = 1 \end{cases}$$

- Zeichne alle f_i in eine Tabelle ein

	0	1	2	3	4	5	6	7	
f_0	0	1	0	0	0	1	1	1	
f_1	0	0	1	1	0	0	1	1	$\dots$
f_2	0	0	0	1	0	0	1	1	
f_3	1	1	0	1	0	0	0	1	
$\vdots$					$\vdots$				
g	1	1	1	0	0	1	0	1	

- Konstruiere für $(f_i)_{i \in \mathbf{N}}$ folgende Funktion

$$g(x) := \begin{cases} 1, \text{falls } f_x(x) = 0 \\ 0, \text{falls } f_x(x) = 1 \end{cases}$$

- Zeichne alle f_i in eine Tabelle ein

	0	1	2	3	4	5	6	7	
f_0	0	1	0	0	0	1	1	1	
f_1	0	0	1	1	0	0	1	1	$\dots$
f_2	0	0	0	1	0	0	1	1	
f_3	1	1	0	1	0	0	0	1	
$\vdots$					$\vdots$				
g	1	1	1	0	0	1	0	1	

- Konstruiere für $(f_i)_{i\in\mathbf{N}}$ folgende Funktion

$$g(x) := \begin{cases} 1, \text{falls } f_x(x) = 0 \\ 0, \text{falls } f_x(x) = 1 \end{cases}$$

- $\forall i\colon$ g unterscheidet sich von f_i an der Stelle i ($f_i(i) \neq g(i)$)

- Zeichne alle f_i in eine Tabelle ein

	0	1	2	3	4	5	6	7	
f_0	0	1	0	0	0	1	1	1	
f_1	0	0	1	1	0	0	1	1	$\dots$
f_2	0	0	0	1	0	0	1	1	
f_3	1	1	0	1	0	0	0	1	
$\vdots$					$\vdots$				
g	1	1	1	0	0	1	0	1	

- Konstruiere für $(f_i)_{i \in \mathbf{N}}$ folgende Funktion

$$g(x) := \begin{cases} 1, \text{falls } f_x(x) = 0 \\ 0, \text{falls } f_x(x) = 1 \end{cases}$$

- $\forall i$: g unterscheidet sich von f_i an der Stelle i $(f_i(i) \neq g(i))$
- g fehlt in der Aufzählung der f_i, g ist aber eine Funktion $\{0, 1\}^* \to \mathbf{N}$

$\square$

- Zeichne alle f_i in eine Tabelle ein

	0	1	2	3	4	5	6	7	
f_0	0	1	0	0	0	1	1	1	
f_1	0	0	1	1	0	0	1	1	$\dots$
f_2	0	0	0	1	0	0	1	1	
f_3	1	1	0	1	0	0	0	1	
$\vdots$					$\vdots$				
g	1	1	1	0	0	1	0	1	

- Konstruiere für $(f_i)_{i\in\mathbf{N}}$ folgende Funktion

$$g(x) := \begin{cases} 1, \text{falls } f_x(x) = 0 \\ 0, \text{falls } f_x(x) = 1 \end{cases}$$

- $\forall i:$ g unterscheidet sich von f_i an der Stelle i $(f_i(i) \neq g(i))$
- g fehlt in der Aufzählung der f_i, g ist aber eine Funktion $\{0,1\}^* \to \mathbf{N}$
- es gibt keine Aufzählung der Funktionen $\{0,1\}^{\mathbf{N}}$, und somit ist diese Menge **überabzählbar** $\square$

Korollar

Die Menge der Funktionen $\{0,1\}^* \to \{0,1\}$ ist überabzählbar.

Korollar

Die Menge der Funktionen $\{0,1\}^* \to \{0,1\}$ ist überabzählbar.

Begründung : Da es eine Bijektion zwischen $\mathbf{N}$ und $\{0,1\}^*$ gibt, wäre sonst auch $\{f \mid f\colon \mathbf{N} \to \{0,1\}\}$ abzählbar.

