

# Theoretische Informatik

## Einheit 3

### Elementare Berechenbarkeitstheorie



1. Aufzählbarkeit und Entscheidbarkeit
2. Universelle Maschinen
3. Beweistechniken für unlösbare Probleme

## Untersuchung von Fragen der Berechenbarkeit unabhängig vom Modell

## Untersuchung von Fragen der Berechenbarkeit unabhängig vom Modell

- **Was kann überhaupt berechnet werden?**
  - Welche **Funktionen** sind berechenbar?
  - Welche **Programmeigenschaften** kann man entscheiden (testen)?

## Untersuchung von Fragen der Berechenbarkeit unabhängig vom Modell

- **Was kann überhaupt berechnet werden?**
  - Welche **Funktionen** sind berechenbar?
  - Welche **Programmeigenschaften** kann man entscheiden (testen)?
- **Wie kann man Lösungen wiederverwenden?**
  - Verwendung von **Abschlußeigenschaften**
  - **Transformation** in ein anderes Problem (Reduktion)

## Untersuchung von Fragen der Berechenbarkeit unabhängig vom Modell

- **Was kann überhaupt berechnet werden?**
  - Welche **Funktionen** sind berechenbar?
  - Welche **Programmeigenschaften** kann man entscheiden (testen)?
- **Wie kann man Lösungen wiederverwenden?**
  - Verwendung von **Abschlußeigenschaften**
  - **Transformation** in ein anderes Problem (Reduktion)
- **Wo liegen die Grenzen?**
  - Terminierung, Korrektheit, Äquivalenz, Optimalität von Programmen

## Untersuchung von Fragen der Berechenbarkeit unabhängig vom Modell

- **Was kann überhaupt berechnet werden?**
  - Welche **Funktionen** sind berechenbar?
  - Welche **Programmeigenschaften** kann man entscheiden (testen)?
- **Wie kann man Lösungen wiederverwenden?**
  - Verwendung von **Abschlußeigenschaften**
  - **Transformation** in ein anderes Problem (Reduktion)
- **Wo liegen die Grenzen?**
  - Terminierung, Korrektheit, Äquivalenz, Optimalität von Programmen
- **Welche Beweistechniken gibt es?**
  - Konstruktion von abstrakten Gegenbeispielen durch **Diagonalisierung**
  - **Problemreduktion**
  - Direkte Beweise (**Busy Beaver**)

- **Konzepte** modellunabhängig präzisieren
  - Berechenbare Funktionen
  - (Semi-)entscheidbare und aufzählbare Mengen

- **Konzepte** modellunabhängig präzisieren
  - Berechenbare Funktionen
  - (Semi-)entscheidbare und aufzählbare Mengen
- **Grundeigenschaften** aus Modellen herleiten
  - Zusammenhänge zwischen den Konzepten
  - Abschlußeigenschaften
  - Codierung von Programmen als Daten
  - Existenz universeller Funktionen

- **Konzepte** modellunabhängig präzisieren
  - Berechenbare Funktionen
  - (Semi-)entscheidbare und aufzählbare Mengen
- **Grundeigenschaften** aus Modellen herleiten
  - Zusammenhänge zwischen den Konzepten
  - Abschlußeigenschaften
  - Codierung von Programmen als Daten
  - Existenz universeller Funktionen
- **Theorie nur auf dieser Basis** weiterführen
  - Hier: die wichtigsten Unmöglichkeitsaussagen

# Theoretische Informatik



## Einheit 3.1

### Aufzählbarkeit und Entscheidbarkeit



1. Präzisierung der Begriffe
2. Zusammenhänge
3. Abschlußeigenschaften

# BERECHENBARKEIT

- Berechenbarkeit auf **Worten**

- $f: X^* \rightarrow Y^*$  berechenbar, falls  $f$  Turing-berechenbar

# BERECHENBARKEIT

- Berechenbarkeit auf **Worten**

- $f: X^* \rightarrow Y^*$  berechenbar, falls  $f$  Turing-berechenbar

- Berechenbarkeit auf **Wort-Tupeln**  $f: X^* \times X^* \rightarrow Y^*$

- $f: X^* \times X^* \rightarrow Y^*$  berechenbar, falls  $f': (X \cup \{\#\})^* \rightarrow Y^*$

- mit  $f'(v\#w) = f(v,w)$  berechenbar

# BERECHENBARKEIT

- Berechenbarkeit auf **Worten**

- $f: X^* \rightarrow Y^*$  berechenbar, falls  $f$  Turing-berechenbar

- Berechenbarkeit auf **Wort-Tupeln**  $f: X^* \times X^* \rightarrow Y^*$

- $f: X^* \times X^* \rightarrow Y^*$  berechenbar, falls  $f': (X \cup \{\#\})^* \rightarrow Y^*$

mit  $f'(v\#w) = f(v,w)$  berechenbar

- Berechenbarkeit auf **Zahlen**

- $f: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, falls  $f_r: X^* \rightarrow X^*$

mit  $f_r(w) = r(f(r^{-1}(w)))$  berechenbar

- $r: \mathbb{N} \rightarrow X^*$  bijektive Repräsentation von Zahlen als Worte

# BERECHENBARKEIT

## ● Berechenbarkeit auf **Worten**

–  $f: X^* \rightarrow Y^*$  berechenbar, falls  $f$  Turing-berechenbar

## ● Berechenbarkeit auf **Wort-Tupeln** $f: X^* \times X^* \rightarrow Y^*$

–  $f: X^* \times X^* \rightarrow Y^*$  berechenbar, falls  $f': (X \cup \{\#\})^* \rightarrow Y^*$

mit  $f'(v\#w) = f(v, w)$  berechenbar

## ● Berechenbarkeit auf **Zahlen**

–  $f: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, falls  $f_r: X^* \rightarrow X^*$

mit  $f_r(w) = r(f(r^{-1}(w)))$  berechenbar

–  $r: \mathbb{N} \rightarrow X^*$  bijektive Repräsentation von Zahlen als Worte

## ● Berechenbarkeit auf **Zahlentupeln und -listen**

–  $f: \mathbb{N}^k \rightarrow \mathbb{N}$  berechenbar, falls  $f': \mathbb{N} \rightarrow \mathbb{N}$

mit  $f'\langle x_1, x_2, \dots, x_k \rangle^k = f(x_1, x_2, \dots, x_k)$  berechenbar

–  $f: \mathbb{N}^* \rightarrow \mathbb{N}$  berechenbar, falls  $f': \mathbb{N} \rightarrow \mathbb{N}$

mit  $f'\langle x_1, x_2, \dots, x_k \rangle^* = f(x_1 x_2 \dots x_k)$  berechenbar

–  $\langle \rangle^k: \mathbb{N}^k \rightarrow \mathbb{N}$ ,  $\langle \rangle^*: \mathbb{N}^* \rightarrow \mathbb{N}$  Standard-Tupelfunktionen

- **Entscheidbarkeit** einer Menge
  - Wir können eine Maschine konstruieren, die *testet*,  
ob ein bestimmtes Element *zur Menge gehört oder nicht*

- **Entscheidbarkeit** einer Menge
  - Wir können eine Maschine konstruieren, die *testet*, ob ein bestimmtes Element *zur Menge gehört oder nicht*
- **Semi-Entscheidbarkeit** einer Menge
  - Wir können eine Maschine konstruieren, die *testet*, ob ein bestimmtes Element *zur Menge gehört*, aber im Mißerfolgsfall *eventuell niemals eine Antwort* gibt

- **Entscheidbarkeit** einer Menge
  - Wir können eine Maschine konstruieren, die *testet*, ob ein bestimmtes Element *zur Menge gehört oder nicht*
- **Semi-Entscheidbarkeit** einer Menge
  - Wir können eine Maschine konstruieren, die *testet*, ob ein bestimmtes Element *zur Menge gehört*, aber im Mißerfolgsfall *eventuell niemals eine Antwort* gibt
- **Aufzählbarkeit** einer Menge
  - Wir können eine Maschine konstruieren, welche die *Elemente* der Menge *schrittweise generiert*, also z.B. bei Eingabe der Zahl  $n$  (oder einer Codierung) das  $n$ -te Element der Menge ausgibt

# AUFZÄHLBARKEIT UND ENTSCHEIDBARKEIT – PRÄZISIERT

- $M \subseteq X^*$  **entscheidbar**

- $\chi_M: X^* \rightarrow \{0,1\}^*$  berechenbar, wobei  $\chi_M(w) = \begin{cases} 1 & \text{falls } w \in M, \\ 0 & \text{sonst} \end{cases}$

# AUFZÄHLBARKEIT UND ENTSCHEIDBARKEIT – PRÄZISIERT

- $M \subseteq X^*$  **entscheidbar**

- $\chi_M: X^* \rightarrow \{0,1\}^*$  berechenbar, wobei  $\chi_M(w) = \begin{cases} 1 & \text{falls } w \in M, \\ 0 & \text{sonst} \end{cases}$

- $M \subseteq X^*$  **semi-entscheidbar**

- $\psi_M: X^* \rightarrow \{0,1\}^*$  berechenbar, wobei  $\psi_M(w) = \begin{cases} 1 & \text{falls } w \in M, \\ \perp & \text{sonst} \end{cases}$

# AUFZÄHLBARKEIT UND ENTSCHEIDBARKEIT – PRÄZISIERT

- $M \subseteq X^*$  **entscheidbar**

- $\chi_M: X^* \rightarrow \{0,1\}^*$  berechenbar, wobei  $\chi_M(w) = \begin{cases} 1 & \text{falls } w \in M, \\ 0 & \text{sonst} \end{cases}$

- $M \subseteq X^*$  **semi-entscheidbar**

- $\psi_M: X^* \rightarrow \{0,1\}^*$  berechenbar, wobei  $\psi_M(w) = \begin{cases} 1 & \text{falls } w \in M, \\ \perp & \text{sonst} \end{cases}$

- $M \subseteq X^*$  **(rekursiv) aufzählbar**

- $M = \emptyset$  oder es gibt eine **totale**, berechenbare Funktion  $f: \{1\}^* \rightarrow X^*$   
mit  $M = \text{range}(f) = \{v \in X^* \mid \exists w \in \{1\}^* f(w) = v\}$

# AUFZÄHLBARKEIT UND ENTSCHEIDBARKEIT – PRÄZISIERT

- $M \subseteq X^*$  **entscheidbar**

- $\chi_M: X^* \rightarrow \{0,1\}^*$  berechenbar, wobei  $\chi_M(w) = \begin{cases} 1 & \text{falls } w \in M, \\ 0 & \text{sonst} \end{cases}$

- $M \subseteq X^*$  **semi-entscheidbar**

- $\psi_M: X^* \rightarrow \{0,1\}^*$  berechenbar, wobei  $\psi_M(w) = \begin{cases} 1 & \text{falls } w \in M, \\ \perp & \text{sonst} \end{cases}$

- $M \subseteq X^*$  **(rekursiv) aufzählbar**

- $M = \emptyset$  oder es gibt eine **totale**, berechenbare Funktion  $f: \{1\}^* \rightarrow X^*$   
mit  $M = \text{range}(f) = \{v \in X^* \mid \exists w \in \{1\}^* f(w) = v\}$

- **Berechenbarkeit von Mengen  $M \subseteq \mathbb{N}$  analog**

- Jede **endliche Menge** ist **aufzählbar**

- $M = \{x_0, \dots, x_n\}$  ist Wertebereich von  $f$  mit  $f(n) = \begin{cases} x_i & \text{falls } i \leq n, \\ x_0 & \text{sonst} \end{cases}$
- $f$  ist primitiv rekursiv, also berechenbar

# AUFZÄHLBARKEIT – ANMERKUNGEN

- Jede **endliche Menge** ist **aufzählbar**

- $M = \{x_0, \dots, x_n\}$  ist Wertebereich von  $f$  mit  $f(n) = \begin{cases} x_i & \text{falls } i \leq n, \\ x_0 & \text{sonst} \end{cases}$
- $f$  ist primitiv rekursiv, also berechenbar

- **Aufzählungen** sind **nicht notwendigerweise injektiv**

- Die Funktion  $g$  mit  $g(n) = \begin{cases} n+1 & \text{falls } n \text{ gerade,} \\ n & \text{sonst} \end{cases}$   
zählt alle ungeraden Zahlen genau zweimal auf

- Jede **endliche Menge** ist **aufzählbar**

- $M = \{x_0, \dots, x_n\}$  ist Wertebereich von  $f$  mit  $f(n) = \begin{cases} x_i & \text{falls } i \leq n, \\ x_0 & \text{sonst} \end{cases}$
- $f$  ist primitiv rekursiv, also berechenbar

- **Aufzählungen** sind **nicht notwendigerweise injektiv**

- Die Funktion  $g$  mit  $g(n) = \begin{cases} n+1 & \text{falls } n \text{ gerade,} \\ n & \text{sonst} \end{cases}$   
zählt alle ungeraden Zahlen genau zweimal auf

- Jede Menge hat **verschiedene Aufzählungen**

- Die Funktion  $h$  mit  $h(n) = 2n+1$   
zählt alle ungeraden Zahlen genau einmal auf

- Jede **endliche Menge** ist **aufzählbar**

- $M = \{x_0, \dots, x_n\}$  ist Wertebereich von  $f$  mit  $f(n) = \begin{cases} x_i & \text{falls } i \leq n, \\ x_0 & \text{sonst} \end{cases}$
- $f$  ist primitiv rekursiv, also berechenbar

- **Aufzählungen** sind **nicht notwendigerweise injektiv**

- Die Funktion  $g$  mit  $g(n) = \begin{cases} n+1 & \text{falls } n \text{ gerade,} \\ n & \text{sonst} \end{cases}$   
zählt alle ungeraden Zahlen genau zweimal auf

- Jede Menge hat **verschiedene Aufzählungen**

- Die Funktion  $h$  mit  $h(n) = 2n+1$   
zählt alle ungeraden Zahlen genau einmal auf

- **Es gibt viele äquivalente Charakterisierungen**

- **Aufzählbar** — semi-entscheidbar
- **Werte- oder Haltebereich** einer (evtl. partiellen) berechenbaren Funktion

Für  $M \subseteq \mathbb{N}$  sind folgende Aussagen äquivalent

Für  $M \subseteq \mathbb{N}$  sind folgende Aussagen äquivalent

1.  $M$  ist **aufzählbar**

Für  $M \subseteq \mathbb{N}$  sind folgende Aussagen äquivalent

1.  $M$  ist aufzählbar

2.  $M = \text{range}(f)$  für ein berechenbares  $f: \mathbb{N} \rightarrow \mathbb{N}$

–  $\text{range}(f) = \{f(i) \mid i \in \mathbb{N}\}$ ,  $f$  nicht notwendigerweise total

Für  $M \subseteq \mathbb{N}$  sind folgende Aussagen äquivalent

1.  $M$  ist **aufzählbar**

2.  $M = \text{range}(f)$  für ein berechenbares  $f: \mathbb{N} \rightarrow \mathbb{N}$

–  $\text{range}(f) = \{f(i) \mid i \in \mathbb{N}\}$ ,  $f$  nicht notwendigerweise total

3.  $M$  ist **semi-entscheidbar**

Für  $M \subseteq \mathbb{N}$  sind folgende Aussagen äquivalent

1.  $M$  ist **aufzählbar**

2.  $M = \text{range}(f)$  für ein berechenbares  $f: \mathbb{N} \rightarrow \mathbb{N}$

–  $\text{range}(f) = \{f(i) \mid i \in \mathbb{N}\}$ ,  $f$  nicht notwendigerweise total

3.  $M$  ist **semi-entscheidbar**

4.  $M = \text{domain}(f)$  für ein berechenbares  $f: \mathbb{N} \rightarrow \mathbb{N}$

–  $\text{domain}(f) = \{i \in \mathbb{N} \mid f(i) \neq \perp\}$

# CHARAKTERISIERUNGEN VON AUFZÄHLBARKEIT

Für  $M \subseteq \mathbb{N}$  sind folgende Aussagen äquivalent

1.  $M$  ist **aufzählbar**

2.  $M = \text{range}(f)$  für ein berechenbares  $f: \mathbb{N} \rightarrow \mathbb{N}$

–  $\text{range}(f) = \{f(i) \mid i \in \mathbb{N}\}$ ,  $f$  nicht notwendigerweise total

3.  $M$  ist **semi-entscheidbar**

4.  $M = \text{domain}(f)$  für ein berechenbares  $f: \mathbb{N} \rightarrow \mathbb{N}$

–  $\text{domain}(f) = \{i \in \mathbb{N} \mid f(i) \neq \perp\}$

**Aussage gilt analog für  $M \subseteq X^*$ ,  $M \subseteq \mathbb{N}^k$ , ...**

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS

- $M$  aufzählbar  $\Rightarrow M = \text{range}(f)$  für ein berechenbares  $f$

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS

- $M$  aufzählbar  $\Rightarrow M = \text{range}(f)$  für ein berechenbares  $f$ 
  - Falls  $M = \emptyset$  dann ist  $M = \text{range}(f_{\perp})$ , wobei  $f_{\perp}(i) = \perp$  für alle  $i \in \mathbb{N}$



# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS

- $M$  aufzählbar  $\Rightarrow M = \text{range}(f)$  für ein berechenbares  $f$ 
  - Falls  $M = \emptyset$  dann ist  $M = \text{range}(f_{\perp})$ , wobei  $f_{\perp}(i) = \perp$  für alle  $i \in \mathbb{N}$  ✓
  - Andernfalls  $M = \text{range}(f)$  für ein berechenbares, totales  $f$  ✓

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS

- $M$  aufzählbar  $\Rightarrow M = \text{range}(f)$  für ein berechenbares  $f$ 
  - Falls  $M = \emptyset$  dann ist  $M = \text{range}(f_{\perp})$ , wobei  $f_{\perp}(i) = \perp$  für alle  $i \in \mathbb{N}$  ✓
  - Andernfalls  $M = \text{range}(f)$  für ein berechenbares, totales  $f$  ✓
- $M = \text{range}(f)$  für ein berechenbares  $f \Rightarrow M$  semi-entscheidbar

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS

- $M$  aufzählbar  $\Rightarrow M = \text{range}(f)$  für ein berechenbares  $f$ 
  - Falls  $M = \emptyset$  dann ist  $M = \text{range}(f_{\perp})$ , wobei  $f_{\perp}(i) = \perp$  für alle  $i \in \mathbb{N}$  ✓
  - Andernfalls  $M = \text{range}(f)$  für ein berechenbares, totales  $f$  ✓
- $M = \text{range}(f)$  für ein berechenbares  $f \Rightarrow M$  semi-entscheidbar
  - Es sei  $M = \text{range}(f)$  für ein berechenbares  $f$

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS

- **$M$  aufzählbar  $\Rightarrow M = \text{range}(f)$  für ein berechenbares  $f$** 
  - Falls  $M = \emptyset$  dann ist  $M = \text{range}(f_{\perp})$ , wobei  $f_{\perp}(i) = \perp$  für alle  $i \in \mathbb{N}$  ✓
  - Andernfalls  $M = \text{range}(f)$  für ein berechenbares, totales  $f$  ✓
- **$M = \text{range}(f)$  für ein berechenbares  $f \Rightarrow M$  semi-entscheidbar**
  - Es sei  $M = \text{range}(f)$  für ein berechenbares  $f$
  - Es ist  $\psi_M(w) = \begin{cases} 1 & \exists x, i. f(x) \text{ hält nach } i \text{ Schritten mit } f(x) = w, \\ \perp & \text{sonst} \end{cases}$
  - Damit ist  $\psi_M$  berechenbar und  $M$  semi-entscheidbar ✓

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS

- **$M$  aufzählbar  $\Rightarrow M = \text{range}(f)$  für ein berechenbares  $f$** 
  - Falls  $M = \emptyset$  dann ist  $M = \text{range}(f_{\perp})$ , wobei  $f_{\perp}(i) = \perp$  für alle  $i \in \mathbb{N}$  ✓
  - Andernfalls  $M = \text{range}(f)$  für ein berechenbares, totales  $f$  ✓
- **$M = \text{range}(f)$  für ein berechenbares  $f \Rightarrow M$  semi-entscheidbar**
  - Es sei  $M = \text{range}(f)$  für ein berechenbares  $f$
  - Es ist  $\psi_M(w) = \begin{cases} 1 & \exists x, i. f(x) \text{ hält nach } i \text{ Schritten mit } f(x) = w, \\ \perp & \text{sonst} \end{cases}$
  - Damit ist  $\psi_M$  berechenbar und  $M$  semi-entscheidbar ✓
- **$M$  semi-entscheidbar  $\Rightarrow M = \text{domain}(f)$  für ein berechenbares  $f$**

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS

- **$M$  aufzählbar  $\Rightarrow M = \text{range}(f)$  für ein berechenbares  $f$** 
  - Falls  $M = \emptyset$  dann ist  $M = \text{range}(f_{\perp})$ , wobei  $f_{\perp}(i) = \perp$  für alle  $i \in \mathbb{N}$  ✓
  - Andernfalls  $M = \text{range}(f)$  für ein berechenbares, totales  $f$  ✓
- **$M = \text{range}(f)$  für ein berechenbares  $f \Rightarrow M$  semi-entscheidbar**
  - Es sei  $M = \text{range}(f)$  für ein berechenbares  $f$
  - Es ist  $\psi_M(w) = \begin{cases} 1 & \exists x, i. f(x) \text{ hält nach } i \text{ Schritten mit } f(x) = w, \\ \perp & \text{sonst} \end{cases}$
  - Damit ist  $\psi_M$  berechenbar und  $M$  semi-entscheidbar ✓
- **$M$  semi-entscheidbar  $\Rightarrow M = \text{domain}(f)$  für ein berechenbares  $f$** 
  - Es sei  $M$  semi-entscheidbar.