Korollar

Die Menge der Funktionen $\{0,1\}^* \to \{0,1\}$ ist überabzählbar.

Begründung : Da es eine Bijektion zwischen $\mathbf{N}$ und $\{0,1\}^*$ gibt, wäre sonst auch $\{f \mid f\colon \mathbf{N} \to \{0,1\}\}$ abzählbar.

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Korollar

Die Menge der Funktionen $\{0,1\}^* \to \{0,1\}$ ist überabzählbar.

Begründung : Da es eine Bijektion zwischen $\mathbf{N}$ und $\{0,1\}^*$ gibt, wäre sonst auch $\{f \mid f\colon \mathbf{N} \to \{0,1\}\}$ abzählbar.

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Vorüberlegung :

- eine TM kann durch einen endliches Wort kodiert werden

Korollar

Die Menge der Funktionen $\{0,1\}^* \to \{0,1\}$ ist überabzählbar.

Begründung : Da es eine Bijektion zwischen $\mathbf{N}$ und $\{0,1\}^*$ gibt, wäre sonst auch $\{f \mid f\colon \mathbf{N} \to \{0,1\}\}$ abzählbar.

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Vorüberlegung :

- eine TM kann durch einen endliches Wort kodiert werden
  - Nummeriere alle Zustände von $q_0, q_1, \dots q_k$, notiere k
  - $q_A = q_x$ und $q_V = q_y$

Korollar

Die Menge der Funktionen $\{0,1\}^* \to \{0,1\}$ ist überabzählbar.

Begründung : Da es eine Bijektion zwischen $\mathbf{N}$ und $\{0,1\}^*$ gibt, wäre sonst auch $\{f \mid f\colon \mathbf{N} \to \{0,1\}\}$ abzählbar.

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Vorüberlegung :

- eine TM kann durch einen endliches Wort kodiert werden
  - Nummeriere alle Zustände von $q_0, q_1, \ldots q_k$, notiere k
  - $q_A = q_x$ und $q_V = q_y$
  - Kodiere jeden Zustandsübergang δ durch ein Wort z_i und verkette diese Wörter $z = z_1 \# z_2 \# z_3 \cdots$

Korollar

Die Menge der Funktionen $\{0,1\}^* \to \{0,1\}$ ist überabzählbar.

Begründung : Da es eine Bijektion zwischen $\mathbf{N}$ und $\{0,1\}^*$ gibt, wäre sonst auch $\{f \mid f\colon \mathbf{N} \to \{0,1\}\}$ abzählbar.

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Vorüberlegung :

- eine TM kann durch einen endliches Wort kodiert werden
  - Nummeriere alle Zustände von $q_0, q_1, \dots q_k$, notiere k
  - $q_A = q_x$ und $q_V = q_y$
  - Kodiere jeden Zustandsübergang δ durch ein Wort z_i und verkette diese Wörter $z = z_1 \# z_2 \# z_3 \cdots$
  - Wort zu $\delta(j, \mathsf{b}) = (i, \mathsf{a}, L)$ ist $1^j 0 0 1^i 0 1$

Korollar

Die Menge der Funktionen $\{0,1\}^* \to \{0,1\}$ ist überabzählbar.

Begründung : Da es eine Bijektion zwischen $\mathbf{N}$ und $\{0,1\}^*$ gibt, wäre sonst auch $\{f \mid f\colon \mathbf{N} \to \{0,1\}\}$ abzählbar.

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Vorüberlegung :

- eine TM kann durch einen endliches Wort kodiert werden
  - Nummeriere alle Zustände von $q_0, q_1, \ldots q_k$, notiere k
  - $q_A = q_x$ und $q_V = q_y$
  - Kodiere jeden Zustandsübergang δ durch ein Wort z_i und verkette diese Wörter $z = z_1\#z_2\#z_3\cdots$
  - Wort zu $\delta(j,\mathsf{b}) = (i,\mathsf{a},L)$ ist 1^j001^i01

0 entspricht a ,00 entspricht b, usw.

1 entspricht L
11 entspricht R

Korollar

Die Menge der Funktionen $\{0,1\}^* \to \{0,1\}$ ist überabzählbar.

Begründung : Da es eine Bijektion zwischen $\mathbf{N}$ und $\{0,1\}^*$ gibt, wäre sonst auch $\{f \mid f\colon \mathbf{N} \to \{0,1\}\}$ abzählbar.

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Vorüberlegung :

- eine TM kann durch einen endliches Wort kodiert werden
  - Nummeriere alle Zustände von $q_0, q_1, \ldots q_k$, notiere k
  - $q_A = q_x$ und $q_V = q_y$
  - Kodiere jeden Zustandsübergang δ durch ein Wort z_i und verkette diese Wörter $z = z_1\#z_2\#z_3\cdots$
  - Wort zu $\delta(j,\mathtt{b}) = (i,\mathtt{a},L)$ ist 1^j001^i01

    0 entspricht a ,00 entspricht b, usw.

    1 entspricht L
    11 entspricht R
  - Kodierung $\langle T\rangle$ ist Binärkodierung von $1^k0^x1^y\#z$ (Turingwort)

- jedem Wort über $\{0,1\}$ kann eine TM zugewiesen werden

$$T_w := \begin{cases} \text{TM für Turingwort } w & \text{, falls } w \text{ Kodierung für TM} \\ \text{alles verwerfende TM} & \text{, sonst} \end{cases}$$

- jedem Wort über $\{0, 1\}$ kann eine TM zugewiesen werden

$$T_w := \begin{cases} \text{TM für Turingwort } w & \text{, falls } w \text{ Kodierung für TM} \\ \text{alles verwerfende TM} & \text{, sonst} \end{cases}$$

- jede TM erscheint als Kodierung T_w für ein Wort w

- jedem Wort über $\{0, 1\}$ kann eine TM zugewiesen werden

$$T_w := \begin{cases} \text{TM für Turingwort } w & \text{, falls } w \text{ Kodierung für TM} \\ \text{alles verwerfende TM} & \text{, sonst} \end{cases}$$

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

- jedem Wort über $\{0,1\}$ kann eine TM zugewiesen werden

$$T_w := \begin{cases} \text{TM für Turingwort } w & \text{, falls } w \text{ Kodierung für TM} \\ \text{alles verwerfende TM} & \text{, sonst} \end{cases}$$

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

- $\mathcal{M} = \{\langle M \rangle \mid M \text{ ist TM } \} = \{0,1\}^*$ ist abzählbar

- jedem Wort über $\{0, 1\}$ kann eine TM zugewiesen werden

$$T_w := \begin{cases} \text{TM für Turingwort } w & \text{, falls } w \text{ Kodierung für TM} \\ \text{alles verwerfende TM} & \text{, sonst} \end{cases}$$

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

- $\mathcal{M} = \{\langle M \rangle \mid M \text{ ist TM } \} = \{0, 1\}^*$ ist abzählbar
  $\rightarrow \mathbb{A} = \{L(M) \mid M \in \mathcal{M}\}$ abzählbar

- jedem Wort über $\{0,1\}$ kann eine TM zugewiesen werden

$$T_w := \begin{cases} \text{TM für Turingwort } w & \text{, falls } w \text{ Kodierung für TM} \\ \text{alles verwerfende TM} & \text{, sonst} \end{cases}$$

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

- $\mathcal{M} = \{\langle M \rangle \mid M \text{ ist TM } \} = \{0,1\}^*$ ist abzählbar
  $\to \mathbb{A} = \{L(M) \mid M \in \mathcal{M}\}$ abzählbar
- $\mathcal{L} = \{L \subseteq \{0,1\}^*\}$ ist die Menge aller Sprachen über $\{0,1\}$

- jedem Wort über $\{0,1\}$ kann eine TM zugewiesen werden

$$T_w := \begin{cases} \text{TM für Turingwort } w & \text{, falls } w \text{ Kodierung für TM} \\ \text{alles verwerfende TM} & \text{, sonst} \end{cases}$$