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS

- **$M$  aufzählbar  $\Rightarrow M = \text{range}(f)$  für ein berechenbares  $f$** 
  - Falls  $M = \emptyset$  dann ist  $M = \text{range}(f_{\perp})$ , wobei  $f_{\perp}(i) = \perp$  für alle  $i \in \mathbb{N}$  ✓
  - Andernfalls  $M = \text{range}(f)$  für ein berechenbares, totales  $f$  ✓
- **$M = \text{range}(f)$  für ein berechenbares  $f \Rightarrow M$  semi-entscheidbar**
  - Es sei  $M = \text{range}(f)$  für ein berechenbares  $f$
  - Es ist  $\psi_M(w) = \begin{cases} 1 & \exists x, i. f(x) \text{ hält nach } i \text{ Schritten mit } f(x) = w, \\ \perp & \text{sonst} \end{cases}$
  - Damit ist  $\psi_M$  berechenbar und  $M$  semi-entscheidbar ✓
- **$M$  semi-entscheidbar  $\Rightarrow M = \text{domain}(f)$  für ein berechenbares  $f$** 
  - Es sei  $M$  semi-entscheidbar.
  - Dann ist  $\psi_M$  berechenbar und  $M = \{i \in \mathbb{N} \mid \psi_M(i) = 1\} = \text{domain}(\psi_M)$  ✓

## BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS (II)

- $M = \text{domain}(f)$  für ein berechenbares  $f \Rightarrow M$  aufzählbar

## BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS (II)

- $M = \text{domain}(f)$  für ein berechenbares  $f \Rightarrow M$  aufzählbar
  - Es sei  $M = \text{domain}(f) = \{i \in \mathbb{N} \mid f(i) \neq \perp\}$  für ein berechenbares  $f$

## BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS (II)

- $M = \text{domain}(f)$  für ein berechenbares  $f \Rightarrow M$  aufzählbar
  - Es sei  $M = \text{domain}(f) = \{i \in \mathbb{N} \mid f(i) \neq \perp\}$  für ein berechenbares  $f$
  - Falls  $M = \emptyset$ , dann ist  $M$  per Definition aufzählbar ✓

## BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS (II)

- $M = \text{domain}(f)$  für ein berechenbares  $f \Rightarrow M$  aufzählbar

- Es sei  $M = \text{domain}(f) = \{i \in \mathbb{N} \mid f(i) \neq \perp\}$  für ein berechenbares  $f$

- Falls  $M = \emptyset$ , dann ist  $M$  per Definition aufzählbar

✓

- Andernfalls gibt es ein  $n_0 \in \mathbb{N}$  mit  $f(n_0) \neq \perp$  und eine TM  $\tau$  mit  $f = h_\tau$

## BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS (II)

- $M = \text{domain}(f)$  für ein berechenbares  $f \Rightarrow M$  aufzählbar

- Es sei  $M = \text{domain}(f) = \{i \in \mathbb{N} \mid f(i) \neq \perp\}$  für ein berechenbares  $f$

- Falls  $M = \emptyset$ , dann ist  $M$  per Definition aufzählbar

✓

- Andernfalls gibt es ein  $n_0 \in \mathbb{N}$  mit  $f(n_0) \neq \perp$  und eine TM  $\tau$  mit  $f = h_\tau$

- Wir konstruieren eine Turingmaschine  $\tau'$  mit  $M = \text{range}(h_{\tau'})$

Bei Eingabe einer (Repräsentation der) Zahl  $n$  arbeite  $\tau'$  wie folgt

## BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS (II)

- $M = \text{domain}(f)$  für ein berechenbares  $f \Rightarrow M$  aufzählbar

- Es sei  $M = \text{domain}(f) = \{i \in \mathbb{N} \mid f(i) \neq \perp\}$  für ein berechenbares  $f$

- Falls  $M = \emptyset$ , dann ist  $M$  per Definition aufzählbar

✓

- Andernfalls gibt es ein  $n_0 \in \mathbb{N}$  mit  $f(n_0) \neq \perp$  und eine TM  $\tau$  mit  $f = h_\tau$

- Wir konstruieren eine Turingmaschine  $\tau'$  mit  $M = \text{range}(h_{\tau'})$

Bei Eingabe einer (Repräsentation der) Zahl  $n$  arbeite  $\tau'$  wie folgt

1. Berechne  $i, j$  so daß  $n = \langle i, j \rangle$

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS (II)

- $M = \text{domain}(f)$  für ein berechenbares  $f \Rightarrow M$  aufzählbar

- Es sei  $M = \text{domain}(f) = \{i \in \mathbb{N} \mid f(i) \neq \perp\}$  für ein berechenbares  $f$

- Falls  $M = \emptyset$ , dann ist  $M$  per Definition aufzählbar

✓

- Andernfalls gibt es ein  $n_0 \in \mathbb{N}$  mit  $f(n_0) \neq \perp$  und eine TM  $\tau$  mit  $f = h_\tau$

- Wir konstruieren eine Turingmaschine  $\tau'$  mit  $M = \text{range}(h_{\tau'})$

Bei Eingabe einer (Repräsentation der) Zahl  $n$  arbeite  $\tau'$  wie folgt

1. Berechne  $i, j$  so daß  $n = \langle i, j \rangle$

2. Simuliere die Berechnung von  $\tau$  bei Eingabe  $i$  für (maximal)  $j$  Schritte  
(Erzeuge Anfangskonfiguration, simuliere  $\hat{\delta}_\tau$  auf separatem Band)

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS (II)

- $M = \text{domain}(f)$  für ein berechenbares  $f \Rightarrow M$  aufzählbar

- Es sei  $M = \text{domain}(f) = \{i \in \mathbb{N} \mid f(i) \neq \perp\}$  für ein berechenbares  $f$

- Falls  $M = \emptyset$ , dann ist  $M$  per Definition aufzählbar



- Andernfalls gibt es ein  $n_0 \in \mathbb{N}$  mit  $f(n_0) \neq \perp$  und eine TM  $\tau$  mit  $f = h_\tau$

- Wir konstruieren eine Turingmaschine  $\tau'$  mit  $M = \text{range}(h_{\tau'})$

Bei Eingabe einer (Repräsentation der) Zahl  $n$  arbeite  $\tau'$  wie folgt

1. Berechne  $i, j$  so daß  $n = \langle i, j \rangle$

2. Simuliere die Berechnung von  $\tau$  bei Eingabe  $i$  für (maximal)  $j$  Schritte  
(Erzeuge Anfangskonfiguration, simuliere  $\hat{\delta}_\tau$  auf separatem Band)

3. Falls  $\tau$  in  $j$  Schritten anhält, gebe  $i$  aus. Andernfalls gebe  $n_0$  aus

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS (II)

- $M = \text{domain}(f)$  für ein berechenbares  $f \Rightarrow M$  aufzählbar

- Es sei  $M = \text{domain}(f) = \{i \in \mathbb{N} \mid f(i) \neq \perp\}$  für ein berechenbares  $f$

- Falls  $M = \emptyset$ , dann ist  $M$  per Definition aufzählbar

✓

- Andernfalls gibt es ein  $n_0 \in \mathbb{N}$  mit  $f(n_0) \neq \perp$  und eine TM  $\tau$  mit  $f = h_\tau$

- Wir konstruieren eine Turingmaschine  $\tau'$  mit  $M = \text{range}(h_{\tau'})$

Bei Eingabe einer (Repräsentation der) Zahl  $n$  arbeite  $\tau'$  wie folgt

1. Berechne  $i, j$  so daß  $n = \langle i, j \rangle$

2. Simuliere die Berechnung von  $\tau$  bei Eingabe  $i$  für (maximal)  $j$  Schritte  
(Erzeuge Anfangskonfiguration, simuliere  $\hat{\delta}_\tau$  auf separatem Band)

3. Falls  $\tau$  in  $j$  Schritten anhält, gebe  $i$  aus. Andernfalls gebe  $n_0$  aus

- $\tau'$  hält auf jeder Eingabe, also ist  $h_{\tau'}$  total

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS (II)

- $M = \text{domain}(f)$  für ein berechenbares  $f \Rightarrow M$  aufzählbar

- Es sei  $M = \text{domain}(f) = \{i \in \mathbb{N} \mid f(i) \neq \perp\}$  für ein berechenbares  $f$

- Falls  $M = \emptyset$ , dann ist  $M$  per Definition aufzählbar

✓

- Andernfalls gibt es ein  $n_0 \in \mathbb{N}$  mit  $f(n_0) \neq \perp$  und eine TM  $\tau$  mit  $f = h_\tau$

- Wir konstruieren eine Turingmaschine  $\tau'$  mit  $M = \text{range}(h_{\tau'})$

Bei Eingabe einer (Repräsentation der) Zahl  $n$  arbeite  $\tau'$  wie folgt

1. Berechne  $i, j$  so daß  $n = \langle i, j \rangle$

2. Simuliere die Berechnung von  $\tau$  bei Eingabe  $i$  für (maximal)  $j$  Schritte  
(Erzeuge Anfangskonfiguration, simuliere  $\hat{\delta}_\tau$  auf separatem Band)

3. Falls  $\tau$  in  $j$  Schritten anhält, gebe  $i$  aus. Andernfalls gebe  $n_0$  aus

- $\tau'$  hält auf jeder Eingabe, also ist  $h_{\tau'}$  total

- $M \subseteq \text{range}(h_{\tau'})$ : Sei  $i \in M$ . Dann gibt es ein  $j$  so daß  $\tau$  nach  $j$  Schritten hält.

Damit  $h_{\tau'}(\langle i, j \rangle) = i$ , also  $i \in \text{range}(h_{\tau'})$

# BEWEIS DER ÄQUIVALENZ DURCH RINGSCHLUSS (II)

- $M = \text{domain}(f)$  für ein berechenbares  $f \Rightarrow M$  aufzählbar

- Es sei  $M = \text{domain}(f) = \{i \in \mathbb{N} \mid f(i) \neq \perp\}$  für ein berechenbares  $f$

- Falls  $M = \emptyset$ , dann ist  $M$  per Definition aufzählbar

✓

- Andernfalls gibt es ein  $n_0 \in \mathbb{N}$  mit  $f(n_0) \neq \perp$  und eine TM  $\tau$  mit  $f = h_\tau$

- Wir konstruieren eine Turingmaschine  $\tau'$  mit  $M = \text{range}(h_{\tau'})$

Bei Eingabe einer (Repräsentation der) Zahl  $n$  arbeite  $\tau'$  wie folgt

1. Berechne  $i, j$  so daß  $n = \langle i, j \rangle$

2. Simuliere die Berechnung von  $\tau$  bei Eingabe  $i$  für (maximal)  $j$  Schritte  
(Erzeuge Anfangskonfiguration, simuliere  $\hat{\delta}_\tau$  auf separatem Band)

3. Falls  $\tau$  in  $j$  Schritten anhält, gebe  $i$  aus. Andernfalls gebe  $n_0$  aus

- $\tau'$  hält auf jeder Eingabe, also ist  $h_{\tau'}$  total

- $M \subseteq \text{range}(h_{\tau'})$ : Sei  $i \in M$ . Dann gibt es ein  $j$  so daß  $\tau$  nach  $j$  Schritten hält.

Damit  $h_{\tau'}(\langle i, j \rangle) = i$ , also  $i \in \text{range}(h_{\tau'})$

- $M \supseteq \text{range}(h_{\tau'})$ : Sei  $i \in \text{range}(h_{\tau'})$ . Falls  $i = n_0$  dann  $i \in M$  nach Voraussetzung.

Andernfalls gibt es ein  $j$  so daß  $\tau$  nach  $j$  Schritten hält.

Also  $i \in \text{domain}(h_\tau) = M$ .

✓

# BEISPIELE ENTSCHIEDBARER / AUFZÄHLBARER MENGEN

Sei  $f:\mathbb{N}\rightarrow\mathbb{N}$  total berechenbar,  $g:\mathbb{N}\rightarrow\mathbb{N}$  berechenbar

# BEISPIELE ENTSCHIEDBARER / AUFZÄHLBARER MENGEN

Sei  $f:\mathbb{N}\rightarrow\mathbb{N}$  **total berechenbar**,  $g:\mathbb{N}\rightarrow\mathbb{N}$  **berechenbar**

- **$\text{graph}(f) = \{(i, j) \mid f(i) = j\}$  ist entscheidbar**
  - Bei Eingabe  $(i, j)$  berechne  $f(i)$  (hält immer) und vergleiche mit  $j$

# BEISPIELE ENTSCHIEDBARER / AUFZÄHLBARER MENGEN

Sei  $f:\mathbb{N}\rightarrow\mathbb{N}$  **total berechenbar**,  $g:\mathbb{N}\rightarrow\mathbb{N}$  **berechenbar**

- **$\text{graph}(f) = \{(i, j) \mid f(i) = j\}$  ist entscheidbar**
  - Bei Eingabe  $(i, j)$  berechne  $f(i)$  (hält immer) und vergleiche mit  $j$
- **$\text{range}(f) = \{j \mid \exists i \in \mathbb{N} f(i) = j\}$  ist aufzählbar**
  - Charakterisierung #2

# BEISPIELE ENTSCHIEDBARER / AUFZÄHLBARER MENGEN

Sei  $f:\mathbb{N}\rightarrow\mathbb{N}$  **total berechenbar**,  $g:\mathbb{N}\rightarrow\mathbb{N}$  **berechenbar**

- **$\text{graph}(f) = \{(i, j) \mid f(i) = j\}$  ist entscheidbar**
  - Bei Eingabe  $(i, j)$  berechne  $f(i)$  (hält immer) und vergleiche mit  $j$
- **$\text{range}(f) = \{j \mid \exists i \in \mathbb{N} f(i) = j\}$  ist aufzählbar**
  - Charakterisierung #2
- **$\text{range}(g) = \{j \mid \exists i \in \mathbb{N} g(i) = j\}$  ist aufzählbar**
  - Charakterisierung #2

# BEISPIELE ENTSCHIEDBARER / AUFZÄHLBARER MENGEN

Sei  $f:\mathbb{N}\rightarrow\mathbb{N}$  **total berechenbar**,  $g:\mathbb{N}\rightarrow\mathbb{N}$  **berechenbar**

- **$\text{graph}(f) = \{(i, j) \mid f(i) = j\}$  ist entscheidbar**
  - Bei Eingabe  $(i, j)$  berechne  $f(i)$  (hält immer) und vergleiche mit  $j$
- **$\text{range}(f) = \{j \mid \exists i \in \mathbb{N} f(i) = j\}$  ist aufzählbar**
  - Charakterisierung #2
- **$\text{range}(g) = \{j \mid \exists i \in \mathbb{N} g(i) = j\}$  ist aufzählbar**
  - Charakterisierung #2
- **$\text{graph}(g) = \{(i, j) \mid g(i) = j\}$  ist aufzählbar**
  - Bei Eingabe  $(i, j)$  berechne  $g(i)$  (hält nicht immer) und vergleiche

# BEISPIELE ENTSCHIEDBARER / AUFZÄHLBARER MENGEN

Sei  $f:\mathbb{N}\rightarrow\mathbb{N}$  **total berechenbar**,  $g:\mathbb{N}\rightarrow\mathbb{N}$  **berechenbar**

- **$\text{graph}(f) = \{(i, j) \mid f(i) = j\}$  ist entscheidbar**
  - Bei Eingabe  $(i, j)$  berechne  $f(i)$  (hält immer) und vergleiche mit  $j$
- **$\text{range}(f) = \{j \mid \exists i \in \mathbb{N} f(i) = j\}$  ist aufzählbar**
  - Charakterisierung #2
- **$\text{range}(g) = \{j \mid \exists i \in \mathbb{N} g(i) = j\}$  ist aufzählbar**
  - Charakterisierung #2
- **$\text{graph}(g) = \{(i, j) \mid g(i) = j\}$  ist aufzählbar**
  - Bei Eingabe  $(i, j)$  berechne  $g(i)$  (hält nicht immer) und vergleiche
- **$\text{domain}(g) = \{i \mid g(i) \text{ hält}\}$  ist aufzählbar**
  - Charakterisierung #4

# AUFZÄHLBARKEIT VS. ENTSCHIEDBARKEIT

$M \subseteq \mathbb{N}$  entscheidbar  $\Leftrightarrow M$  und  $\overline{M}$  aufzählbar

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

$M \subseteq \mathbb{N}$  entscheidbar  $\Leftrightarrow M$  und  $\overline{M}$  aufzählbar

$\Rightarrow$  Es sei  $M$  entscheidbar. Dann ist  $\chi_M: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar.

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

$M \subseteq \mathbb{N}$  entscheidbar  $\Leftrightarrow M$  und  $\overline{M}$  aufzählbar

$\Rightarrow$  Es sei  $M$  entscheidbar. Dann ist  $\chi_M: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar.

Es ist  $\psi_M(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=1, \\ \perp & \text{sonst} \end{cases}$  und  $\psi_{\overline{M}}(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=0, \\ \perp & \text{sonst} \end{cases}$

# AUFZÄHLBARKEIT VS. ENTSCHIEDBARKEIT

**$M \subseteq \mathbb{N}$  entscheidbar  $\Leftrightarrow M$  und  $\overline{M}$  aufzählbar**

$\Rightarrow$  Es sei  $M$  entscheidbar. Dann ist  $\chi_M: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar.

Es ist  $\psi_M(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=1, \\ \perp & \text{sonst} \end{cases}$ , und  $\psi_{\overline{M}}(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=0, \\ \perp & \text{sonst} \end{cases}$

Damit sind  $\psi_M$  und  $\psi_{\overline{M}}$  berechenbar, also  $M$  und  $\overline{M}$  aufzählbar ✓

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

$M \subseteq \mathbb{N}$  entscheidbar  $\Leftrightarrow M$  und  $\overline{M}$  aufzählbar

$\Rightarrow$  Es sei  $M$  entscheidbar. Dann ist  $\chi_M: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar.

Es ist  $\psi_M(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=1, \\ \perp & \text{sonst} \end{cases}$  und  $\psi_{\overline{M}}(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=0, \\ \perp & \text{sonst} \end{cases}$

Damit sind  $\psi_M$  und  $\psi_{\overline{M}}$  berechenbar, also  $M$  und  $\overline{M}$  aufzählbar ✓

$\Leftarrow$  Seien  $M$  und  $\overline{M}$  aufzählbar.

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

$M \subseteq \mathbb{N}$  entscheidbar  $\Leftrightarrow M$  und  $\overline{M}$  aufzählbar

$\Rightarrow$  Es sei  $M$  entscheidbar. Dann ist  $\chi_M: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar.

Es ist  $\psi_M(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=1, \\ \perp & \text{sonst} \end{cases}$  und  $\psi_{\overline{M}}(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=0, \\ \perp & \text{sonst} \end{cases}$

Damit sind  $\psi_M$  und  $\psi_{\overline{M}}$  berechenbar, also  $M$  und  $\overline{M}$  aufzählbar ✓

$\Leftarrow$  Seien  $M$  und  $\overline{M}$  aufzählbar.

Falls  $M = \emptyset$  oder  $\overline{M} = \emptyset$ , so ist  $M$  trivialerweise entscheidbar ✓

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

$M \subseteq \mathbb{N}$  entscheidbar  $\Leftrightarrow M$  und  $\overline{M}$  aufzählbar

$\Rightarrow$  Es sei  $M$  entscheidbar. Dann ist  $\chi_M: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar.

Es ist  $\psi_M(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=1, \\ \perp & \text{sonst} \end{cases}$  und  $\psi_{\overline{M}}(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=0, \\ \perp & \text{sonst} \end{cases}$

Damit sind  $\psi_M$  und  $\psi_{\overline{M}}$  berechenbar, also  $M$  und  $\overline{M}$  aufzählbar ✓

$\Leftarrow$  Seien  $M$  und  $\overline{M}$  aufzählbar.

Falls  $M = \emptyset$  oder  $\overline{M} = \emptyset$ , so ist  $M$  trivialerweise entscheidbar ✓

Andernfalls  $M = \text{range}(f)$  und  $\overline{M} = \text{range}(g)$  wobei  $f, g$  total berechenbar.

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

$M \subseteq \mathbb{N}$  entscheidbar  $\Leftrightarrow M$  und  $\overline{M}$  aufzählbar

$\Rightarrow$  Es sei  $M$  entscheidbar. Dann ist  $\chi_M: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar.

Es ist  $\psi_M(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=1, \\ \perp & \text{sonst} \end{cases}$  und  $\psi_{\overline{M}}(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=0, \\ \perp & \text{sonst} \end{cases}$

Damit sind  $\psi_M$  und  $\psi_{\overline{M}}$  berechenbar, also  $M$  und  $\overline{M}$  aufzählbar ✓

$\Leftarrow$  Seien  $M$  und  $\overline{M}$  aufzählbar.

Falls  $M = \emptyset$  oder  $\overline{M} = \emptyset$ , so ist  $M$  trivialerweise entscheidbar ✓

Andernfalls  $M = \text{range}(f)$  und  $\overline{M} = \text{range}(g)$  wobei  $f, g$  total berechenbar.

Wir konstruieren eine Turingmaschine  $\tau$  mit  $\chi_M = h_\tau$

- Bei Eingabe von  $n$  berechne  $j = \min\{i \mid f(i)=n \vee g(i)=n\}$
- Falls  $f(j)=n$ , gebe 1 aus, ansonsten 0

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

$M \subseteq \mathbb{N}$  entscheidbar  $\Leftrightarrow M$  und  $\overline{M}$  aufzählbar

$\Rightarrow$  Es sei  $M$  entscheidbar. Dann ist  $\chi_M: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar.

Es ist  $\psi_M(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=1, \\ \perp & \text{sonst} \end{cases}$  und  $\psi_{\overline{M}}(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=0, \\ \perp & \text{sonst} \end{cases}$

Damit sind  $\psi_M$  und  $\psi_{\overline{M}}$  berechenbar, also  $M$  und  $\overline{M}$  aufzählbar ✓

$\Leftarrow$  Seien  $M$  und  $\overline{M}$  aufzählbar.

Falls  $M = \emptyset$  oder  $\overline{M} = \emptyset$ , so ist  $M$  trivialerweise entscheidbar ✓

Andernfalls  $M = \text{range}(f)$  und  $\overline{M} = \text{range}(g)$  wobei  $f, g$  total berechenbar.

Wir konstruieren eine Turingmaschine  $\tau$  mit  $\chi_M = h_\tau$

– Bei Eingabe von  $n$  berechne  $j = \min\{i \mid f(i)=n \vee g(i)=n\}$

– Falls  $f(j)=n$ , gebe 1 aus, ansonsten 0

$\tau$  hält für jedes  $n$ , da  $n \in \text{range}(f)$  oder  $n \in \text{range}(g)$  ( $n \in M$  /  $n \notin M$ )

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

$M \subseteq \mathbb{N}$  entscheidbar  $\Leftrightarrow M$  und  $\overline{M}$  aufzählbar

$\Rightarrow$  Es sei  $M$  entscheidbar. Dann ist  $\chi_M: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar.

Es ist  $\psi_M(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=1, \\ \perp & \text{sonst} \end{cases}$  und  $\psi_{\overline{M}}(n) = \begin{cases} 1 & \text{falls } \chi_M(n)=0, \\ \perp & \text{sonst} \end{cases}$

Damit sind  $\psi_M$  und  $\psi_{\overline{M}}$  berechenbar, also  $M$  und  $\overline{M}$  aufzählbar ✓

$\Leftarrow$  Seien  $M$  und  $\overline{M}$  aufzählbar.

Falls  $M = \emptyset$  oder  $\overline{M} = \emptyset$ , so ist  $M$  trivialerweise entscheidbar ✓

Andernfalls  $M = \text{range}(f)$  und  $\overline{M} = \text{range}(g)$  wobei  $f, g$  total berechenbar.

Wir konstruieren eine Turingmaschine  $\tau$  mit  $\chi_M = h_\tau$

– Bei Eingabe von  $n$  berechne  $j = \min\{i \mid f(i)=n \vee g(i)=n\}$

– Falls  $f(j)=n$ , gebe 1 aus, ansonsten 0

$\tau$  hält für jedes  $n$ , da  $n \in \text{range}(f)$  oder  $n \in \text{range}(g)$  ( $n \in M$  /  $n \notin M$ )

Für  $n \in M$  gilt  $f(j)=n$  für ein  $j$ , also  $h_\tau(n)=1$ , ansonsten  $h_\tau(n)=0$  ✓

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

- Jede entscheidbare Menge ist aufzählbar

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

- Jede **entscheidbare Menge ist aufzählbar**
  - Die **Umkehrung gilt nicht** (Beispiel folgt später)

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

- Jede **entscheidbare Menge ist aufzählbar**
  - Die **Umkehrung gilt nicht** (Beispiel folgt später)
- Ist  $M$  **endlich**, so ist  $M$  **entscheidbar und aufzählbar**

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

- Jede **entscheidbare Menge ist aufzählbar**
  - Die **Umkehrung gilt nicht** (Beispiel folgt später)
- Ist  $M$  **endlich**, so ist  $M$  **entscheidbar und aufzählbar**
  - Für  $M = \{x_1, \dots, x_n\}$  ist  $\chi_M(n) = \begin{cases} 1 & n=x_1 \vee \dots \vee n=x_n, \\ 0 & \text{sonst} \end{cases}$
  - $\chi_M$  ist berechenbar, also ist  $M$  entscheidbar



# AUFZÄHLBARKEIT VS. ENTSCHIEDBARKEIT

- Jede **entscheidbare Menge ist aufzählbar**
  - Die **Umkehrung gilt nicht** (Beispiel folgt später)
- Ist  $M$  **endlich**, so ist  $M$  **entscheidbar und aufzählbar**
  - Für  $M = \{x_1, \dots, x_n\}$  ist  $\chi_M(n) = \begin{cases} 1 & n=x_1 \vee \dots \vee n=x_n, \\ 0 & \text{sonst} \end{cases}$
  - $\chi_M$  ist berechenbar, also ist  $M$  **entscheidbar** ✓
- $M \subseteq \mathbb{N}$  **aufzählbar**  $\Leftrightarrow$  es gibt ein **entscheidbares**  $M' \subseteq \mathbb{N} \times \mathbb{N}$   
mit  $M = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in M'\}$

# AUFZÄHLBARKEIT VS. ENTSCHIEDBARKEIT

- Jede **entscheidbare Menge ist aufzählbar**
  - Die **Umkehrung gilt nicht** (Beispiel folgt später)
- Ist  $M$  **endlich**, so ist  $M$  **entscheidbar und aufzählbar**
  - Für  $M = \{x_1, \dots, x_n\}$  ist  $\chi_M(n) = \begin{cases} 1 & n=x_1 \vee \dots \vee n=x_n, \\ 0 & \text{sonst} \end{cases}$
  - $\chi_M$  ist berechenbar, also ist  $M$  **entscheidbar** ✓
- **$M \subseteq \mathbb{N}$  aufzählbar**  $\Leftrightarrow$  es gibt ein **entscheidbares**  $M' \subseteq \mathbb{N} \times \mathbb{N}$   
mit  **$M = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in M'\}$**   
 $\Rightarrow$  : Es sei  $M \subseteq \mathbb{N}$  aufzählbar

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

- Jede **entscheidbare Menge ist aufzählbar**
  - Die **Umkehrung gilt nicht** (Beispiel folgt später)
- Ist  $M$  **endlich**, so ist  $M$  **entscheidbar und aufzählbar**
  - Für  $M = \{x_1, \dots, x_n\}$  ist  $\chi_M(n) = \begin{cases} 1 & n=x_1 \vee \dots \vee n=x_n, \\ 0 & \text{sonst} \end{cases}$
  - $\chi_M$  ist berechenbar, also ist  $M$  **entscheidbar** ✓
- $M \subseteq \mathbb{N}$  **aufzählbar**  $\Leftrightarrow$  es gibt ein **entscheidbares**  $M' \subseteq \mathbb{N} \times \mathbb{N}$   
mit  $M = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in M'\}$ 
  - $\Rightarrow$ : Es sei  $M \subseteq \mathbb{N}$  aufzählbar
    - Falls  $M = \emptyset$ , so ist  $M = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in \emptyset\}$  ✓

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

- Jede **entscheidbare Menge ist aufzählbar**

- Die **Umkehrung gilt nicht** (Beispiel folgt später)

- Ist  $M$  **endlich**, so ist  $M$  **entscheidbar und aufzählbar**

- Für  $M = \{x_1, \dots, x_n\}$  ist  $\chi_M(n) = \begin{cases} 1 & n=x_1 \vee \dots \vee n=x_n, \\ 0 & \text{sonst} \end{cases}$

- $\chi_M$  ist berechenbar, also ist  $M$  entscheidbar



- $M \subseteq \mathbb{N}$  **aufzählbar**  $\Leftrightarrow$  es gibt ein **entscheidbares**  $M' \subseteq \mathbb{N} \times \mathbb{N}$   
mit  $M = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in M'\}$

$\Rightarrow$ : Es sei  $M \subseteq \mathbb{N}$  aufzählbar

- Falls  $M = \emptyset$ , so ist  $M = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in \emptyset\}$



- Andernfalls ist  $M = \text{range}(f)$  für ein berechenbares, totales  $f$

und  $\text{range}(f) = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in \text{graph}(f)\}$



# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

- Jede **entscheidbare Menge ist aufzählbar**

- Die **Umkehrung gilt nicht** (Beispiel folgt später)

- Ist  $M$  **endlich**, so ist  $M$  **entscheidbar und aufzählbar**

- Für  $M = \{x_1, \dots, x_n\}$  ist  $\chi_M(n) = \begin{cases} 1 & n=x_1 \vee \dots \vee n=x_n, \\ 0 & \text{sonst} \end{cases}$

- $\chi_M$  ist berechenbar, also ist  $M$  entscheidbar



- $M \subseteq \mathbb{N}$  **aufzählbar**  $\Leftrightarrow$  es gibt ein **entscheidbares**  $M' \subseteq \mathbb{N} \times \mathbb{N}$   
mit  $M = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in M'\}$

$\Rightarrow$ : Es sei  $M \subseteq \mathbb{N}$  aufzählbar

- Falls  $M = \emptyset$ , so ist  $M = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in \emptyset\}$



- Andernfalls ist  $M = \text{range}(f)$  für ein berechenbares, totales  $f$

und  $\text{range}(f) = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in \text{graph}(f)\}$



$\Leftarrow$ : Es sei  $M = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in M'\}$  für ein entscheidbares  $M'$

# AUFZÄHLBARKEIT VS. ENTSCHEIDBARKEIT

- Jede **entscheidbare Menge ist aufzählbar**

- Die **Umkehrung gilt nicht** (Beispiel folgt später)

- Ist  $M$  **endlich**, so ist  $M$  **entscheidbar und aufzählbar**

- Für  $M = \{x_1, \dots, x_n\}$  ist  $\chi_M(n) = \begin{cases} 1 & n=x_1 \vee \dots \vee n=x_n, \\ 0 & \text{sonst} \end{cases}$

- $\chi_M$  ist berechenbar, also ist  $M$  entscheidbar



- $M \subseteq \mathbb{N}$  **aufzählbar**  $\Leftrightarrow$  es gibt ein **entscheidbares**  $M' \subseteq \mathbb{N} \times \mathbb{N}$   
mit  $M = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in M'\}$

$\Rightarrow$ : Es sei  $M \subseteq \mathbb{N}$  aufzählbar

- Falls  $M = \emptyset$ , so ist  $M = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in \emptyset\}$



- Andernfalls ist  $M = \text{range}(f)$  für ein berechenbares, totales  $f$

- und  $\text{range}(f) = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in \text{graph}(f)\}$



$\Leftarrow$ : Es sei  $M = \{y \in \mathbb{N} \mid \exists n \in \mathbb{N} (n, y) \in M'\}$  für ein entscheidbares  $M'$

- Dann ist  $\psi_M(y) = \text{sign}(\min\{n \in \mathbb{N} \mid (n, y) \in M'\} + 1)$

- $= \text{sign}(\min\{n \in \mathbb{N} \mid \chi_{M'}(n, y) = 1\} + 1)$



# ABSCHLUSSEIGENSCHAFTEN

Sei  $M, M' \subseteq X^*$ ,  $f$  berechenbar und total,  $g$  berechenbar

# ABSCHLUSSEIGENSCHAFTEN

Sei  $M, M' \subseteq X^*$ ,  $f$  berechenbar und total,  $g$  berechenbar

- Sind  $M, M'$  **aufzählbar**, dann auch

$M \cup M'$ ,  $M \cap M'$ ,  $M \circ M'$ ,  $M^*$ ,  $g(M)$ ,  $g^{-1}(M)$

*Vereinigung, Durchschnitt, Konkatenation, Konkatenationsabschluß, Urbild*

*Abschluß unter Komplement oder Differenz gilt nicht*

# ABSCHLUSSEIGENSCHAFTEN

Sei  $M, M' \subseteq X^*$ ,  $f$  berechenbar und total,  $g$  berechenbar

- Sind  $M, M'$  **aufzählbar**, dann auch

$$M \cup M', M \cap M', M \circ M', M^*, g(M), g^{-1}(M)$$

*Vereinigung, Durchschnitt, Konkatenation, Konkatenationsabschluß, Urbild  
Abschluß unter Komplement oder Differenz gilt nicht*

- Sind  $M, M'$  **entscheidbar**, dann auch

$$M \cup M', M \cap M', M \setminus M', \overline{M}, M \circ M', M^*, f^{-1}(M)$$

*Vereinigung, Durchschnitt, Differenz, Komplement, Konkatenation, -abschluß, Urbild*

$$\begin{aligned} \overline{M} &= \{w \in X^* \mid w \notin M\} & M \circ M' &= \{wv \mid w \in M \wedge v \in M'\} & M^* &= \{w_1..w_n \mid w_i \in M\} \\ f^{-1}(M) &= \{w \in X^* \mid f(w) \in M\} & f(M) &= \{f(w) \mid w \in M\} \end{aligned}$$

# ABSCHLUSSEIGENSCHAFTEN

Sei  $M, M' \subseteq X^*$ ,  $f$  berechenbar und total,  $g$  berechenbar

- Sind  $M, M'$  **aufzählbar**, dann auch

$$M \cup M', M \cap M', M \circ M', M^*, g(M), g^{-1}(M)$$

*Vereinigung, Durchschnitt, Konkatenation, Konkatenationsabschluß, Urbild  
Abschluß unter Komplement oder Differenz gilt nicht*

- Sind  $M, M'$  **entscheidbar**, dann auch

$$M \cup M', M \cap M', M \setminus M', \overline{M}, M \circ M', M^*, f^{-1}(M)$$

*Vereinigung, Durchschnitt, Differenz, Komplement, Konkatenation, -abschluß, Urbild*

**Aussage gilt analog für Teilmengen von  $\mathbb{N}, \mathbb{N}^k, \dots$**

$$\begin{aligned} \overline{M} &= \{w \in X^* \mid w \notin M\} & M \circ M' &= \{wv \mid w \in M \wedge v \in M'\} & M^* &= \{w_1..w_n \mid w_i \in M\} \\ f^{-1}(M) &= \{w \in X^* \mid f(w) \in M\} & f(M) &= \{f(w) \mid w \in M\} \end{aligned}$$

# BEWEIS DER ABSCHLUSSEIGENSCHAFTEN

- Es seien  $M, M'$  aufzählbar mit Funktionen  $f$  und  $f'$

$M \cup M'$ :

$M \cap M'$ :

$M \circ M'$ :

$M^*$ :

$g(M)$ :

$g^{-1}(M)$ :

# BEWEIS DER ABSCHLUSSEIGENSCHAFTEN

- Es seien  $M, M'$  aufzählbar mit Funktionen  $f$  und  $f'$

$M \cup M'$ : Für  $h(w) = \begin{cases} f(v) & \text{falls } w=0v, \\ f'(v) & \text{falls } w=1v \end{cases}$  ist  $\text{range}(h) = M \cup M'$

$M \cap M'$ :

$M \circ M'$ :

$M^*$ :

$g(M)$ :

$g^{-1}(M)$ :

# BEWEIS DER ABSCHLUSSEIGENSCHAFTEN

- Es seien  $M, M'$  aufzählbar mit Funktionen  $f$  und  $f'$

$M \cup M'$ : Für  $h(w) = \begin{cases} f(v) & \text{falls } w=0v, \\ f'(v) & \text{falls } w=1v \end{cases}$  ist  $\text{range}(h) = M \cup M'$

$M \cap M'$ :  $\psi_{M \cap M'}(w) = \psi_M(w) * \psi_{M'}(w)$

$M \circ M'$ :

$M^*$ :

$g(M)$ :

$g^{-1}(M)$ :

# BEWEIS DER ABSCHLUSSEIGENSCHAFTEN

- Es seien  $M, M'$  aufzählbar mit Funktionen  $f$  und  $f'$

$M \cup M'$ : Für  $h(w) = \begin{cases} f(v) & \text{falls } w=0v, \\ f'(v) & \text{falls } w=1v \end{cases}$  ist  $\text{range}(h) = M \cup M'$

$M \cap M'$ :  $\psi_{M \cap M'}(w) = \psi_M(w) * \psi_{M'}(w)$

$M \circ M'$ : Für  $h(w \# v) = f(w) \circ f'(v)$  ist  $\text{range}(h) = M \circ M'$

$M^*$ :

$g(M)$ :

$g^{-1}(M)$ :

# BEWEIS DER ABSCHLUSSEIGENSCHAFTEN

- Es seien  $M, M'$  aufzählbar mit Funktionen  $f$  und  $f'$

$M \cup M'$ : Für  $h(w) = \begin{cases} f(v) & \text{falls } w=0v, \\ f'(v) & \text{falls } w=1v \end{cases}$  ist  $\text{range}(h) = M \cup M'$

$M \cap M'$ :  $\psi_{M \cap M'}(w) = \psi_M(w) * \psi_{M'}(w)$

$M \circ M'$ : Für  $h(w \# v) = f(w) \circ f'(v)$  ist  $\text{range}(h) = M \circ M'$

$M^*$ : Für  $h(w_1 \# \dots \# w_n) = f(w_1) \circ \dots \circ f(w_n)$  ist  $\text{range}(h) = M^*$

$g(M)$ :

$g^{-1}(M)$ :

# BEWEIS DER ABSCHLUSSEIGENSCHAFTEN

- Es seien  $M, M'$  aufzählbar mit Funktionen  $f$  und  $f'$

$M \cup M'$ : Für  $h(w) = \begin{cases} f(v) & \text{falls } w=0v, \\ f'(v) & \text{falls } w=1v \end{cases}$  ist  $\text{range}(h) = M \cup M'$

$M \cap M'$ :  $\psi_{M \cap M'}(w) = \psi_M(w) * \psi_{M'}(w)$

$M \circ M'$ : Für  $h(w \# v) = f(w) \circ f'(v)$  ist  $\text{range}(h) = M \circ M'$

$M^*$ : Für  $h(w_1 \# \dots \# w_n) = f(w_1) \circ \dots \circ f(w_n)$  ist  $\text{range}(h) = M^*$

$g(M)$ : Für  $h(w) = g(f(w))$  ist  $\text{range}(h) = g(M)$

$g^{-1}(M)$ :

# BEWEIS DER ABSCHLUSSEIGENSCHAFTEN

- Es seien  $M, M'$  aufzählbar mit Funktionen  $f$  und  $f'$

$M \cup M'$ : Für  $h(w) = \begin{cases} f(v) & \text{falls } w=0v, \\ f'(v) & \text{falls } w=1v \end{cases}$  ist  $\text{range}(h) = M \cup M'$

$M \cap M'$ :  $\psi_{M \cap M'}(w) = \psi_M(w) * \psi_{M'}(w)$

$M \circ M'$ : Für  $h(w \# v) = f(w) \circ f'(v)$  ist  $\text{range}(h) = M \circ M'$

$M^*$ : Für  $h(w_1 \# \dots \# w_n) = f(w_1) \circ \dots \circ f(w_n)$  ist  $\text{range}(h) = M^*$

$g(M)$ : Für  $h(w) = g(f(w))$  ist  $\text{range}(h) = g(M)$

$g^{-1}(M)$ :  $\psi_{g^{-1}(M)}(w) = \psi_M(g(w))$

# BEWEIS DER ABSCHLUSSEIGENSCHAFTEN

- Es seien  $M, M'$  aufzählbar mit Funktionen  $f$  und  $f'$

$$M \cup M': \text{ Für } h(w) = \begin{cases} f(v) & \text{falls } w=0v, \\ f'(v) & \text{falls } w=1v \end{cases} \text{ ist } \text{range}(h) = M \cup M'$$

$$M \cap M': \psi_{M \cap M'}(w) = \psi_M(w) * \psi_{M'}(w)$$

$$M \circ M': \text{ Für } h(w \# v) = f(w) \circ f'(v) \text{ ist } \text{range}(h) = M \circ M'$$

$$M^*: \text{ Für } h(w_1 \# \dots \# w_n) = f(w_1) \circ \dots \circ f(w_n) \text{ ist } \text{range}(h) = M^*$$

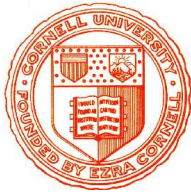
$$g(M): \text{ Für } h(w) = g(f(w)) \text{ ist } \text{range}(h) = g(M)$$

$$g^{-1}(M): \psi_{g^{-1}(M)}(w) = \psi_M(g(w))$$

- Entscheidbare Mengen sind abgeschlossen unter  $\cup, \cap, \setminus, \bar{\phantom{x}}, \circ, *, f^{-1}$

*Beweis in Übungen*

# Theoretische Informatik



## Einheit 3.2

### Universelle Maschinen



1. Standardnumerierung berechenbarer Funktionen
2. Universelle Funktion
3. Grundeigenschaften berechenbarer Funktionen

# NOCH OFFENE FRAGEN

- Gibt es **unentscheidbare Mengen**?
  - Unentscheidbar aber aufzählbar?
  - Nicht aufzählbar'?

# NOCH OFFENE FRAGEN

- Gibt es **unentscheidbare Mengen**?
  - Unentscheidbar aber aufzählbar?
  - Nicht aufzählbar'?
- Gibt es **unberechenbare Funktionen**?