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

- $\mathcal{M} = \{\langle M\rangle \mid M \text{ ist TM } \} = \{0,1\}^*$ ist abzählbar
  $\rightarrow \mathbb{A} = \{L(M) \mid M \in \mathcal{M}\}$ abzählbar
- $\mathcal{L} = \{L \subseteq \{0,1\}^*\}$ ist die Menge aller Sprachen über $\{0,1\}$
- charakteristische Funktion zu $L \in \mathcal{L}$: $\chi_L(w) = \begin{cases} 0 & w \notin L \\ 1 & w \in L \end{cases}$

- jedem Wort über $\{0,1\}$ kann eine TM zugewiesen werden

$$T_w := \begin{cases} \text{TM für Turingwort } w & \text{, falls } w \text{ Kodierung für TM} \\ \text{alles verwerfende TM} & \text{, sonst} \end{cases}$$

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

- $\mathcal{M} = \{\langle M \rangle \mid M \text{ ist TM } \} = \{0,1\}^*$ ist abzählbar
  $\to \mathbb{A} = \{L(M) \mid M \in \mathcal{M}\}$ abzählbar
- $\mathcal{L} = \{L \subseteq \{0,1\}^*\}$ ist die Menge aller Sprachen über $\{0,1\}$
- charakteristische Funktion zu $L \in \mathcal{L}$: $\chi_L(w) = \begin{cases} 0 & w \notin L \\ 1 & w \in L \end{cases}$
- 1-1 Beziehung zwischen charakteristischen Funktionen und $\mathcal{L}$

- jedem Wort über $\{0,1\}$ kann eine TM zugewiesen werden
  $$T_w := \begin{cases} \text{TM für Turingwort } w & \text{, falls } w \text{ Kodierung für TM} \\ \text{alles verwerfende TM} & \text{, sonst} \end{cases}$$
- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

- $\mathcal{M} = \{\langle M\rangle \mid \ M \text{ ist TM } \} = \{0,1\}^*$ ist abzählbar
  $\rightarrow \mathbb{A} = \{L(M) \mid M \in \mathcal{M}\}$ abzählbar
- $\mathcal{L} = \{L \subseteq \{0,1\}^*\}$ ist die Menge aller Sprachen über $\{0,1\}$
- charakteristische Funktion zu $L \in \mathcal{L}$: $\chi_L(w) = \begin{cases} 0 & w \notin L \\ 1 & w \in L \end{cases}$
- 1-1 Beziehung zwischen charakteristischen Funktionen und $\mathcal{L}$
- Menge der charakteristischen Funktionen ist $\{0,1\}^{\{0,1\}^*}$
  $\rightarrow$ überabzählbar $\rightarrow \mathcal{L}$ ist überabzählbar

- jedem Wort über $\{0,1\}$ kann eine TM zugewiesen werden

$$T_w := \begin{cases} \text{TM für Turingwort } w & \text{, falls } w \text{ Kodierung für TM} \\ \text{alles verwerfende TM} & \text{, sonst} \end{cases}$$

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

- $\mathcal{M} = \{\langle M \rangle \mid M \text{ ist TM } \} = \{0,1\}^*$ ist abzählbar
  $\rightarrow \mathbb{A} = \{L(M) \mid M \in \mathcal{M}\}$ abzählbar
- $\mathcal{L} = \{L \subseteq \{0,1\}^*\}$ ist die Menge aller Sprachen über $\{0,1\}$
- charakteristische Funktion zu $L \in \mathcal{L}$: $\chi_L(w) = \begin{cases} 0 & w \notin L \\ 1 & w \in L \end{cases}$
- 1-1 Beziehung zwischen charakteristischen Funktionen und $\mathcal{L}$
- Menge der charakteristischen Funktionen ist $\{0,1\}^{\{0,1\}^*}$
  $\rightarrow$ überabzählbar $\rightarrow \mathcal{L}$ ist überabzählbar
- es gibt "mehr" Sprachen als Turingmaschinen, mindestens eine Sprache wird nicht erkannt $\square$