# NOCH OFFENE FRAGEN

- **Gibt es unentscheidbare Mengen?**
  - Unentscheidbar aber aufzählbar?
  - Nicht aufzählbar'?
- **Gibt es unberechenbare Funktionen?**
- **Wie beweist man Unlösbarkeit?**
  - **Kardinalitätsargument**: es gibt mehr Funktionen als Programme
  - Konkretes **Gegenbeispiel** konstruieren

# NOCH OFFENE FRAGEN

- **Gibt es unentscheidbare Mengen?**
  - Unentscheidbar aber aufzählbar?
  - Nicht aufzählbar'?
- **Gibt es unberechenbare Funktionen?**
- **Wie beweist man Unlösbarkeit?**
  - **Kardinalitätsargument**: es gibt mehr Funktionen als Programme
  - Konkretes **Gegenbeispiel** konstruieren
- **Was benötigt man für diese Argumente?**
  - Präzisierung der **Grundannahmen** zur Berechenbarkeit
  - Nachweis, daß diese Grundannahmen erfüllt sind

- **Programme und Daten sind als Zahlen codierbar**
  - Programme und Daten werden als Worte dargestellt
  - Worte, die Programme darstellen, können durchnumeriert werden
  - $\varphi_i$ : Berechnete Funktion des Programms  $i$  ( $\varphi_i: \mathbb{N} \rightarrow \mathbb{N}$ )
  - $\Phi_i$ : Rechenzeitfunktion zum Programm  $i$  ( $\text{domain}(\Phi_i) = \text{domain}(\varphi_i)$ )

- **Programme und Daten sind als Zahlen codierbar**

- Programme und Daten werden als Worte dargestellt
- Worte, die Programme darstellen, können durchnummeriert werden
- $\varphi_i$ : Berechnete Funktion des Programms  $i$  ( $\varphi_i: \mathbb{N} \rightarrow \mathbb{N}$ )
- $\Phi_i$ : Rechenzeitfunktion zum Programm  $i$  ( $\text{domain}(\Phi_i) = \text{domain}(\varphi_i)$ )

- **Computer sind universelle Maschinen**

- Bei Eingabe beliebiger Programme und Daten berechnen sie das Ergebnis
- Die Funktion  $u: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  mit  $u(i, n) = \varphi_i(n)$  ist berechenbar

# GRUNDANNAHMEN ÜBER BERECHNUNGEN

- **Programme und Daten sind als Zahlen codierbar**

- Programme und Daten werden als Worte dargestellt
- Worte, die Programme darstellen, können durchnumeriert werden
- $\varphi_i$ : Berechnete Funktion des Programms  $i$  ( $\varphi_i: \mathbb{N} \rightarrow \mathbb{N}$ )
- $\Phi_i$ : Rechenzeitfunktion zum Programm  $i$  ( $\text{domain}(\Phi_i) = \text{domain}(\varphi_i)$ )

- **Computer sind universelle Maschinen**

- Bei Eingabe beliebiger Programme und Daten berechnen sie das Ergebnis
- Die Funktion  $u: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  mit  $u(i, n) = \varphi_i(n)$  ist berechenbar

- **Man kann Programme effektiv zusammensetzen**

- Die Nummer des entstehenden Programms kann berechnet werden
- Es gibt eine berechenbare totale Funktion  $h$  mit  $\varphi_{h(i,j)} = \varphi_i \circ \varphi_j$

# GRUNDANNAHMEN ÜBER BERECHNUNGEN

- **Programme und Daten sind als Zahlen codierbar**

- Programme und Daten werden als Worte dargestellt
- Worte, die Programme darstellen, können durchnumeriert werden
- $\varphi_i$ : Berechnete Funktion des Programms  $i$  ( $\varphi_i: \mathbb{N} \rightarrow \mathbb{N}$ )
- $\Phi_i$ : Rechenzeitfunktion zum Programm  $i$  ( $\text{domain}(\Phi_i) = \text{domain}(\varphi_i)$ )

- **Computer sind universelle Maschinen**

- Bei Eingabe beliebiger Programme und Daten berechnen sie das Ergebnis
- Die Funktion  $u: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  mit  $u(i, n) = \varphi_i(n)$  ist berechenbar

- **Man kann Programme effektiv zusammensetzen**

- Die Nummer des entstehenden Programms kann berechnet werden
- Es gibt eine berechenbare totale Funktion  $h$  mit  $\varphi_{h(i,j)} = \varphi_i \circ \varphi_j$

- **Rechenzeit ist entscheidbar**

- Man kann für beliebige  $i, n, t \in \mathbb{N}$  testen ob  $\Phi_i(n) = t$  ist oder nicht

# NUMERIERUNG VON TURINGMASCHINEN

- Codierung aller Alphabete in einem Alphabet  $\hat{\Gamma}$

# NUMERIERUNG VON TURINGMASCHINEN

- Codierung aller Alphabete in einem Alphabet  $\hat{\Gamma}$
- Codiere Zustandsüberföhrungsfunktion  $\delta$  als Wort in  $\hat{\Gamma}^*$

# NUMERIERUNG VON TURINGMASCHINEN

- Codierung aller Alphabete in einem Alphabet  $\hat{\Gamma}$
- Codiere Zustandsüberföhrungsfunktion  $\delta$  als Wort in  $\hat{\Gamma}^*$
- Codiere Komponenten der Turingmaschine  $\tau$  als Wort in  $\hat{\Gamma}^*$

# NUMERIERUNG VON TURINGMASCHINEN

- Codierung aller Alphabete in einem Alphabet  $\hat{\Gamma}$
- Codiere Zustandsüberföhrungsfunktion  $\delta$  als Wort in  $\hat{\Gamma}^*$
- Codiere Komponenten der Turingmaschine  $\tau$  als Wort in  $\hat{\Gamma}^*$
- Numeriere Worte über  $X = \{x_1, \dots, x_n\}$  gemäß der **lexikographischen Ordnung**

$$\epsilon < x_1 < \dots < x_n < x_1x_1 < x_1x_2 < \dots < x_nx_n < x_1x_1x_1 < \dots$$

# NUMERIERUNG VON TURINGMASCHINEN

- **Codierung aller Alphabete** in einem Alphabet  $\hat{\Gamma}$
- **Codiere Zustandsüberföhrungsfunktion  $\delta$**  als Wort in  $\hat{\Gamma}^*$
- **Codiere Komponenten der Turingmaschine  $\tau$**  als Wort in  $\hat{\Gamma}^*$
- **Numeriere Worte über  $X = \{x_1, \dots, x_n\}$  gemäß der lexikographischen Ordnung**

$$\epsilon < x_1 < \dots < x_n < x_1x_1 < x_1x_2 < \dots < x_nx_n < x_1x_1x_1 < \dots$$

- **Numeriere Worte, die Turingmaschinen codieren**
  - Benutze Numerierung aller Worte
  - Man kann testen, ob ein Wort  $w \in \hat{\Gamma}^*$  ein Turingprogramm beschreibt
  - $\tau_i$ : Turingmaschine  $\tau$  mit  $w_\tau = \nu(n_\tau(i))$  “die  $i$ -te Turingmaschine”
  - **Gödelnummer** der Turingmaschine  $\tau$ : Zahl  $i$  mit  $\tau = \tau_i$

## ● Berechenbare Funktionen auf Worten

- $\hat{\varphi}_i \equiv h_{\tau_i}$ : “die von der  $i$ -ten Turingmaschine berechnete Funktion”

## ● Berechenbare Funktionen auf Worten

–  $\hat{\varphi}_i \equiv h_{\tau_i}$ : “die von der  $i$ -ten Turingmaschine berechnete Funktion”

–  $t_i$ : **Schrittzahlfunktion** der Turingmaschine  $\tau_i$

$$t_i(w) = \begin{cases} m & \text{falls Berechnung von } \tau_i(w) \text{ in } m \text{ Schritten terminiert,} \\ \perp & \text{sonst} \end{cases}$$

## ● Berechenbare Funktionen auf Worten

–  $\hat{\varphi}_i \equiv h_{\tau_i}$ : “die von der  $i$ -ten Turingmaschine berechnete Funktion”

–  $t_i$ : **Schrittzahlfunktion** der Turingmaschine  $\tau_i$

$$t_i(w) = \begin{cases} m & \text{falls Berechnung von } \tau_i(w) \text{ in } m \text{ Schritten terminiert,} \\ \perp & \text{sonst} \end{cases}$$

## ● Berechenbare Funktionen auf Zahlen

–  $\varphi_i \equiv r^{-1} \circ \hat{\varphi}_i \circ r : \mathbb{N} \rightarrow \mathbb{N}$  “die  $i$ -te berechenbare Funktion”

$r: \mathbb{N} \rightarrow X^*$  bijektive Repräsentation von Zahlen als Worte

–  $\Phi_i \equiv t_i \circ r : \mathbb{N} \rightarrow \mathbb{N}$  “Schrittzahlfunktion von  $\varphi_i$ ”

## ● Berechenbare Funktionen auf Worten

–  $\hat{\varphi}_i \equiv h_{\tau_i}$ : “die von der  $i$ -ten Turingmaschine berechnete Funktion”

–  $t_i$ : **Schrittzahlfunktion** der Turingmaschine  $\tau_i$

$$t_i(w) = \begin{cases} m & \text{falls Berechnung von } \tau_i(w) \text{ in } m \text{ Schritten terminiert,} \\ \perp & \text{sonst} \end{cases}$$

## ● Berechenbare Funktionen auf Zahlen

–  $\varphi_i \equiv r^{-1} \circ \hat{\varphi}_i \circ r : \mathbb{N} \rightarrow \mathbb{N}$  “die  $i$ -te berechenbare Funktion”

$r: \mathbb{N} \rightarrow X^*$  bijektive Repräsentation von Zahlen als Worte

–  $\Phi_i \equiv t_i \circ r : \mathbb{N} \rightarrow \mathbb{N}$  “Schrittzahlfunktion von  $\varphi_i$ ”

## ● Eigenschaften von $\varphi$ und $\Phi$

–  $\varphi$  is **surjektiv**, aber nicht bijektiv

–  $\text{domain}(\Phi_i) = \text{domain}(\varphi_i)$  ( $\Phi_i$  terminiert auf den gleichen Eingaben wie  $\varphi_i$ )

–  $\{(i, n, t) \mid \Phi_i(n)=t\}$  ist **entscheidbar** “Rechenzeit ist entscheidbar”

# NUMERIERUNG BERECHENBARER FUNKTIONEN

## ● Berechenbare Funktionen auf Worten

–  $\hat{\varphi}_i \equiv h_{\tau_i}$ : “die von der  $i$ -ten Turingmaschine berechnete Funktion”

–  $t_i$ : **Schrittzahlfunktion** der Turingmaschine  $\tau_i$

$$t_i(w) = \begin{cases} m & \text{falls Berechnung von } \tau_i(w) \text{ in } m \text{ Schritten terminiert,} \\ \perp & \text{sonst} \end{cases}$$

## ● Berechenbare Funktionen auf Zahlen

–  $\varphi_i \equiv r^{-1} \circ \hat{\varphi}_i \circ r : \mathbb{N} \rightarrow \mathbb{N}$  “die  $i$ -te berechenbare Funktion”

$r: \mathbb{N} \rightarrow X^*$  bijektive Repräsentation von Zahlen als Worte

–  $\Phi_i \equiv t_i \circ r : \mathbb{N} \rightarrow \mathbb{N}$  “Schrittzahlfunktion von  $\varphi_i$ ”

## ● Eigenschaften von $\varphi$ und $\Phi$

–  $\varphi$  is **surjektiv**, aber nicht bijektiv

–  $\text{domain}(\Phi_i) = \text{domain}(\varphi_i)$  ( $\Phi_i$  terminiert auf den gleichen Eingaben wie  $\varphi_i$ )

–  $\{(i, n, t) \mid \Phi_i(n)=t\}$  ist **entscheidbar** “Rechenzeit ist entscheidbar”

**Die Numerierung berechenbarer Funktionen ist nur surjektiv**

# UNIVERSELLE TURINGMASCHINEN

Kann man alle Turingprogramme auf einer  
einzigsten Maschine ausführen?

Kann man alle Turingprogramme auf einer einzigen Maschine ausführen?

- **Universelle Maschinen**

- $\tau_u$  ist universell, wenn  $h_{\tau_u}(w_\tau, v) = h_\tau(v)$  für jede TM  $\tau$  und jedes  $v \in X^*$

Kann man alle Turingprogramme auf einer einzigen Maschine ausführen?

## ● Universelle Maschinen

- $\tau_u$  ist universell, wenn  $h_{\tau_u}(w_\tau, v) = h_\tau(v)$  für jede TM  $\tau$  und jedes  $v \in X^*$
- Insbesondere  $h_{\tau_u}(r(i), v) = h_{\tau_i}(v)$  für alle  $i, v$  ( $r: \mathbb{N} \rightarrow X^*$  Zahlendarstellung)

Kann man alle Turingprogramme auf einer einzigen Maschine ausführen?

## ● Universelle Maschinen

- $\tau_u$  ist universell, wenn  $h_{\tau_u}(w_\tau, v) = h_\tau(v)$  für jede TM  $\tau$  und jedes  $v \in X^*$
- Insbesondere  $h_{\tau_u}(r(i), v) = h_{\tau_i}(v)$  für alle  $i, v$  ( $r: \mathbb{N} \rightarrow X^*$  Zahlendarstellung)

## ● Universelle Funktionen

- $u: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  ist universell, wenn  $u(i, n) = \varphi_i(n)$  für alle  $i, n \in \mathbb{N}$

Kann man alle Turingprogramme auf einer einzigen Maschine ausführen?

## ● Universelle Maschinen

- $\tau_u$  ist universell, wenn  $h_{\tau_u}(w_\tau, v) = h_\tau(v)$  für jede TM  $\tau$  und jedes  $v \in X^*$
- Insbesondere  $h_{\tau_u}(r(i), v) = h_{\tau_i}(v)$  für alle  $i, v$  ( $r: \mathbb{N} \rightarrow X^*$  Zahlendarstellung)

## ● Universelle Funktionen

- $u: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  ist universell, wenn  $u(i, n) = \varphi_i(n)$  für alle  $i, n \in \mathbb{N}$

## ● Gibt es universelle Maschinen?

## Kann man alle Turingprogramme auf einer einzigen Maschine ausführen?

### ● Universelle Maschinen

- $\tau_u$  ist universell, wenn  $h_{\tau_u}(w_\tau, v) = h_\tau(v)$  für jede TM  $\tau$  und jedes  $v \in X^*$
- Insbesondere  $h_{\tau_u}(r(i), v) = h_{\tau_i}(v)$  für alle  $i, v$  ( $r: \mathbb{N} \rightarrow X^*$  Zahlendarstellung)

### ● Universelle Funktionen

- $u: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  ist universell, wenn  $u(i, n) = \varphi_i(n)$  für alle  $i, n \in \mathbb{N}$

### ● Gibt es universelle Maschinen?

- Die Numerierung  $n_\tau$  ist berechenbar

## Kann man alle Turingprogramme auf einer einzigen Maschine ausführen?

### ● Universelle Maschinen

- $\tau_u$  ist universell, wenn  $h_{\tau_u}(w_\tau, v) = h_\tau(v)$  für jede TM  $\tau$  und jedes  $v \in X^*$
- Insbesondere  $h_{\tau_u}(r(i), v) = h_{\tau_i}(v)$  für alle  $i, v$  ( $r: \mathbb{N} \rightarrow X^*$  Zahlendarstellung)

### ● Universelle Funktionen

- $u: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  ist universell, wenn  $u(i, n) = \varphi_i(n)$  für alle  $i, n \in \mathbb{N}$

### ● Gibt es universelle Maschinen?

- Die Numerierung  $n_\tau$  ist berechenbar
- Turingprogramme lassen sich simulieren

## Kann man alle Turingprogramme auf einer einzigen Maschine ausführen?

### ● Universelle Maschinen

- $\tau_u$  ist universell, wenn  $h_{\tau_u}(w_\tau, v) = h_\tau(v)$  für jede TM  $\tau$  und jedes  $v \in X^*$
- Insbesondere  $h_{\tau_u}(r(i), v) = h_{\tau_i}(v)$  für alle  $i, v$  ( $r: \mathbb{N} \rightarrow X^*$  Zahlendarstellung)

### ● Universelle Funktionen

- $u: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  ist universell, wenn  $u(i, n) = \varphi_i(n)$  für alle  $i, n \in \mathbb{N}$

### ● Gibt es universelle Maschinen?

- Die Numerierung  $n_\tau$  ist berechenbar
- Turingprogramme lassen sich simulieren
- Baue universelle Maschine mit  $\nu \circ n_\tau$  und Einzelschrittsimulation

## Kann man alle Turingprogramme auf einer einzigen Maschine ausführen?

### ● Universelle Maschinen

- $\tau_u$  ist universell, wenn  $h_{\tau_u}(w_\tau, v) = h_\tau(v)$  für jede TM  $\tau$  und jedes  $v \in X^*$
- Insbesondere  $h_{\tau_u}(r(i), v) = h_{\tau_i}(v)$  für alle  $i, v$  ( $r: \mathbb{N} \rightarrow X^*$  Zahlendarstellung)

### ● Universelle Funktionen

- $u: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  ist universell, wenn  $u(i, n) = \varphi_i(n)$  für alle  $i, n \in \mathbb{N}$

### ● Gibt es universelle Maschinen?

- Die Numerierung  $n_\tau$  ist berechenbar
- Turingprogramme lassen sich simulieren
- Baue universelle Maschine mit  $\nu \circ n_\tau$  und Einzelschrittsimulation

Details z.B. in Hopcroft, Motwani, Ullman, Seite 387–389

# DAS ÜBERSETZUNGSLEMMA

**Turingmaschinen sind effektiv kombinierbar**

# DAS ÜBERSETZUNGSLEMMA

Turingmaschinen sind effektiv kombinierbar

- Kombiniere  $\tau_1$  und  $\tau_2$  zu  $\tau$  mit  $h_\tau = h_{\tau_1} \circ h_{\tau_2}$

# DAS ÜBERSETZUNGSLEMMA

## Turingmaschinen sind effektiv kombinierbar

- **Kombiniere  $\tau_1$  und  $\tau_2$  zu  $\tau$  mit  $h_\tau = h_{\tau_1} \circ h_{\tau_2}$** 
  - Umbenennung der Zustände von  $\tau_2$
  - Springe vom “Endzustand” von  $\tau_1$  zum Anfangszustand von  $\tau_2$

# DAS ÜBERSETZUNGSLEMMA

## Turingmaschinen sind effektiv kombinierbar

- **Kombiniere  $\tau_1$  und  $\tau_2$  zu  $\tau$  mit  $h_\tau = h_{\tau_1} \circ h_{\tau_2}$** 
  - Umbenennung der Zustände von  $\tau_2$
  - Springe vom “Endzustand” von  $\tau_1$  zum Anfangszustand von  $\tau_2$
  - Programm  $w_\tau$  kann aus  $w_{\tau_1}$  und  $w_{\tau_2}$  berechnet werden

# DAS ÜBERSETZUNGSLEMMA

## Turingmaschinen sind effektiv kombinierbar

- **Kombiniere  $\tau_1$  und  $\tau_2$  zu  $\tau$  mit  $h_\tau = h_{\tau_1} \circ h_{\tau_2}$** 
  - Umbenennung der Zustände von  $\tau_2$
  - Springe vom “Endzustand” von  $\tau_1$  zum Anfangszustand von  $\tau_2$
  - Programm  $w_\tau$  kann aus  $w_{\tau_1}$  und  $w_{\tau_2}$  berechnet werden
  - Gödelnummer  $k$  von  $\tau$  kann aus denen für  $\tau_1$  und  $\tau_2$  berechnet werden

# DAS ÜBERSETZUNGSLEMMA

## Turingmaschinen sind effektiv kombinierbar

- **Kombiniere  $\tau_1$  und  $\tau_2$  zu  $\tau$  mit  $h_\tau = h_{\tau_1} \circ h_{\tau_2}$** 
  - Umbenennung der Zustände von  $\tau_2$
  - Springe vom “Endzustand” von  $\tau_1$  zum Anfangszustand von  $\tau_2$
  - Programm  $w_\tau$  kann aus  $w_{\tau_1}$  und  $w_{\tau_2}$  berechnet werden
  - Gödelnummer  $k$  von  $\tau$  kann aus denen für  $\tau_1$  und  $\tau_2$  berechnet werden
- **Kombiniere  $\varphi_i$  und  $\varphi_j$  zu  $\varphi_k$  mit  $\varphi_k = \varphi_i \circ \varphi_j$**

# DAS ÜBERSETZUNGSLEMMA

## Turingmaschinen sind effektiv kombinierbar

- **Kombiniere  $\tau_1$  und  $\tau_2$  zu  $\tau$  mit  $h_\tau = h_{\tau_1} \circ h_{\tau_2}$** 
  - Umbenennung der Zustände von  $\tau_2$
  - Springe vom “Endzustand” von  $\tau_1$  zum Anfangszustand von  $\tau_2$
  - Programm  $w_\tau$  kann aus  $w_{\tau_1}$  und  $w_{\tau_2}$  berechnet werden
  - Gödelnummer  $k$  von  $\tau$  kann aus denen für  $\tau_1$  und  $\tau_2$  berechnet werden
- **Kombiniere  $\varphi_i$  und  $\varphi_j$  zu  $\varphi_k$  mit  $\varphi_k = \varphi_i \circ \varphi_j$** 
  - Index  $k$  kann aus  $i$  und  $j$  berechnet werden

# DAS ÜBERSETZUNGSLEMMA

## Turingmaschinen sind effektiv kombinierbar

- **Kombiniere  $\tau_1$  und  $\tau_2$  zu  $\tau$  mit  $h_\tau = h_{\tau_1} \circ h_{\tau_2}$** 
  - Umbenennung der Zustände von  $\tau_2$
  - Springe vom “Endzustand” von  $\tau_1$  zum Anfangszustand von  $\tau_2$
  - Programm  $w_\tau$  kann aus  $w_{\tau_1}$  und  $w_{\tau_2}$  berechnet werden
  - Gödelnummer  $k$  von  $\tau$  kann aus denen für  $\tau_1$  und  $\tau_2$  berechnet werden
- **Kombiniere  $\varphi_i$  und  $\varphi_j$  zu  $\varphi_k$  mit  $\varphi_k = \varphi_i \circ \varphi_j$** 
  - Index  $k$  kann aus  $i$  und  $j$  berechnet werden
  - Es gibt eine berechenbare totale Funktion  $h$  mit  $\varphi_{h(i,j)} = \varphi_i \circ \varphi_j$

# DAS ÜBERSETZUNGSLEMMA

## Turingmaschinen sind effektiv kombinierbar

- **Kombiniere  $\tau_1$  und  $\tau_2$  zu  $\tau$  mit  $h_\tau = h_{\tau_1} \circ h_{\tau_2}$** 
  - Umbenennung der Zustände von  $\tau_2$
  - Springe vom “Endzustand” von  $\tau_1$  zum Anfangszustand von  $\tau_2$
  - Programm  $w_\tau$  kann aus  $w_{\tau_1}$  und  $w_{\tau_2}$  berechnet werden
  - Gödelnummer  $k$  von  $\tau$  kann aus denen für  $\tau_1$  und  $\tau_2$  berechnet werden
- **Kombiniere  $\varphi_i$  und  $\varphi_j$  zu  $\varphi_k$  mit  $\varphi_k = \varphi_i \circ \varphi_j$** 
  - Index  $k$  kann aus  $i$  und  $j$  berechnet werden
  - Es gibt eine berechenbare totale Funktion  $h$  mit  $\varphi_{h(i,j)} = \varphi_i \circ \varphi_j$
- **Allgemeinste Version: SMN Theorem**

# DAS ÜBERSETZUNGSLEMMA

## Turingmaschinen sind effektiv kombinierbar

- **Kombiniere  $\tau_1$  und  $\tau_2$  zu  $\tau$  mit  $h_\tau = h_{\tau_1} \circ h_{\tau_2}$** 
  - Umbenennung der Zustände von  $\tau_2$
  - Springe vom “Endzustand” von  $\tau_1$  zum Anfangszustand von  $\tau_2$
  - Programm  $w_\tau$  kann aus  $w_{\tau_1}$  und  $w_{\tau_2}$  berechnet werden
  - Gödelnummer  $k$  von  $\tau$  kann aus denen für  $\tau_1$  und  $\tau_2$  berechnet werden
- **Kombiniere  $\varphi_i$  und  $\varphi_j$  zu  $\varphi_k$  mit  $\varphi_k = \varphi_i \circ \varphi_j$** 
  - Index  $k$  kann aus  $i$  und  $j$  berechnet werden
  - Es gibt eine berechenbare totale Funktion  $h$  mit  $\varphi_{h(i,j)} = \varphi_i \circ \varphi_j$
- **Allgemeinste Version: SMN Theorem**
  - Es gibt eine berechenbare totale Funktion  $s$  mit  $\varphi_{s\langle m,n \rangle}(i) = \varphi_m\langle n, i \rangle$

# DAS ÜBERSETZUNGSLEMMA

## Turingmaschinen sind effektiv kombinierbar

- **Kombiniere  $\tau_1$  und  $\tau_2$  zu  $\tau$  mit  $h_\tau = h_{\tau_1} \circ h_{\tau_2}$** 
  - Umbenennung der Zustände von  $\tau_2$
  - Springe vom “Endzustand” von  $\tau_1$  zum Anfangszustand von  $\tau_2$
  - Programm  $w_\tau$  kann aus  $w_{\tau_1}$  und  $w_{\tau_2}$  berechnet werden
  - Gödelnummer  $k$  von  $\tau$  kann aus denen für  $\tau_1$  und  $\tau_2$  berechnet werden
- **Kombiniere  $\varphi_i$  und  $\varphi_j$  zu  $\varphi_k$  mit  $\varphi_k = \varphi_i \circ \varphi_j$** 
  - Index  $k$  kann aus  $i$  und  $j$  berechnet werden
  - Es gibt eine berechenbare totale Funktion  $h$  mit  $\varphi_{h(i,j)} = \varphi_i \circ \varphi_j$
- **Allgemeinste Version: SMN Theorem**
  - Es gibt eine berechenbare totale Funktion  $s$  mit  $\varphi_{s\langle m,n \rangle}(i) = \varphi_m\langle n, i \rangle$

Technisches Resultat mit wenig eigener Bedeutung

# ZUSAMMENFASSUNG: KERNAXIOME DER BERECHENBARKEITSTHEORIE

- **Berechenbare Funktionen sind effektiv numerierbar**
  - $\varphi_i : \mathbb{N} \rightarrow \mathbb{N}$ : berechnete Funktion des Programms  $i$
  - $\Phi_i$ : Rechenzeitfunktion zum Programm  $i$
  - **$\text{domain}(\Phi_i) = \text{domain}(\varphi_i)$**

# ZUSAMMENFASSUNG: KERNAXIOME DER BERECHENBAREITSTHEORIE

- **Berechenbare Funktionen sind effektiv numerierbar**
  - $\varphi_i: \mathbb{N} \rightarrow \mathbb{N}$ : berechnete Funktion des Programms  $i$
  - $\Phi_i$ : Rechenzeitfunktion zum Programm  $i$
  - **$\text{domain}(\Phi_i) = \text{domain}(\varphi_i)$**
- **Die Menge  $\{(i, n, t) \mid \Phi_i(n) = t\}$  ist entscheidbar**

# ZUSAMMENFASSUNG: KERNAXIOME DER BERECHENBAREITSTHEORIE

- **Berechenbare Funktionen sind effektiv numerierbar**
  - $\varphi_i : \mathbb{N} \rightarrow \mathbb{N}$ : berechnete Funktion des Programms  $i$
  - $\Phi_i$ : Rechenzeitfunktion zum Programm  $i$
  - **$\text{domain}(\Phi_i) = \text{domain}(\varphi_i)$**
- **Die Menge  $\{(i, n, t) \mid \Phi_i(n) = t\}$  ist entscheidbar**
- **Die universelle Funktion ist berechenbar**
  - $u : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  mit  $u(i, n) = \varphi_i(n)$  ist berechenbar

# ZUSAMMENFASSUNG: KERNAXIOME DER BERECHENBAREITSTHEORIE

- **Berechenbare Funktionen sind effektiv numerierbar**
  - $\varphi_i : \mathbb{N} \rightarrow \mathbb{N}$ : berechnete Funktion des Programms  $i$
  - $\Phi_i$ : Rechenzeitfunktion zum Programm  $i$
  - **$\text{domain}(\Phi_i) = \text{domain}(\varphi_i)$**
- **Die Menge  $\{(i, n, t) \mid \Phi_i(n) = t\}$  ist entscheidbar**
- **Die universelle Funktion ist berechenbar**
  - $u : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  mit  $u(i, n) = \varphi_i(n)$  ist berechenbar
- **Programme sind effektiv kombinierbar**
  - Es gibt eine berechenbare totale Funktion  $h$  mit  $\varphi_{h(i,j)} = \varphi_i \circ \varphi_j$
  - Es gibt eine berechenbare totale Funktion  $s$  mit  $\varphi_{s\langle m,n \rangle}(i) = \varphi_m\langle n, i \rangle$

# ZUSAMMENFASSUNG: KERNAXIOME DER BERECHENBAREITSTHEORIE

- **Berechenbare Funktionen sind effektiv numerierbar**
  - $\varphi_i : \mathbb{N} \rightarrow \mathbb{N}$ : berechnete Funktion des Programms  $i$
  - $\Phi_i$ : Rechenzeitfunktion zum Programm  $i$
  - $\text{domain}(\Phi_i) = \text{domain}(\varphi_i)$
- **Die Menge  $\{(i, n, t) \mid \Phi_i(n) = t\}$  ist entscheidbar**
- **Die universelle Funktion ist berechenbar**
  - $u : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  mit  $u(i, n) = \varphi_i(n)$  ist berechenbar
- **Programme sind effektiv kombinierbar**
  - Es gibt eine berechenbare totale Funktion  $h$  mit  $\varphi_{h(i,j)} = \varphi_i \circ \varphi_j$
  - Es gibt eine berechenbare totale Funktion  $s$  mit  $\varphi_{s\langle m,n \rangle}(i) = \varphi_m\langle n, i \rangle$

**Alles weitere folgt aus diesen Axiomen**

# WICHTIGE ENTSCHIEDBARE UND AUFZÄHLBARE MENGEN

- $\{(i, n, t) \mid \Phi_i(n) = t\}$  ist entscheidbar

# WICHTIGE ENTSCHIEDBARE UND AUFZÄHLBARE MENGEN

- $\{(i, n, t) \mid \Phi_i(n) = t\}$  ist entscheidbar
  - Kernaxiom der Berechenbarkeitstheorie

# WICHTIGE ENTSCHIEDBARE UND AUFZÄHLBARE MENGEN

- $\{(i, n, t) \mid \Phi_i(n) = t\}$  ist entscheidbar
  - Kernaxiom der Berechenbarkeitstheorie
- $\{(i, n, y) \mid \varphi_i(n) = y\}$  ist aufzählbar

# WICHTIGE ENTSCHIEDBARE UND AUFZÄHLBARE MENGEN

- $\{(i, n, t) \mid \Phi_i(n) = t\}$  ist entscheidbar
  - Kernaxiom der Berechenbarkeitstheorie
- $\{(i, n, y) \mid \varphi_i(n) = y\}$  ist aufzählbar
  - Graph der universellen Funktion

# WICHTIGE ENTSCHIEDBARE UND AUFZÄHLBARE MENGEN

- $\{(i, n, t) \mid \Phi_i(n) = t\}$  ist entscheidbar
  - Kernaxiom der Berechenbarkeitstheorie
- $\{(i, n, y) \mid \varphi_i(n) = y\}$  ist aufzählbar
  - Graph der universellen Funktion
- $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist aufzählbar

# WICHTIGE ENTSCHIEDBARE UND AUFZÄHLBARE MENGEN

- $\{(i, n, t) \mid \Phi_i(n) = t\}$  ist entscheidbar
  - Kernaxiom der Berechenbarkeitstheorie
- $\{(i, n, y) \mid \varphi_i(n) = y\}$  ist aufzählbar
  - Graph der universellen Funktion
- $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist aufzählbar
  - Haltebereich der universellen Funktion

# WICHTIGE ENTSCHIEDBARE UND AUFZÄHLBARE MENGEN

- $\{(i, n, t) \mid \Phi_i(n) = t\}$  ist entscheidbar
  - Kernaxiom der Berechenbarkeitstheorie
- $\{(i, n, y) \mid \varphi_i(n) = y\}$  ist aufzählbar
  - Graph der universellen Funktion
- $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist aufzählbar
  - Haltebereich der universellen Funktion
- $S = \{i \mid \varphi_i(i) \neq \perp\}$  ist aufzählbar

# WICHTIGE ENTSCHEIDBARE UND AUFZÄHLBARE MENGEN

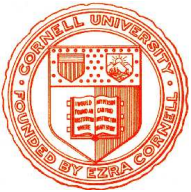
- $\{(i, n, t) \mid \Phi_i(n) = t\}$  ist entscheidbar
  - Kernaxiom der Berechenbarkeitstheorie
- $\{(i, n, y) \mid \varphi_i(n) = y\}$  ist aufzählbar
  - Graph der universellen Funktion
- $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist aufzählbar
  - Haltebereich der universellen Funktion
- $S = \{i \mid \varphi_i(i) \neq \perp\}$  ist aufzählbar
  - Haltebereich von  $\lambda i.u(i, i)$

# Theoretische Informatik

## Einheit 3.3

### Beweistechniken für unlösbare Probleme

1. Diagonalisierung
2. Monotonieargumente
3. Problemreduktion
4. Der Satz von Rice



# GRENZEN DER BERECHENBARKEIT

Wie beweist man die Unlösbarkeit eines Problems?

## Wie beweist man die Unlösbarkeit eines Problems?

### ● Diagonalisierung

- Zeige, daß eine Funktion von jeder berechenbaren Funktion an mindestens einer Stelle abweicht, also selbst nicht berechenbar sein kann

## Wie beweist man die Unlösbarkeit eines Problems?

- **Diagonalisierung**

- Zeige, daß eine Funktion **von jeder berechenbaren Funktion an mindestens einer Stelle abweicht**, also selbst nicht berechenbar sein kann

- **Wachstums- und Monotonieargumente**

- Zeige, daß eine Funktion **stärker wächst** als jede berechenbare Funktion

## Wie beweist man die Unlösbarkeit eines Problems?

### ● Diagonalisierung

- Zeige, daß eine Funktion **von jeder berechenbaren Funktion an mindestens einer Stelle abweicht**, also selbst nicht berechenbar sein kann

### ● Wachstums- und Monotonieargumente

- Zeige, daß eine Funktion **stärker wächst** als jede berechenbare Funktion

### ● Reduktionsmethode und Abschlußeigenschaften

- Zeige, daß Lösung des Problems **zu einer Lösung eines bekanntermaßen unlösbaren Problems führen würde**

## Wie beweist man die Unlösbarkeit eines Problems?

### ● Diagonalisierung

- Zeige, daß eine Funktion **von jeder berechenbaren Funktion an mindestens einer Stelle abweicht**, also selbst nicht berechenbar sein kann

### ● Wachstums- und Monotonieargumente

- Zeige, daß eine Funktion **stärker wächst** als jede berechenbare Funktion

### ● Reduktionsmethode und Abschlußeigenschaften

- Zeige, daß Lösung des Problems **zu einer Lösung eines bekanntermaßen unlösbaren Problems führen würde**

### ● Anwendung allgemeiner theoretischer Resultate

- Unlösbarkeit **folgt direkt aus bekannten Sätzen**

# DIAGONALISIERUNG

- **Ziel**

- Zeige, daß eine unendliche Menge  $M$  eine Eigenschaft  $P$  nicht besitzt
- z.B. Entscheidbarkeit, Aufzählbarkeit, Abzählbarkeit

# DIAGONALISIERUNG

- **Ziel**

- Zeige, daß eine unendliche Menge  $M$  eine Eigenschaft  $P$  nicht besitzt
- z.B. Entscheidbarkeit, Aufzählbarkeit, Abzählbarkeit

- **Methodik**

# DIAGONALISIERUNG

- **Ziel**

- Zeige, daß eine unendliche Menge  $M$  eine Eigenschaft  $P$  nicht besitzt
- z.B. Entscheidbarkeit, Aufzählbarkeit, Abzählbarkeit

- **Methodik**

- Wir nehmen an,  $M$  habe die Eigenschaft  $P$

# DIAGONALISIERUNG

## ● Ziel

- Zeige, daß eine unendliche Menge  $M$  eine Eigenschaft  $P$  nicht besitzt
- z.B. Entscheidbarkeit, Aufzählbarkeit, Abzählbarkeit

## ● Methodik

- Wir nehmen an,  $M$  habe die Eigenschaft  $P$
- Konstruiere ein Element  $x$ , das von allen Elementen von  $M$  verschieden ist

# DIAGONALISIERUNG

## ● Ziel

- Zeige, daß eine unendliche Menge  $M$  eine Eigenschaft  $P$  nicht besitzt
- z.B. Entscheidbarkeit, Aufzählbarkeit, Abzählbarkeit

## ● Methodik

- Wir nehmen an,  $M$  habe die Eigenschaft  $P$
- Konstruiere ein Element  $x$ , das von allen Elementen von  $M$  verschieden ist
- Zeige, daß  $x \in M$  aufgrund der Annahme gelten muß

# DIAGONALISIERUNG

## ● Ziel

- Zeige, daß eine unendliche Menge  $M$  eine Eigenschaft  $P$  nicht besitzt
- z.B. Entscheidbarkeit, Aufzählbarkeit, Abzählbarkeit

## ● Methodik

- Wir nehmen an,  $M$  habe die Eigenschaft  $P$
- Konstruiere ein Element  $x$ , das von allen Elementen von  $M$  verschieden ist
- Zeige, daß  $x \in M$  aufgrund der Annahme gelten muß
- Aus dem Widerspruch folgt, daß die Annahme nicht gelten kann

# DIAGONALISIERUNG

## ● Ziel

- Zeige, daß eine unendliche Menge  $M$  eine Eigenschaft  $P$  nicht besitzt
- z.B. Entscheidbarkeit, Aufzählbarkeit, Abzählbarkeit

## ● Methodik

- Wir nehmen an,  $M$  habe die Eigenschaft  $P$
- Konstruiere ein Element  $x$ , das von allen Elementen von  $M$  verschieden ist
- Zeige, daß  $x \in M$  aufgrund der Annahme gelten muß
- Aus dem Widerspruch folgt, daß die Annahme nicht gelten kann

## ● Konstruktion des neuen Elementes

**Cantor'sches Diagonalverfahren:**

# DIAGONALISIERUNG

## ● Ziel

- Zeige, daß eine unendliche Menge  $M$  eine Eigenschaft  $P$  nicht besitzt
- z.B. Entscheidbarkeit, Aufzählbarkeit, Abzählbarkeit

## ● Methodik

- Wir nehmen an,  $M$  habe die Eigenschaft  $P$
- Konstruiere ein Element  $x$ , das von allen Elementen von  $M$  verschieden ist
- Zeige, daß  $x \in M$  aufgrund der Annahme gelten muß
- Aus dem Widerspruch folgt, daß die Annahme nicht gelten kann

## ● Konstruktion des neuen Elementes

### Cantor'sches Diagonalverfahren:

- Trage alle Elemente von  $M$  als Zeilen einer Tabelle auf

# DIAGONALISIERUNG

## ● Ziel

- Zeige, daß eine unendliche Menge  $M$  eine Eigenschaft  $P$  nicht besitzt
- z.B. Entscheidbarkeit, Aufzählbarkeit, Abzählbarkeit

## ● Methodik

- Wir nehmen an,  $M$  habe die Eigenschaft  $P$
- Konstruiere ein Element  $x$ , das von allen Elementen von  $M$  verschieden ist
- Zeige, daß  $x \in M$  aufgrund der Annahme gelten muß
- Aus dem Widerspruch folgt, daß die Annahme nicht gelten kann

## ● Konstruktion des neuen Elementes

### Cantor'sches Diagonalverfahren:

- Trage alle Elemente von  $M$  als Zeilen einer Tabelle auf
- Konstruiere  $x$  auf Diagonale mit Abweichung an jedem Punkt

# DIAGONALISIERUNG

## ● Ziel

- Zeige, daß eine unendliche Menge  $M$  eine Eigenschaft  $P$  nicht besitzt
- z.B. Entscheidbarkeit, Aufzählbarkeit, Abzählbarkeit

## ● Methodik

- Wir nehmen an,  $M$  habe die Eigenschaft  $P$
- Konstruiere ein Element  $x$ , das von allen Elementen von  $M$  verschieden ist
- Zeige, daß  $x \in M$  aufgrund der Annahme gelten muß
- Aus dem Widerspruch folgt, daß die Annahme nicht gelten kann

## ● Konstruktion des neuen Elementes

### Cantor'sches Diagonalverfahren:

- Trage alle Elemente von  $M$  als Zeilen einer Tabelle auf
- Konstruiere  $x$  auf Diagonale mit Abweichung an jedem Punkt
- Also kann  $x$  nicht als Zeile vorkommen

# DIAGONALBEWEISE I: ÜBERABZÄHLBARKEIT VON $\mathbb{N} \rightarrow \mathbb{N}$

- **Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  “überabzählbar” unendlich**
  - Die Menge aller Funktionen über  $\mathbb{N}$  kann nicht durchnumeriert werden

# DIAGONALBEWEISE I: ÜBERABZÄHLBARKEIT VON $\mathbb{N} \rightarrow \mathbb{N}$

- **Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  “überabzählbar” unendlich**
  - Die Menge aller Funktionen über  $\mathbb{N}$  kann nicht durchnumeriert werden
  - **Annahme:**  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar

# DIAGONALBEWEISE I: ÜBERABZÄHLBARKEIT VON $\mathbb{N} \rightarrow \mathbb{N}$

- Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  “überabzählbar” unendlich

- Die Menge aller Funktionen über  $\mathbb{N}$  kann nicht durchnumeriert werden

- **Annahme:**  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar

- Dann können alle Funktionen über  $\mathbb{N}$  in eine Tabelle eingetragen werden

|          | 0        | 1        | 2        | 3        | 4        | ... |
|----------|----------|----------|----------|----------|----------|-----|
| $f_0$    | $f_0(0)$ | $f_0(1)$ | $f_0(2)$ | $f_0(3)$ | $f_0(4)$ | ... |
| $f_1$    | $f_1(0)$ | $f_1(1)$ | $f_1(2)$ | $f_1(3)$ | $f_1(4)$ | ... |
| $f_2$    | $f_2(0)$ | $f_2(1)$ | $f_2(2)$ | $f_2(3)$ | $f_2(4)$ | ... |
| $f_3$    | $f_3(0)$ | $f_3(1)$ | $f_3(2)$ | $f_3(3)$ | $f_3(4)$ | ... |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | ... |

# DIAGONALBEWEISE I: ÜBERABZÄHLBARKEIT VON $\mathbb{N} \rightarrow \mathbb{N}$

- Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  “überabzählbar” unendlich

- Die Menge aller Funktionen über  $\mathbb{N}$  kann nicht durchnumeriert werden

- **Annahme:**  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar

- Dann können alle Funktionen über  $\mathbb{N}$  in eine Tabelle eingetragen werden

|          | 0          | 1        | 2        | 3        | 4        | ... |
|----------|------------|----------|----------|----------|----------|-----|
| $f_0$    | $f_0(0)+1$ | $f_0(1)$ | $f_0(2)$ | $f_0(3)$ | $f_0(4)$ | ... |
| $f_1$    | $f_1(0)$   | $f_1(1)$ | $f_1(2)$ | $f_1(3)$ | $f_1(4)$ | ... |
| $f_2$    | $f_2(0)$   | $f_2(1)$ | $f_2(2)$ | $f_2(3)$ | $f_2(4)$ | ... |
| $f_3$    | $f_3(0)$   | $f_3(1)$ | $f_3(2)$ | $f_3(3)$ | $f_3(4)$ | ... |
| $\vdots$ | $\vdots$   | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | ... |

- Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch  $f(x) = f_x(x) + 1$

# DIAGONALBEWEISE I: ÜBERABZÄHLBARKEIT VON $\mathbb{N} \rightarrow \mathbb{N}$

- Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  “überabzählbar” unendlich

- Die Menge aller Funktionen über  $\mathbb{N}$  kann nicht durchnumeriert werden

- **Annahme:**  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar

- Dann können alle Funktionen über  $\mathbb{N}$  in eine Tabelle eingetragen werden

|          | 0          | 1          | 2        | 3        | 4        | ... |
|----------|------------|------------|----------|----------|----------|-----|
| $f_0$    | $f_0(0)+1$ | $f_0(1)$   | $f_0(2)$ | $f_0(3)$ | $f_0(4)$ | ... |
| $f_1$    | $f_1(0)$   | $f_1(1)+1$ | $f_1(2)$ | $f_1(3)$ | $f_1(4)$ | ... |
| $f_2$    | $f_2(0)$   | $f_2(1)$   | $f_2(2)$ | $f_2(3)$ | $f_2(4)$ | ... |
| $f_3$    | $f_3(0)$   | $f_3(1)$   | $f_3(2)$ | $f_3(3)$ | $f_3(4)$ | ... |
| $\vdots$ | $\vdots$   | $\vdots$   | $\vdots$ | $\vdots$ | $\vdots$ | ... |

- Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch  $f(x) = f_x(x) + 1$

# DIAGONALBEWEISE I: ÜBERABZÄHLBARKEIT VON $\mathbb{N} \rightarrow \mathbb{N}$

- Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  “überabzählbar” unendlich

- Die Menge aller Funktionen über  $\mathbb{N}$  kann nicht durchnumeriert werden

- **Annahme:**  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar

- Dann können alle Funktionen über  $\mathbb{N}$  in eine Tabelle eingetragen werden

|          | 0          | 1          | 2          | 3        | 4        | ... |
|----------|------------|------------|------------|----------|----------|-----|
| $f_0$    | $f_0(0)+1$ | $f_0(1)$   | $f_0(2)$   | $f_0(3)$ | $f_0(4)$ | ... |
| $f_1$    | $f_1(0)$   | $f_1(1)+1$ | $f_1(2)$   | $f_1(3)$ | $f_1(4)$ | ... |
| $f_2$    | $f_2(0)$   | $f_2(1)$   | $f_2(2)+1$ | $f_2(3)$ | $f_2(4)$ | ... |
| $f_3$    | $f_3(0)$   | $f_3(1)$   | $f_3(2)$   | $f_3(3)$ | $f_3(4)$ | ... |
| $\vdots$ | $\vdots$   | $\vdots$   | $\vdots$   | $\vdots$ | $\vdots$ | ... |

- Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch  $f(x) = f_x(x) + 1$

# DIAGONALBEWEISE I: ÜBERABZÄHLBARKEIT VON $\mathbb{N} \rightarrow \mathbb{N}$

- Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  “überabzählbar” unendlich

- Die Menge aller Funktionen über  $\mathbb{N}$  kann nicht durchnumeriert werden

- **Annahme:**  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar

- Dann können alle Funktionen über  $\mathbb{N}$  in eine Tabelle eingetragen werden

|          | 0          | 1          | 2          | 3          | 4        | ... |
|----------|------------|------------|------------|------------|----------|-----|
| $f_0$    | $f_0(0)+1$ | $f_0(1)$   | $f_0(2)$   | $f_0(3)$   | $f_0(4)$ | ... |
| $f_1$    | $f_1(0)$   | $f_1(1)+1$ | $f_1(2)$   | $f_1(3)$   | $f_1(4)$ | ... |
| $f_2$    | $f_2(0)$   | $f_2(1)$   | $f_2(2)+1$ | $f_2(3)$   | $f_2(4)$ | ... |
| $f_3$    | $f_3(0)$   | $f_3(1)$   | $f_3(2)$   | $f_3(3)+1$ | $f_3(4)$ | ... |
| $\vdots$ | $\vdots$   | $\vdots$   | $\vdots$   | $\vdots$   | $\vdots$ | ... |

- Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch  $f(x) = f_x(x) + 1$

# DIAGONALBEWEISE I: ÜBERABZÄHLBARKEIT VON $\mathbb{N} \rightarrow \mathbb{N}$

- Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  “überabzählbar” unendlich

- Die Menge aller Funktionen über  $\mathbb{N}$  kann nicht durchnumeriert werden

- **Annahme:**  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar

- Dann können alle Funktionen über  $\mathbb{N}$  in eine Tabelle eingetragen werden

|          | 0          | 1          | 2          | 3          | 4        | ... |
|----------|------------|------------|------------|------------|----------|-----|
| $f_0$    | $f_0(0)+1$ | $f_0(1)$   | $f_0(2)$   | $f_0(3)$   | $f_0(4)$ | ... |
| $f_1$    | $f_1(0)$   | $f_1(1)+1$ | $f_1(2)$   | $f_1(3)$   | $f_1(4)$ | ... |
| $f_2$    | $f_2(0)$   | $f_2(1)$   | $f_2(2)+1$ | $f_2(3)$   | $f_2(4)$ | ... |
| $f_3$    | $f_3(0)$   | $f_3(1)$   | $f_3(2)$   | $f_3(3)+1$ | $f_3(4)$ | ... |
| $\vdots$ | $\vdots$   | $\vdots$   | $\vdots$   | $\vdots$   | $\vdots$ | ... |

- Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch  $f(x) = f_x(x) + 1$

- $f$  ist offensichtlich total, kann aber in der Tabelle nicht vorkommen

# DIAGONALBEWEISE I: ÜBERABZÄHLBARKEIT VON $\mathbb{N} \rightarrow \mathbb{N}$

- Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  “überabzählbar” unendlich

- Die Menge aller Funktionen über  $\mathbb{N}$  kann nicht durchnumeriert werden

- **Annahme:**  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar

- Dann können alle Funktionen über  $\mathbb{N}$  in eine Tabelle eingetragen werden

|          | 0          | 1          | 2          | 3          | 4        | ... |
|----------|------------|------------|------------|------------|----------|-----|
| $f_0$    | $f_0(0)+1$ | $f_0(1)$   | $f_0(2)$   | $f_0(3)$   | $f_0(4)$ | ... |
| $f_1$    | $f_1(0)$   | $f_1(1)+1$ | $f_1(2)$   | $f_1(3)$   | $f_1(4)$ | ... |
| $f_2$    | $f_2(0)$   | $f_2(1)$   | $f_2(2)+1$ | $f_2(3)$   | $f_2(4)$ | ... |
| $f_3$    | $f_3(0)$   | $f_3(1)$   | $f_3(2)$   | $f_3(3)+1$ | $f_3(4)$ | ... |
| $\vdots$ | $\vdots$   | $\vdots$   | $\vdots$   | $\vdots$   | $\vdots$ | ... |

- Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch  $f(x) = f_x(x) + 1$

- $f$  ist offensichtlich total, kann aber in der Tabelle nicht vorkommen

- Ansonsten wäre  $f = f_i$  für ein  $i$  und  $f_i(i) = f(i) = f_i(i) + 1$



# EXISTENZ UNBERECHENBARER FUNKTIONEN

- **Abstraktes Argument: es gibt zu viele Funktionen**

# EXISTENZ UNBERECHENBARER FUNKTIONEN

- **Abstraktes Argument: es gibt zu viele Funktionen**
  - Die Menge der berechenbaren Funktionen in  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar
  - Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  ist nicht abzählbar

- **Abstraktes Argument: es gibt zu viele Funktionen**

- Die Menge der berechenbaren Funktionen in  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar
- Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  ist nicht abzählbar
- Es gibt **mehr Funktionen** als es berechenbare Funktionen geben kann

- **Abstraktes Argument: es gibt zu viele Funktionen**

- Die Menge der berechenbaren Funktionen in  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar
- Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  ist nicht abzählbar
- Es gibt mehr Funktionen als es berechenbare Funktionen geben kann
- Es gibt nichtberechenbare Funktionen auf  $\mathbb{N} \rightarrow \mathbb{N}$



# EXISTENZ UNBERECHENBARER FUNKTIONEN

- **Abstraktes Argument: es gibt zu viele Funktionen**

- Die Menge der berechenbaren Funktionen in  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar
- Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  ist nicht abzählbar
- Es gibt mehr Funktionen als es berechenbare Funktionen geben kann
- Es gibt nichtberechenbare Funktionen auf  $\mathbb{N} \rightarrow \mathbb{N}$



- **Konkretes Argument: Angabe eines Beispiels**

# EXISTENZ UNBERECHENBARER FUNKTIONEN

- **Abstraktes Argument: es gibt zu viele Funktionen**
  - Die Menge der berechenbaren Funktionen in  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar
  - Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  ist nicht abzählbar
  - Es gibt mehr Funktionen als es berechenbare Funktionen geben kann
  - Es gibt nichtberechenbare Funktionen auf  $\mathbb{N} \rightarrow \mathbb{N}$  ✓
- **Konkretes Argument: Angabe eines Beispiels**
  - Das Halteproblem  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist unentscheidbar (Beweis folgt)

# EXISTENZ UNBERECHENBARER FUNKTIONEN

## ● Abstraktes Argument: es gibt zu viele Funktionen

- Die Menge der berechenbaren Funktionen in  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar
- Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  ist nicht abzählbar
- Es gibt mehr Funktionen als es berechenbare Funktionen geben kann
- Es gibt nichtberechenbare Funktionen auf  $\mathbb{N} \rightarrow \mathbb{N}$  ✓

## ● Konkretes Argument: Angabe eines Beispiels

- Das Halteproblem  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist unentscheidbar (Beweis folgt)
- Die charakteristische Funktion  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  ist nicht berechenbar ✓

# EXISTENZ UNBERECHENBARER FUNKTIONEN

## ● Abstraktes Argument: es gibt zu viele Funktionen

- Die Menge der berechenbaren Funktionen in  $\mathbb{N} \rightarrow \mathbb{N}$  ist abzählbar
- Die Menge  $\mathbb{N} \rightarrow \mathbb{N}$  ist nicht abzählbar
- Es gibt mehr Funktionen als es berechenbare Funktionen geben kann
- Es gibt nichtberechenbare Funktionen auf  $\mathbb{N} \rightarrow \mathbb{N}$  ✓

## ● Konkretes Argument: Angabe eines Beispiels

- Das Halteproblem  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist unentscheidbar (Beweis folgt)
- Die charakteristische Funktion  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  ist nicht berechenbar ✓
- Weitere konkrete Beispiele folgen

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

– Dann ist  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, wobei  $\chi_H(i, n) = \begin{cases} 1 & \text{wenn } \varphi_i(n) \text{ hält} \\ 0 & \text{sonst} \end{cases}$

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

– Dann ist  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, wobei  $\chi_H(i, n) = \begin{cases} 1 & \text{wenn } \varphi_i(n) \text{ h\"alt} \\ 0 & \text{sonst} \end{cases}$

|             | 0 | 1 | 2 | 3 | 4 | ... |
|-------------|---|---|---|---|---|-----|
| $\varphi_0$ | × | × | × | ⊥ | × | ... |
| $\varphi_1$ | ⊥ | ⊥ | × | × | × | ... |
| $\varphi_2$ | × | × | ⊥ | × | × | ... |
| $\varphi_3$ | ⊥ | × | ⊥ | × | ⊥ | ... |
| ⋮           | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ... |

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

– Dann ist  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, wobei  $\chi_H(i, n) = \begin{cases} 1 & \text{wenn } \varphi_i(n) \text{ hält} \\ 0 & \text{sonst} \end{cases}$

– Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch

$$f(n) := \begin{cases} 0 & \text{wenn } \varphi_n(n) \text{ nicht hält} \\ \perp & \text{sonst} \end{cases}$$

|             | 0 | 1 | 2 | 3 | 4 | ... |
|-------------|---|---|---|---|---|-----|
| $\varphi_0$ | × | × | × | ⊥ | × | ... |
| $\varphi_1$ | ⊥ | ⊥ | × | × | × | ... |
| $\varphi_2$ | × | × | ⊥ | × | × | ... |
| $\varphi_3$ | ⊥ | × | ⊥ | × | ⊥ | ... |
| ⋮           | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ... |

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

– Dann ist  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, wobei  $\chi_H(i, n) = \begin{cases} 1 & \text{wenn } \varphi_i(n) \text{ hält} \\ 0 & \text{sonst} \end{cases}$

– Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch

$$f(n) := \begin{cases} 0 & \text{wenn } \varphi_n(n) \text{ nicht hält} \\ \perp & \text{sonst} \end{cases}$$

|             | 0        | 1        | 2        | 3        | 4        | ... |
|-------------|----------|----------|----------|----------|----------|-----|
| $\varphi_0$ | $\perp$  | $\times$ | $\times$ | $\perp$  | $\times$ | ... |
| $\varphi_1$ | $\perp$  | $\perp$  | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_2$ | $\times$ | $\times$ | $\perp$  | $\times$ | $\times$ | ... |
| $\varphi_3$ | $\perp$  | $\times$ | $\perp$  | $\times$ | $\perp$  | ... |
| $\vdots$    | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | ... |

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

– Dann ist  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, wobei  $\chi_H(i, n) = \begin{cases} 1 & \text{wenn } \varphi_i(n) \text{ hält} \\ 0 & \text{sonst} \end{cases}$

– Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch

$$f(n) := \begin{cases} 0 & \text{wenn } \varphi_n(n) \text{ nicht hält} \\ \perp & \text{sonst} \end{cases}$$

|             | 0        | 1        | 2        | 3        | 4        | ... |
|-------------|----------|----------|----------|----------|----------|-----|
| $\varphi_0$ | $\perp$  | $\times$ | $\times$ | $\perp$  | $\times$ | ... |
| $\varphi_1$ | $\perp$  | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_2$ | $\times$ | $\times$ | $\perp$  | $\times$ | $\times$ | ... |
| $\varphi_3$ | $\perp$  | $\times$ | $\perp$  | $\times$ | $\perp$  | ... |
| $\vdots$    | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | ... |

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

– Dann ist  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, wobei  $\chi_H(i, n) = \begin{cases} 1 & \text{wenn } \varphi_i(n) \text{ hält} \\ 0 & \text{sonst} \end{cases}$

– Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch

$$f(n) := \begin{cases} 0 & \text{wenn } \varphi_n(n) \text{ nicht hält} \\ \perp & \text{sonst} \end{cases}$$

|             | 0        | 1        | 2        | 3        | 4        | ... |
|-------------|----------|----------|----------|----------|----------|-----|
| $\varphi_0$ | $\perp$  | $\times$ | $\times$ | $\perp$  | $\times$ | ... |
| $\varphi_1$ | $\perp$  | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_2$ | $\times$ | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_3$ | $\perp$  | $\times$ | $\perp$  | $\times$ | $\perp$  | ... |
| $\vdots$    | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | ... |

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

– Dann ist  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, wobei  $\chi_H(i, n) = \begin{cases} 1 & \text{wenn } \varphi_i(n) \text{ hält} \\ 0 & \text{sonst} \end{cases}$

– Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch

$$f(n) := \begin{cases} 0 & \text{wenn } \varphi_n(n) \text{ nicht hält} \\ \perp & \text{sonst} \end{cases}$$

|             | 0        | 1        | 2        | 3        | 4        | ... |
|-------------|----------|----------|----------|----------|----------|-----|
| $\varphi_0$ | $\perp$  | $\times$ | $\times$ | $\perp$  | $\times$ | ... |
| $\varphi_1$ | $\perp$  | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_2$ | $\times$ | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_3$ | $\perp$  | $\times$ | $\perp$  | $\perp$  | $\perp$  | ... |
| $\vdots$    | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | ... |

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

– Dann ist  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, wobei  $\chi_H(i, n) = \begin{cases} 1 & \text{wenn } \varphi_i(n) \text{ hält} \\ 0 & \text{sonst} \end{cases}$

– Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch

$$f(n) := \begin{cases} 0 & \text{wenn } \varphi_n(n) \text{ nicht hält} \\ \perp & \text{sonst} \end{cases}$$

|             | 0        | 1        | 2        | 3        | 4        | ... |
|-------------|----------|----------|----------|----------|----------|-----|
| $\varphi_0$ | $\perp$  | ×        | ×        | $\perp$  | ×        | ... |
| $\varphi_1$ | $\perp$  | ×        | ×        | ×        | ×        | ... |
| $\varphi_2$ | ×        | ×        | ×        | ×        | ×        | ... |
| $\varphi_3$ | $\perp$  | ×        | $\perp$  | $\perp$  | $\perp$  | ... |
| $\vdots$    | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | ... |

– Dann ist  $f$  berechenbar, denn  $f(n) = \mu_z[\chi_H(n, n) = 0]$

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

– Dann ist  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, wobei  $\chi_H(i, n) = \begin{cases} 1 & \text{wenn } \varphi_i(n) \text{ hält} \\ 0 & \text{sonst} \end{cases}$

– Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch

$$f(n) := \begin{cases} 0 & \text{wenn } \varphi_n(n) \text{ nicht hält} \\ \perp & \text{sonst} \end{cases}$$

|             | 0        | 1        | 2        | 3        | 4        | ... |
|-------------|----------|----------|----------|----------|----------|-----|
| $\varphi_0$ | $\perp$  | $\times$ | $\times$ | $\perp$  | $\times$ | ... |
| $\varphi_1$ | $\perp$  | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_2$ | $\times$ | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_3$ | $\perp$  | $\times$ | $\perp$  | $\perp$  | $\perp$  | ... |
| $\vdots$    | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | ... |

– Dann ist  $f$  berechenbar, denn  $f(n) = \mu_z[\chi_H(n, n) = 0]$

– Also gibt es ein  $i$  mit  $f = \varphi_i$

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

– Dann ist  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, wobei  $\chi_H(i, n) = \begin{cases} 1 & \text{wenn } \varphi_i(n) \text{ hält} \\ 0 & \text{sonst} \end{cases}$

– Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch

$$f(n) := \begin{cases} 0 & \text{wenn } \varphi_n(n) \text{ nicht hält} \\ \perp & \text{sonst} \end{cases}$$

|             | 0        | 1        | 2        | 3        | 4        | ... |
|-------------|----------|----------|----------|----------|----------|-----|
| $\varphi_0$ | $\perp$  | $\times$ | $\times$ | $\perp$  | $\times$ | ... |
| $\varphi_1$ | $\perp$  | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_2$ | $\times$ | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_3$ | $\perp$  | $\times$ | $\perp$  | $\perp$  | $\perp$  | ... |
| $\vdots$    | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | ... |

– Dann ist  $f$  berechenbar, denn  $f(n) = \mu_z[\chi_H(n, n) = 0]$

– Also gibt es ein  $i$  mit  $f = \varphi_i$

– Aber für dieses  $i$  gilt:  $\varphi_i(i) \text{ hält} \Leftrightarrow f(i) \text{ hält} \Leftrightarrow \varphi_i(i) \text{ hält nicht}$

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

– Dann ist  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, wobei  $\chi_H(i, n) = \begin{cases} 1 & \text{wenn } \varphi_i(n) \text{ h\"alt} \\ 0 & \text{sonst} \end{cases}$

– Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch

$$f(n) := \begin{cases} 0 & \text{wenn } \varphi_n(n) \text{ nicht h\"alt} \\ \perp & \text{sonst} \end{cases}$$

|             | 0        | 1        | 2        | 3        | 4        | ... |
|-------------|----------|----------|----------|----------|----------|-----|
| $\varphi_0$ | $\perp$  | $\times$ | $\times$ | $\perp$  | $\times$ | ... |
| $\varphi_1$ | $\perp$  | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_2$ | $\times$ | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_3$ | $\perp$  | $\times$ | $\perp$  | $\perp$  | $\perp$  | ... |
| $\vdots$    | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | ... |

– Dann ist  $f$  berechenbar, denn  $f(n) = \mu_z[\chi_H(n, n) = 0]$

– Also gibt es ein  $i$  mit  $f = \varphi_i$

– Aber für dieses  $i$  gilt:  $\varphi_i(i) \text{ h\"alt} \Leftrightarrow f(i) \text{ h\"alt} \Leftrightarrow \varphi_i(i) \text{ h\"alt nicht}$

– Dies ist ein Widerspruch, also ist die Annahme “ $H$  entscheidbar” falsch ✓

# DIAGONALBEWEISE II: UNENTSCHEIDBARKEIT DES HALTEPROBLEMS

**Annahme:**  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$  ist entscheidbar

– Dann ist  $\chi_H: \mathbb{N} \rightarrow \mathbb{N}$  berechenbar, wobei  $\chi_H(i, n) = \begin{cases} 1 & \text{wenn } \varphi_i(n) \text{ hält} \\ 0 & \text{sonst} \end{cases}$

– Definiere eine neue Funktion  $f: \mathbb{N} \rightarrow \mathbb{N}$  durch

$$f(n) := \begin{cases} 0 & \text{wenn } \varphi_n(n) \text{ nicht hält} \\ \perp & \text{sonst} \end{cases}$$

|             | 0        | 1        | 2        | 3        | 4        | ... |
|-------------|----------|----------|----------|----------|----------|-----|
| $\varphi_0$ | $\perp$  | $\times$ | $\times$ | $\perp$  | $\times$ | ... |
| $\varphi_1$ | $\perp$  | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_2$ | $\times$ | $\times$ | $\times$ | $\times$ | $\times$ | ... |
| $\varphi_3$ | $\perp$  | $\times$ | $\perp$  | $\perp$  | $\perp$  | ... |
| $\vdots$    | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | ... |

– Dann ist  $f$  berechenbar, denn  $f(n) = \mu_z[\chi_H(n, n) = 0]$

– Also gibt es ein  $i$  mit  $f = \varphi_i$

– Aber für dieses  $i$  gilt:  $\varphi_i(i) \text{ hält} \Leftrightarrow f(i) \text{ hält} \Leftrightarrow \varphi_i(i) \text{ hält nicht}$

– Dies ist ein Widerspruch, also ist die Annahme “ $H$  entscheidbar” falsch ✓

**Terminierung von Programmen ist nicht testbar**

# DIAGONALBEWEISE III: TOTAL BERECHENBARE FUNKTIONEN SIND NICHT AUFZÄHLBAR

Annahme:  $\mathcal{R}_\varphi = \{i \mid \varphi_i \text{ total}\}$  ist aufzählbar

## DIAGONALBEWEISE III:

# TOTAL BERECHENBARE FUNKTIONEN SIND NICHT AUFZÄHLBAR

**Annahme:**  $\mathcal{R}_\varphi = \{i \mid \varphi_i \text{ total}\}$  ist aufzählbar

- Dann gibt es eine berechenbare totale Funktion  $f$  mit  $\text{range}(f) = \mathcal{R}_\varphi$

## DIAGONALBEWEISE III:

# TOTAL BERECHENBARE FUNKTIONEN SIND NICHT AUFZÄHLBAR

**Annahme:**  $\mathcal{R}_\varphi = \{i \mid \varphi_i \text{ total}\}$  ist aufzählbar

- Dann gibt es eine berechenbare totale Funktion  $f$  mit  $\text{range}(f) = \mathcal{R}_\varphi$
- Dann lassen sich alle Funktionen aus  $\mathcal{R}$  wie folgt in eine Tabelle eintragen

|                  | 0                   | 1                   | 2                   | 3                   | 4                   | ... |
|------------------|---------------------|---------------------|---------------------|---------------------|---------------------|-----|
| $\varphi_{f(0)}$ | $\varphi_{f(0)}(0)$ | $\varphi_{f(0)}(1)$ | $\varphi_{f(0)}(2)$ | $\varphi_{f(0)}(3)$ | $\varphi_{f(0)}(4)$ | ... |
| $\varphi_{f(1)}$ | $\varphi_{f(1)}(0)$ | $\varphi_{f(1)}(1)$ | $\varphi_{f(1)}(2)$ | $\varphi_{f(1)}(3)$ | $\varphi_{f(1)}(4)$ | ... |
| $\varphi_{f(2)}$ | $\varphi_{f(2)}(0)$ | $\varphi_{f(2)}(1)$ | $\varphi_{f(2)}(2)$ | $\varphi_{f(2)}(3)$ | $\varphi_{f(2)}(4)$ | ... |
| $\varphi_{f(3)}$ | $\varphi_{f(3)}(0)$ | $\varphi_{f(3)}(1)$ | $\varphi_{f(3)}(2)$ | $\varphi_{f(3)}(3)$ | $\varphi_{f(3)}(4)$ | ... |
| $\vdots$         | $\vdots$            | $\vdots$            | $\vdots$            | $\vdots$            | $\vdots$            | ... |

## DIAGONALBEWEISE III:

# TOTAL BERECHENBARE FUNKTIONEN SIND NICHT AUFZÄHLBAR

**Annahme:**  $\mathcal{R}_\varphi = \{i \mid \varphi_i \text{ total}\}$  ist aufzählbar

- Dann gibt es eine berechenbare totale Funktion  $f$  mit  $\text{range}(f) = \mathcal{R}_\varphi$
- Dann lassen sich alle Funktionen aus  $\mathcal{R}$  wie folgt in eine Tabelle eintragen

|                  | 0                     | 1                   | 2                   | 3                   | 4                   | ... |
|------------------|-----------------------|---------------------|---------------------|---------------------|---------------------|-----|
| $\varphi_{f(0)}$ | $\varphi_{f(0)}(0)+1$ | $\varphi_{f(0)}(1)$ | $\varphi_{f(0)}(2)$ | $\varphi_{f(0)}(3)$ | $\varphi_{f(0)}(4)$ | ... |
| $\varphi_{f(1)}$ | $\varphi_{f(1)}(0)$   | $\varphi_{f(1)}(1)$ | $\varphi_{f(1)}(2)$ | $\varphi_{f(1)}(3)$ | $\varphi_{f(1)}(4)$ | ... |
| $\varphi_{f(2)}$ | $\varphi_{f(2)}(0)$   | $\varphi_{f(2)}(1)$ | $\varphi_{f(2)}(2)$ | $\varphi_{f(2)}(3)$ | $\varphi_{f(2)}(4)$ | ... |
| $\varphi_{f(3)}$ | $\varphi_{f(3)}(0)$   | $\varphi_{f(3)}(1)$ | $\varphi_{f(3)}(2)$ | $\varphi_{f(3)}(3)$ | $\varphi_{f(3)}(4)$ | ... |
| $\vdots$         | $\vdots$              | $\vdots$            | $\vdots$            | $\vdots$            | $\vdots$            | ... |

- Definiere eine neue Funktion  $h:\mathbb{N}\rightarrow\mathbb{N}$  durch  $h(n) = \varphi_{f(n)}(n)+1$

## DIAGONALBEWEISE III:

# TOTAL BERECHENBARE FUNKTIONEN SIND NICHT AUFZÄHLBAR

**Annahme:**  $\mathcal{R}_\varphi = \{i \mid \varphi_i \text{ total}\}$  ist aufzählbar

- Dann gibt es eine berechenbare totale Funktion  $f$  mit  $\text{range}(f) = \mathcal{R}_\varphi$
- Dann lassen sich alle Funktionen aus  $\mathcal{R}$  wie folgt in eine Tabelle eintragen

|                  | 0                     | 1                     | 2                   | 3                   | 4                   | ... |
|------------------|-----------------------|-----------------------|---------------------|---------------------|---------------------|-----|
| $\varphi_{f(0)}$ | $\varphi_{f(0)}(0)+1$ | $\varphi_{f(0)}(1)$   | $\varphi_{f(0)}(2)$ | $\varphi_{f(0)}(3)$ | $\varphi_{f(0)}(4)$ | ... |
| $\varphi_{f(1)}$ | $\varphi_{f(1)}(0)$   | $\varphi_{f(1)}(1)+1$ | $\varphi_{f(1)}(2)$ | $\varphi_{f(1)}(3)$ | $\varphi_{f(1)}(4)$ | ... |
| $\varphi_{f(2)}$ | $\varphi_{f(2)}(0)$   | $\varphi_{f(2)}(1)$   | $\varphi_{f(2)}(2)$ | $\varphi_{f(2)}(3)$ | $\varphi_{f(2)}(4)$ | ... |
| $\varphi_{f(3)}$ | $\varphi_{f(3)}(0)$   | $\varphi_{f(3)}(1)$   | $\varphi_{f(3)}(2)$ | $\varphi_{f(3)}(3)$ | $\varphi_{f(3)}(4)$ | ... |
| ⋮                | ⋮                     | ⋮                     | ⋮                   | ⋮                   | ⋮                   | ... |

- Definiere eine neue Funktion  $h: \mathbb{N} \rightarrow \mathbb{N}$  durch  $h(n) = \varphi_{f(n)}(n)+1$

## DIAGONALBEWEISE III:

# TOTAL BERECHENBARE FUNKTIONEN SIND NICHT AUFZÄHLBAR

**Annahme:**  $\mathcal{R}_\varphi = \{i \mid \varphi_i \text{ total}\}$  ist aufzählbar

- Dann gibt es eine berechenbare totale Funktion  $f$  mit  $\text{range}(f) = \mathcal{R}_\varphi$
- Dann lassen sich alle Funktionen aus  $\mathcal{R}$  wie folgt in eine Tabelle eintragen

|                  | 0                     | 1                     | 2                     | 3                   | 4                   | ... |
|------------------|-----------------------|-----------------------|-----------------------|---------------------|---------------------|-----|
| $\varphi_{f(0)}$ | $\varphi_{f(0)}(0)+1$ | $\varphi_{f(0)}(1)$   | $\varphi_{f(0)}(2)$   | $\varphi_{f(0)}(3)$ | $\varphi_{f(0)}(4)$ | ... |
| $\varphi_{f(1)}$ | $\varphi_{f(1)}(0)$   | $\varphi_{f(1)}(1)+1$ | $\varphi_{f(1)}(2)$   | $\varphi_{f(1)}(3)$ | $\varphi_{f(1)}(4)$ | ... |
| $\varphi_{f(2)}$ | $\varphi_{f(2)}(0)$   | $\varphi_{f(2)}(1)$   | $\varphi_{f(2)}(2)+1$ | $\varphi_{f(2)}(3)$ | $\varphi_{f(2)}(4)$ | ... |
| $\varphi_{f(3)}$ | $\varphi_{f(3)}(0)$   | $\varphi_{f(3)}(1)$   | $\varphi_{f(3)}(2)$   | $\varphi_{f(3)}(3)$ | $\varphi_{f(3)}(4)$ | ... |
| ⋮                | ⋮                     | ⋮                     | ⋮                     | ⋮                   | ⋮                   | ... |

- Definiere eine neue Funktion  $h: \mathbb{N} \rightarrow \mathbb{N}$  durch  $h(n) = \varphi_{f(n)}(n)+1$

## DIAGONALBEWEISE III:

# TOTAL BERECHENBARE FUNKTIONEN SIND NICHT AUFZÄHLBAR

**Annahme:**  $\mathcal{R}_\varphi = \{i \mid \varphi_i \text{ total}\}$  ist aufzählbar

- Dann gibt es eine berechenbare totale Funktion  $f$  mit  $\text{range}(f) = \mathcal{R}_\varphi$
- Dann lassen sich alle Funktionen aus  $\mathcal{R}$  wie folgt in eine Tabelle eintragen

|                  | 0                     | 1                     | 2                     | 3                     | 4                   | ... |
|------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|-----|
| $\varphi_{f(0)}$ | $\varphi_{f(0)}(0)+1$ | $\varphi_{f(0)}(1)$   | $\varphi_{f(0)}(2)$   | $\varphi_{f(0)}(3)$   | $\varphi_{f(0)}(4)$ | ... |
| $\varphi_{f(1)}$ | $\varphi_{f(1)}(0)$   | $\varphi_{f(1)}(1)+1$ | $\varphi_{f(1)}(2)$   | $\varphi_{f(1)}(3)$   | $\varphi_{f(1)}(4)$ | ... |
| $\varphi_{f(2)}$ | $\varphi_{f(2)}(0)$   | $\varphi_{f(2)}(1)$   | $\varphi_{f(2)}(2)+1$ | $\varphi_{f(2)}(3)$   | $\varphi_{f(2)}(4)$ | ... |
| $\varphi_{f(3)}$ | $\varphi_{f(3)}(0)$   | $\varphi_{f(3)}(1)$   | $\varphi_{f(3)}(2)$   | $\varphi_{f(3)}(3)+1$ | $\varphi_{f(3)}(4)$ | ... |
| ⋮                | ⋮                     | ⋮                     | ⋮                     | ⋮                     | ⋮                   | ... |

- Definiere eine neue Funktion  $h:\mathbb{N}\rightarrow\mathbb{N}$  durch  $h(n) = \varphi_{f(n)}(n)+1$

## DIAGONALBEWEISE III:

# TOTAL BERECHENBARE FUNKTIONEN SIND NICHT AUFZÄHLBAR

**Annahme:**  $\mathcal{R}_\varphi = \{i \mid \varphi_i \text{ total}\}$  ist aufzählbar

- Dann gibt es eine berechenbare totale Funktion  $f$  mit  $\text{range}(f) = \mathcal{R}_\varphi$
- Dann lassen sich alle Funktionen aus  $\mathcal{R}$  wie folgt in eine Tabelle eintragen

|                  | 0                     | 1                     | 2                     | 3                     | 4                   | ... |
|------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|-----|
| $\varphi_{f(0)}$ | $\varphi_{f(0)}(0)+1$ | $\varphi_{f(0)}(1)$   | $\varphi_{f(0)}(2)$   | $\varphi_{f(0)}(3)$   | $\varphi_{f(0)}(4)$ | ... |
| $\varphi_{f(1)}$ | $\varphi_{f(1)}(0)$   | $\varphi_{f(1)}(1)+1$ | $\varphi_{f(1)}(2)$   | $\varphi_{f(1)}(3)$   | $\varphi_{f(1)}(4)$ | ... |
| $\varphi_{f(2)}$ | $\varphi_{f(2)}(0)$   | $\varphi_{f(2)}(1)$   | $\varphi_{f(2)}(2)+1$ | $\varphi_{f(2)}(3)$   | $\varphi_{f(2)}(4)$ | ... |
| $\varphi_{f(3)}$ | $\varphi_{f(3)}(0)$   | $\varphi_{f(3)}(1)$   | $\varphi_{f(3)}(2)$   | $\varphi_{f(3)}(3)+1$ | $\varphi_{f(3)}(4)$ | ... |
| $\vdots$         | $\vdots$              | $\vdots$              | $\vdots$              | $\vdots$              | $\vdots$            | ... |

- Definiere eine neue Funktion  $h:\mathbb{N}\rightarrow\mathbb{N}$  durch  $h(n) = \varphi_{f(n)}(n)+1$
- $h$  ist offensichtlich total und berechenbar, denn  $h(n) = u(f(n), n)+1$

## DIAGONALBEWEISE III:

# TOTAL BERECHENBARE FUNKTIONEN SIND NICHT AUFZÄHLBAR

**Annahme:**  $\mathcal{R}_\varphi = \{i \mid \varphi_i \text{ total}\}$  ist aufzählbar

- Dann gibt es eine berechenbare totale Funktion  $f$  mit  $\text{range}(f) = \mathcal{R}_\varphi$
- Dann lassen sich alle Funktionen aus  $\mathcal{R}$  wie folgt in eine Tabelle eintragen

|                  | 0                     | 1                     | 2                     | 3                     | 4                   | ... |
|------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|-----|
| $\varphi_{f(0)}$ | $\varphi_{f(0)}(0)+1$ | $\varphi_{f(0)}(1)$   | $\varphi_{f(0)}(2)$   | $\varphi_{f(0)}(3)$   | $\varphi_{f(0)}(4)$ | ... |
| $\varphi_{f(1)}$ | $\varphi_{f(1)}(0)$   | $\varphi_{f(1)}(1)+1$ | $\varphi_{f(1)}(2)$   | $\varphi_{f(1)}(3)$   | $\varphi_{f(1)}(4)$ | ... |
| $\varphi_{f(2)}$ | $\varphi_{f(2)}(0)$   | $\varphi_{f(2)}(1)$   | $\varphi_{f(2)}(2)+1$ | $\varphi_{f(2)}(3)$   | $\varphi_{f(2)}(4)$ | ... |
| $\varphi_{f(3)}$ | $\varphi_{f(3)}(0)$   | $\varphi_{f(3)}(1)$   | $\varphi_{f(3)}(2)$   | $\varphi_{f(3)}(3)+1$ | $\varphi_{f(3)}(4)$ | ... |
| $\vdots$         | $\vdots$              | $\vdots$              | $\vdots$              | $\vdots$              | $\vdots$            | ... |

- Definiere eine neue Funktion  $h: \mathbb{N} \rightarrow \mathbb{N}$  durch  $h(n) = \varphi_{f(n)}(n)+1$
- $h$  ist offensichtlich total und berechenbar, denn  $h(n) = u(f(n), n)+1$
- Also gibt es ein  $i \in \mathcal{R}_\varphi$  mit  $h = \varphi_i$  und damit ein  $j \in \mathbb{N}$  mit  $i=f(j)$

## DIAGONALBEWEISE III:

# TOTAL BERECHENBARE FUNKTIONEN SIND NICHT AUFZÄHLBAR

**Annahme:**  $\mathcal{R}_\varphi = \{i \mid \varphi_i \text{ total}\}$  ist aufzählbar

- Dann gibt es eine berechenbare totale Funktion  $f$  mit  $\text{range}(f) = \mathcal{R}_\varphi$
- Dann lassen sich alle Funktionen aus  $\mathcal{R}$  wie folgt in eine Tabelle eintragen

|                  | 0                     | 1                     | 2                     | 3                     | 4                   | ... |
|------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|-----|
| $\varphi_{f(0)}$ | $\varphi_{f(0)}(0)+1$ | $\varphi_{f(0)}(1)$   | $\varphi_{f(0)}(2)$   | $\varphi_{f(0)}(3)$   | $\varphi_{f(0)}(4)$ | ... |
| $\varphi_{f(1)}$ | $\varphi_{f(1)}(0)$   | $\varphi_{f(1)}(1)+1$ | $\varphi_{f(1)}(2)$   | $\varphi_{f(1)}(3)$   | $\varphi_{f(1)}(4)$ | ... |
| $\varphi_{f(2)}$ | $\varphi_{f(2)}(0)$   | $\varphi_{f(2)}(1)$   | $\varphi_{f(2)}(2)+1$ | $\varphi_{f(2)}(3)$   | $\varphi_{f(2)}(4)$ | ... |
| $\varphi_{f(3)}$ | $\varphi_{f(3)}(0)$   | $\varphi_{f(3)}(1)$   | $\varphi_{f(3)}(2)$   | $\varphi_{f(3)}(3)+1$ | $\varphi_{f(3)}(4)$ | ... |
| $\vdots$         | $\vdots$              | $\vdots$              | $\vdots$              | $\vdots$              | $\vdots$            | ... |

- Definiere eine neue Funktion  $h: \mathbb{N} \rightarrow \mathbb{N}$  durch  $h(n) = \varphi_{f(n)}(n)+1$
- $h$  ist offensichtlich total und berechenbar, denn  $h(n) = u(f(n), n)+1$
- Also gibt es ein  $i \in \mathcal{R}_\varphi$  mit  $h = \varphi_i$  und damit ein  $j \in \mathbb{N}$  mit  $i=f(j)$
- Für dieses  $j$  gilt  $\varphi_{f(j)}(j) = h(j) = \varphi_{f(j)}(j)+1$



# ANDERE UNLÖSBARE PROBLEME MIT DIAGONALBEWEISEN

- **Selbstanwendbarkeitsproblem:**

- $S = \{i \mid \varphi_i(i) \neq \perp\}$

unentscheidbar, aber aufzählbar

# ANDERE UNLÖSBARE PROBLEME MIT DIAGONALBEWEISEN

- **Selbstanwendbarkeitsproblem:**

- $S = \{i \mid \varphi_i(i) \neq \perp\}$  unentscheidbar, aber aufzählbar

- **Entscheidungsproblem:**

- $E = \{(i, j) \mid \varphi_i(j) = 1\}$  unentscheidbar, aber aufzählbar

# ANDERE UNLÖSBARE PROBLEME MIT DIAGONALBEWEISEN

- **Selbstanwendbarkeitsproblem:**
  - $S = \{i \mid \varphi_i(i) \neq \perp\}$  unentscheidbar, aber aufzählbar
- **Entscheidungsproblem:**
  - $E = \{(i, j) \mid \varphi_i(j) = 1\}$  unentscheidbar, aber aufzählbar
- **Monotone Funktionen:**
  - $MON = \{i \mid \forall k. \varphi_i(k) < \varphi_i(k+1)\}$  nicht aufzählbar

# ANDERE UNLÖSBARE PROBLEME MIT DIAGONALBEWEISEN

- **Selbstanwendbarkeitsproblem:**
  - $S = \{i \mid \varphi_i(i) \neq \perp\}$  unentscheidbar, aber aufzählbar
- **Entscheidungsproblem:**
  - $E = \{(i, j) \mid \varphi_i(j) = 1\}$  unentscheidbar, aber aufzählbar
- **Monotone Funktionen:**
  - $MON = \{i \mid \forall k. \varphi_i(k) < \varphi_i(k+1)\}$  nicht aufzählbar
- **Entscheidungsfunktionen:**
  - $EF = \{i \mid \forall j. \varphi_i(j) \in \{0, 1\}\}$  nicht aufzählbar

# MONOTONIEARGUMENTE

- **Ziel**

- Zeige, daß eine Funktion  $f$  eine Eigenschaft  $P$  nicht besitzt
- z.B. primitiv rekursiv, berechenbar, maximale Komplexität (Rechenzeit)

# MONOTONIEARGUMENTE

- **Ziel**

- Zeige, daß eine Funktion  $f$  eine Eigenschaft  $P$  nicht besitzt
- z.B. primitiv rekursiv, berechenbar, maximale Komplexität (Rechenzeit)

- **Methodik**

- Zeige daß  $f$  stärker wächst als jede Funktion mit Eigenschaft  $P$

# MONOTONIEARGUMENTE

## ● Ziel

- Zeige, daß eine Funktion  $f$  eine Eigenschaft  $P$  nicht besitzt
- z.B. primitiv rekursiv, berechenbar, maximale Komplexität (Rechenzeit)

## ● Methodik

- Zeige daß  $f$  stärker wächst als jede Funktion mit Eigenschaft  $P$ 
  - Induktive Analyse des Wachstumsverhaltens von  $f$
  - Analyse des maximalen Wachstums von Funktionen mit Eigenschaft  $P$

# MONOTONIEARGUMENTE

## ● Ziel

- Zeige, daß eine Funktion  $f$  eine Eigenschaft  $P$  nicht besitzt
- z.B. primitiv rekursiv, berechenbar, maximale Komplexität (Rechenzeit)

## ● Methodik

- Zeige daß  $f$  stärker wächst als jede Funktion mit Eigenschaft  $P$ 
  - Induktive Analyse des Wachstumsverhaltens von  $f$
  - Analyse des maximalen Wachstums von Funktionen mit Eigenschaft  $P$
- $f$  kann also nicht selbst Eigenschaft  $P$  besitzen

# MONOTONIEARGUMENTE

## ● Ziel

- Zeige, daß eine Funktion  $f$  eine Eigenschaft  $P$  nicht besitzt
- z.B. primitiv rekursiv, berechenbar, maximale Komplexität (Rechenzeit)

## ● Methodik

- Zeige daß  $f$  stärker wächst als jede Funktion mit Eigenschaft  $P$ 
  - Induktive Analyse des Wachstumsverhaltens von  $f$
  - Analyse des maximalen Wachstums von Funktionen mit Eigenschaft  $P$
- $f$  kann also nicht selbst Eigenschaft  $P$  besitzen

## ● Beispiele

- Die Ackermann Funktion ist nicht primitiv-rekursiv
- Die Busy-Beaver Funktion ist nicht berechenbar
- Komplexitätsanalysen

folgt

# DAS BUSY-BEAVER PROBLEM

Biber stauen Bäche, indem sie Holzstücke in den Bach tragen. Fleißige Biber tragen mehr Holzstücke zusammen als faule. Größere Biber können mehr leisten als kleine. Die Busy-Beaver Funktion liefert die Länge der längsten ununterbrochenen Staumauer, die ein Biber zusammentragen kann, ohne daß schon eine Teilmauer vorhanden war.

# DAS BUSY-BEAVER PROBLEM

Biber stauen Bäche, indem sie Holzstücke in den Bach tragen. Fleißige Biber tragen mehr Holzstücke zusammen als faule. Größere Biber können mehr leisten als kleine. Die Busy-Beaver Funktion liefert die Länge der längsten ununterbrochenen Staumauer, die ein Biber zusammentragen kann, ohne daß schon eine Teilmauer vorhanden war.

## ● Beschreibe Biber durch Turingmaschinen

- Holzstücke werden durch das Symbol  $|$  beschrieben
- $\tau = (\{1..n\}, \{|\}, \{|\}, \delta, 1, b)$  heißt **Busy-Beaver TM der Größe  $n$**
- **BBT( $n$ )** sei die Menge aller Busy-Beaver Turingmaschinen der Größe  $n$

# DAS BUSY-BEAVER PROBLEM

Biber stauen Bäche, indem sie Holzstücke in den Bach tragen. Fleißige Biber tragen mehr Holzstücke zusammen als faule. Größere Biber können mehr leisten als kleine. Die Busy-Beaver Funktion liefert die Länge der längsten ununterbrochenen Staumauer, die ein Biber zusammentragen kann, ohne daß schon eine Teilmauer vorhanden war.

## ● Beschreibe Biber durch Turingmaschinen

- Holzstücke werden durch das Symbol  $|$  beschrieben
- $\tau = (\{1..n\}, \{|\}, \{|\}, \delta, 1, b)$  heißt **Busy-Beaver TM der Größe  $n$**
- **BBT( $n$ )** sei die Menge aller Busy-Beaver Turingmaschinen der Größe  $n$

## ● Beschreibe Produktivität von Bibern

- **Produktivität( $\tau$ )** = 
$$\begin{cases} n & \text{wenn } h_\tau(\epsilon) = |^n \\ 0 & \text{wenn } \tau \text{ bei Eingabe } \epsilon \text{ nicht hält} \end{cases}$$

# DAS BUSY-BEAVER PROBLEM

Biber stauen Bäche, indem sie Holzstücke in den Bach tragen. Fleißige Biber tragen mehr Holzstücke zusammen als faule. Größere Biber können mehr leisten als kleine. Die Busy-Beaver Funktion liefert die Länge der längsten ununterbrochenen Staumauer, die ein Biber zusammentragen kann, ohne daß schon eine Teilmauer vorhanden war.

## ● Beschreibe Biber durch Turingmaschinen

- Holzstücke werden durch das Symbol  $|$  beschrieben
- $\tau = (\{1..n\}, \{|\}, \{|\}, \delta, 1, b)$  heißt **Busy-Beaver TM der Größe  $n$**
- **BBT( $n$ )** sei die Menge aller Busy-Beaver Turingmaschinen der Größe  $n$

## ● Beschreibe Produktivität von Bibern

- **Produktivität( $\tau$ )** = 
$$\begin{cases} n & \text{wenn } h_\tau(\epsilon) = |^n \\ 0 & \text{wenn } \tau \text{ bei Eingabe } \epsilon \text{ nicht hält} \end{cases}$$

## ● Beschreibe maximal mögliche Leistung von Bibern

- **BB( $n$ )** =  $\max \{\text{Produktivität}(\tau) \mid \tau \in \text{BBT}(n)\}$

# DAS BUSY-BEAVER PROBLEM

Biber stauen Bäche, indem sie Holzstücke in den Bach tragen. Fleißige Biber tragen mehr Holzstücke zusammen als faule. Größere Biber können mehr leisten als kleine. Die Busy-Beaver Funktion liefert die Länge der längsten ununterbrochenen Staumauer, die ein Biber zusammentragen kann, ohne daß schon eine Teilmauer vorhanden war.

## ● Beschreibe Biber durch Turingmaschinen

- Holzstücke werden durch das Symbol  $|$  beschrieben
- $\tau = (\{1..n\}, \{|\}, \{|\}, \delta, 1, b)$  heißt **Busy-Beaver TM der Größe  $n$**
- **BBT( $n$ )** sei die Menge aller Busy-Beaver Turingmaschinen der Größe  $n$

## ● Beschreibe Produktivität von Bibern

- **Produktivität( $\tau$ )** = 
$$\begin{cases} n & \text{wenn } h_\tau(\epsilon) = |^n \\ 0 & \text{wenn } \tau \text{ bei Eingabe } \epsilon \text{ nicht hält} \end{cases}$$

## ● Beschreibe maximal mögliche Leistung von Bibern

- **BB( $n$ )** =  $\max \{\text{Produktivität}(\tau) \mid \tau \in \text{BBT}(n)\}$

**Ist die Busy-Beaver Funktion berechenbar?**

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta =$$

| $s$ | $a$ | $s'$ | $a'$ | $P$ |
|-----|-----|------|------|-----|
| 1   |     | 2    |      | r   |
| 1   | b   | 2    |      | l   |
| 2   |     | 2    |      | h   |
| 2   | b   | 1    |      | l   |

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta =$$

| $s$ | $a$ | $s'$ | $a'$ | $P$ |
|-----|-----|------|------|-----|
| 1   |     | 2    |      | r   |
| 1   | b   | 2    |      | l   |
| 2   |     | 2    |      | h   |
| 2   | b   | 1    |      | l   |

Arbeitsweise:

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta =$$

| $s$ | $a$ | $s'$ | $a'$ | $P$ |
|-----|-----|------|------|-----|
| 1   |     | 2    |      | r   |
| 1   | b   | 2    |      | l   |
| 2   |     | 2    |      | h   |
| 2   | b   | 1    |      | l   |

Arbeitsweise:  $b1b$

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta =$$

| $s$ | $a$ | $s'$ | $a'$ | $P$ |
|-----|-----|------|------|-----|
| 1   |     | 2    |      | r   |
| 1   | b   | 2    |      | l   |
| 2   |     | 2    |      | h   |
| 2   | b   | 1    |      | l   |

Arbeitsweise:  $b1b$

$\mapsto |2b$

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta =$$

| $s$ | $a$ | $s'$ | $a'$ | $P$ |
|-----|-----|------|------|-----|
| 1   |     | 2    |      | r   |
| 1   | b   | 2    |      | l   |
| 2   |     | 2    |      | h   |
| 2   | b   | 1    |      | l   |

Arbeitsweise:  $b1b$   
 $\mapsto |2b$   
 $\mapsto b1||$

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta = \begin{array}{c|cc|cc} s & a & s' & a' & P \\ \hline 1 & | & 2 & | & r \\ 1 & b & 2 & | & l \\ 2 & | & 2 & | & h \\ 2 & b & 1 & | & l \\ \hline \end{array}$$

Arbeitsweise:

$b1b$   
 $\mapsto |2b$   
 $\mapsto b1||$   
 $\mapsto b2b||$

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta = \begin{array}{c|cc|cc} s & a & s' & a' & P \\ \hline 1 & | & 2 & | & r \\ 1 & b & 2 & | & l \\ 2 & | & 2 & | & h \\ 2 & b & 1 & | & l \\ \hline \end{array}$$

Arbeitsweise:

$b1b$   
 $\mapsto |2b$   
 $\mapsto b1||$   
 $\mapsto b2b||$   
 $\mapsto b1b|||$

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta = \begin{array}{c|cc|cc} s & a & s' & a' & P \\ \hline 1 & | & 2 & | & r \\ 1 & b & 2 & | & l \\ 2 & | & 2 & | & h \\ 2 & b & 1 & | & l \\ \hline \end{array}$$

Arbeitsweise:

$b1b$   
 $\mapsto |2b$   
 $\mapsto b1||$   
 $\mapsto b2b||$   
 $\mapsto b1b|||$   
 $\mapsto |2|||$

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta = \begin{array}{c|cc|cc} s & a & s' & a' & P \\ \hline 1 & | & 2 & | & r \\ 1 & b & 2 & | & l \\ 2 & | & 2 & | & h \\ 2 & b & 1 & | & l \\ \hline \end{array}$$

Arbeitsweise:

$b1b$   
 $\mapsto |2b$   
 $\mapsto b1||$   
 $\mapsto b2b||$   
 $\mapsto b1b|||$   
 $\mapsto |2|||$   
stop

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta = \begin{array}{c|cc|cc} s & a & s' & a' & P \\ \hline 1 & | & 2 & | & r \\ 1 & b & 2 & | & l \\ 2 & | & 2 & | & h \\ 2 & b & 1 & | & l \\ \hline \end{array}$$

Arbeitsweise:  $b1b$   
 $\mapsto |2b$   
 $\mapsto b1||$   
 $\mapsto b2b||$   
 $\mapsto b1b|||$   
 $\mapsto |2|||$   
stop

– Produktivität ist 3 (4, wenn man alle Holzstücke zählt)

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta = \begin{array}{c|c|c|c|c} s & a & s' & a' & P \\ \hline 1 & | & 2 & | & r \\ 1 & b & 2 & | & l \\ 2 & | & 2 & | & h \\ 2 & b & 1 & | & l \\ \hline \end{array}$$

Arbeitsweise:  $b1b$   
 $\mapsto |2b$   
 $\mapsto b1||$   
 $\mapsto b2b||$   
 $\mapsto b1b|||$   
 $\mapsto |2|||$   
 stop

– Produktivität ist 3 (4, wenn man alle Holzstücke zählt)

## ● $BB(n)$ bekannt für kleine $n$ :

| $n$ | 1 | 2 | 3 | 4  | 5           | 6                         | ... |
|-----|---|---|---|----|-------------|---------------------------|-----|
|     | 1 | 4 | 6 | 13 | $\geq 4098$ | $\geq 6.4 \cdot 10^{462}$ |     |

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta = \begin{array}{c|cc|cc|c} s & a & s' & a' & P \\ \hline 1 & | & 2 & | & r \\ 1 & b & 2 & | & l \\ 2 & | & 2 & | & h \\ 2 & b & 1 & | & l \end{array}$$

Arbeitsweise:  $b1b$   
 $\mapsto |2b$   
 $\mapsto b1||$   
 $\mapsto b2b||$   
 $\mapsto b1b|||$   
 $\mapsto |2|||$   
 stop

– Produktivität ist 3 (4, wenn man alle Holzstücke zählt)

## ● $BB(n)$ bekannt für kleine $n$ :

|     |   |   |   |    |             |                           |     |
|-----|---|---|---|----|-------------|---------------------------|-----|
| $n$ | 1 | 2 | 3 | 4  | 5           | 6                         | ... |
|     | 1 | 4 | 6 | 13 | $\geq 4098$ | $\geq 6.4 \cdot 10^{462}$ |     |

## ● Vollständige Analyse nicht möglich

–  $|BBT(n)|$  ist  $(|S| * |\Gamma| * |\{r, l\}|)^{(|S| * |\Gamma| - 1)} * (|S| * |\Gamma| * |\{h\}|) * (|S| * |\Gamma|) = (4n)^{2n}$   
 $|BBT(1)|=16, |BBT(2)|=4096, |BBT(3)|=2985984, \dots$

# BUSY-BEAVER PROBLEM: INTUITIVE ANALYSE

## ● Beispiel einer BBT(2) Maschine

$$\delta = \begin{array}{c|cc|cc} s & a & s' & a' & P \\ \hline 1 & | & 2 & | & r \\ 1 & b & 2 & | & l \\ 2 & | & 2 & | & h \\ 2 & b & 1 & | & l \\ \hline \end{array}$$

Arbeitsweise:  $b1b$   
 $\mapsto |2b$   
 $\mapsto b1||$   
 $\mapsto b2b||$   
 $\mapsto b1b|||$   
 $\mapsto |2|||$   
 stop

– Produktivität ist 3 (4, wenn man alle Holzstücke zählt)

## ● $BB(n)$ bekannt für kleine $n$ :

|     |   |   |   |    |             |                           |     |
|-----|---|---|---|----|-------------|---------------------------|-----|
| $n$ | 1 | 2 | 3 | 4  | 5           | 6                         | ... |
|     | 1 | 4 | 6 | 13 | $\geq 4098$ | $\geq 6.4 \cdot 10^{462}$ |     |

## ● Vollständige Analyse nicht möglich

- $|BBT(n)|$  ist  $(|S| * |\Gamma| * |\{r, l\}|)^{(|S| * |\Gamma| - 1)} * (|S| * |\Gamma| * |\{h\}|) * (|S| * |\Gamma|) = (4n)^{2n}$   
 $|BBT(1)|=16$ ,  $|BBT(2)|=4096$ ,  $|BBT(3)|=2985984$ , ...
- Anzahl möglicher Bandkonfigurationen einer TM ist unbegrenzt

# DAS BUSY-BEAVER PROBLEM IST UNLÖSBAR

- **BB ist streng monoton:**  $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$

# DAS BUSY-BEAVER PROBLEM IST UNLÖSBAR

- **BB ist streng monoton:**  $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$ 
  - Für alle  $n$  gilt  $\text{BB}(n+1) > \text{BB}(n)$ 
    - Schreibe in Zustand 1 ein  $|$  und beginne mit der  $\text{BB}(n)$ -Maschine

# DAS BUSY-BEAVER PROBLEM IST UNLÖSBAR

- **BB ist streng monoton:**  $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$ 
  - Für alle  $n$  gilt  $\text{BB}(n+1) > \text{BB}(n)$ 
    - Schreibe in Zustand 1 ein  $|$  und beginne mit der  $\text{BB}(n)$ -Maschine
  - $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$  folgt nun durch **Induktion** über  $i - j$

# DAS BUSY-BEAVER PROBLEM IST UNLÖSBAR

- **BB ist streng monoton:**  $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$ 
  - Für alle  $n$  gilt  $\text{BB}(n+1) > \text{BB}(n)$ 
    - Schreibe in Zustand 1 ein | und beginne mit der  $\text{BB}(n)$ -Maschine
  - $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$  folgt nun durch Induktion über  $i - j$
- **Für alle  $n$  gilt:**  $\text{BB}(n+8) \geq 2n$

# DAS BUSY-BEAVER PROBLEM IST UNLÖSBAR

- **BB ist streng monoton:**  $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$ 
  - Für alle  $n$  gilt  $\text{BB}(n+1) > \text{BB}(n)$ 
    - Schreibe in Zustand 1 ein | und beginne mit der  $\text{BB}(n)$ -Maschine
  - $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$  folgt nun durch **Induktion** über  $i - j$
- **Für alle  $n$  gilt:**  $\text{BB}(n+8) \geq 2n$ 
  - Mit  $n$  Zuständen kann man  $n$  Striche generieren

# DAS BUSY-BEAVER PROBLEM IST UNLÖSBAR

- **BB ist streng monoton:**  $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$ 
  - Für alle  $n$  gilt  $\text{BB}(n+1) > \text{BB}(n)$ 
    - Schreibe in Zustand 1 ein | und beginne mit der  $\text{BB}(n)$ -Maschine
  - $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$  folgt nun durch Induktion über  $i - j$
- **Für alle  $n$  gilt:**  $\text{BB}(n+8) \geq 2n$ 
  - Mit  $n$  Zuständen kann man  $n$  Striche generieren
  - Mit 8 Zuständen kann man Striche verdoppeln (vgl  $\tau_4$  aus Kapitel 6.1)

# DAS BUSY-BEAVER PROBLEM IST UNLÖSBAR

- **BB ist streng monoton:  $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$** 
  - Für alle  $n$  gilt  $\text{BB}(n+1) > \text{BB}(n)$ 
    - Schreibe in Zustand 1 ein | und beginne mit der  $\text{BB}(n)$ -Maschine
  - $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$  folgt nun durch Induktion über  $i - j$
- **Für alle  $n$  gilt:  $\text{BB}(n+8) \geq 2n$** 
  - Mit  $n$  Zuständen kann man  $n$  Striche generieren
  - Mit 8 Zuständen kann man Striche verdoppeln (vgl  $\tau_4$  aus Kapitel 6.1)
- **BB berechenbar  $\Rightarrow \text{BB}(n+2k) \geq \text{BB}(\text{BB}(n))$  für ein  $k$**

# DAS BUSY-BEAVER PROBLEM IST UNLÖSBAR

- **BB ist streng monoton:  $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$** 
  - Für alle  $n$  gilt  $\text{BB}(n+1) > \text{BB}(n)$ 
    - Schreibe in Zustand 1 ein  $|$  und beginne mit der  $\text{BB}(n)$ -Maschine
  - $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$  folgt nun durch Induktion über  $i - j$
- **Für alle  $n$  gilt:  $\text{BB}(n+8) \geq 2n$** 
  - Mit  $n$  Zuständen kann man  $n$  Striche generieren
  - Mit 8 Zuständen kann man Striche verdoppeln (vgl  $\tau_4$  aus Kapitel 6.1)
- **BB berechenbar  $\Rightarrow \text{BB}(n+2k) \geq \text{BB}(\text{BB}(n))$  für ein  $k$** 
  - Wähle  $k :=$  Anzahl der Zustände der TM über  $\Gamma = \{ |, b \}$ , die BB berechnet

# DAS BUSY-BEAVER PROBLEM IST UNLÖSBAR

- **BB ist streng monoton:**  $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$ 
  - Für alle  $n$  gilt  $\text{BB}(n+1) > \text{BB}(n)$ 
    - Schreibe in Zustand 1 ein  $|$  und beginne mit der  $\text{BB}(n)$ -Maschine
  - $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$  folgt nun durch Induktion über  $i - j$
- **Für alle  $n$  gilt:**  $\text{BB}(n+8) \geq 2n$ 
  - Mit  $n$  Zuständen kann man  $n$  Striche generieren
  - Mit 8 Zuständen kann man Striche verdoppeln (vgl  $\tau_4$  aus Kapitel 6.1)
- **BB berechenbar  $\Rightarrow \text{BB}(n+2k) \geq \text{BB}(\text{BB}(n))$  für ein  $k$** 
  - Wähle  $k :=$  Anzahl der Zustände der TM über  $\Gamma = \{ |, b \}$ , die BB berechnet
  - Mit  $n$  Zuständen generiere  $n$  Striche
  - Mit  $k$  Zuständen berechne jetzt  $\text{BB}(n)$
  - Mit weiteren  $k$  Zuständen berechne  $\text{BB}(\text{BB}(n))$

# DAS BUSY-BEAVER PROBLEM IST UNLÖSBAR

- **BB ist streng monoton:**  $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$ 
  - Für alle  $n$  gilt  $\text{BB}(n+1) > \text{BB}(n)$ 
    - Schreibe in Zustand 1 ein  $|$  und beginne mit der  $\text{BB}(n)$ -Maschine
  - $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$  folgt nun durch Induktion über  $i - j$
- **Für alle  $n$  gilt:**  $\text{BB}(n+8) \geq 2n$ 
  - Mit  $n$  Zuständen kann man  $n$  Striche generieren
  - Mit 8 Zuständen kann man Striche verdoppeln (vgl  $\tau_4$  aus Kapitel 6.1)
- **BB berechenbar  $\Rightarrow \text{BB}(n+2k) \geq \text{BB}(\text{BB}(n))$  für ein  $k$** 
  - Wähle  $k :=$  Anzahl der Zustände der TM über  $\Gamma = \{ |, b \}$ , die BB berechnet
  - Mit  $n$  Zuständen generiere  $n$  Striche
  - Mit  $k$  Zuständen berechne jetzt  $\text{BB}(n)$
  - Mit weiteren  $k$  Zuständen berechne  $\text{BB}(\text{BB}(n))$
- **BB kann nicht berechenbar sein**

# DAS BUSY-BEAVER PROBLEM IST UNLÖSBAR

- **BB ist streng monoton:**  $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$ 
  - Für alle  $n$  gilt  $\text{BB}(n+1) > \text{BB}(n)$ 
    - Schreibe in Zustand 1 ein  $|$  und beginne mit der  $\text{BB}(n)$ -Maschine
  - $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$  folgt nun durch Induktion über  $i - j$
- **Für alle  $n$  gilt:**  $\text{BB}(n+8) \geq 2n$ 
  - Mit  $n$  Zuständen kann man  $n$  Striche generieren
  - Mit 8 Zuständen kann man Striche verdoppeln (vgl  $\tau_4$  aus Kapitel 6.1)
- **BB berechenbar  $\Rightarrow \text{BB}(n+2k) \geq \text{BB}(\text{BB}(n))$  für ein  $k$** 
  - Wähle  $k :=$  Anzahl der Zustände der TM über  $\Gamma = \{ |, b \}$ , die BB berechnet
  - Mit  $n$  Zuständen generiere  $n$  Striche
  - Mit  $k$  Zuständen berechne jetzt  $\text{BB}(n)$
  - Mit weiteren  $k$  Zuständen berechne  $\text{BB}(\text{BB}(n))$
- **BB kann nicht berechenbar sein**
  - Sonst gibt es ein  $k$ , so daß für alle  $n$ :  $\text{BB}(n+8+2k) \geq \text{BB}(\text{BB}(n+8)) \geq \text{BB}(2n)$

# DAS BUSY-BEAVER PROBLEM IST UNLÖSBAR

- **BB ist streng monoton:**  $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$ 
  - Für alle  $n$  gilt  $\text{BB}(n+1) > \text{BB}(n)$ 
    - Schreibe in Zustand 1 ein  $|$  und beginne mit der  $\text{BB}(n)$ -Maschine
  - $i > j \Leftrightarrow \text{BB}(i) > \text{BB}(j)$  folgt nun durch Induktion über  $i - j$
- **Für alle  $n$  gilt:**  $\text{BB}(n+8) \geq 2n$ 
  - Mit  $n$  Zuständen kann man  $n$  Striche generieren
  - Mit 8 Zuständen kann man Striche verdoppeln (vgl  $\tau_4$  aus Kapitel 6.1)
- **BB berechenbar  $\Rightarrow \text{BB}(n+2k) \geq \text{BB}(\text{BB}(n))$  für ein  $k$** 
  - Wähle  $k :=$  Anzahl der Zustände der TM über  $\Gamma = \{ |, b \}$ , die BB berechnet
  - Mit  $n$  Zuständen generiere  $n$  Striche
  - Mit  $k$  Zuständen berechne jetzt  $\text{BB}(n)$
  - Mit weiteren  $k$  Zuständen berechne  $\text{BB}(\text{BB}(n))$
- **BB kann nicht berechenbar sein**
  - Sonst gibt es ein  $k$ , so daß für alle  $n$ :  $\text{BB}(n+8+2k) \geq \text{BB}(\text{BB}(n+8)) \geq \text{BB}(2n)$
  - Für  $n=2k+9$  widersprüche  $\text{BB}(4k+17) \geq \text{BB}(4k+18)$  der Monotonie ✓

# PROBLEMREDUKTION

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit

# PROBLEMREDUKTION

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit
  - Unlösbar  $\hat{=}$  unentscheidbar, nicht aufzählbar, nicht in Zeit  $t$  lösbar, ...

# PROBLEMREDUKTION

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit
  - Unlösbar  $\hat{=}$  unentscheidbar, nicht aufzählbar, nicht in Zeit  $t$  lösbar, ...
- **Methodik zum Nachweis der Unlösbarkeit**
  - Transformiere  $P$  in ein anderes Problem  $P'$ , das als unlösbar bekannt ist

# PROBLEMREDUKTION

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit
  - Unlösbar  $\hat{=}$  unentscheidbar, nicht aufzählbar, nicht in Zeit  $t$  lösbar, ...
- **Methodik zum Nachweis der Unlösbarkeit**
  - Transformiere  $P$  in ein anderes Problem  $P'$ , das als unlösbar bekannt ist
  - Zeige, daß jede Lösung für  $P$  in eine Lösung für  $P'$  transformiert würde

# PROBLEMREDUKTION

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit
  - Unlösbar  $\hat{=}$  unentscheidbar, nicht aufzählbar, nicht in Zeit  $t$  lösbar, ...
- **Methodik zum Nachweis der Unlösbarkeit**
  - Transformiere  $P$  in ein anderes Problem  $P'$ , das als unlösbar bekannt ist
  - Zeige, daß jede Lösung für  $P$  in eine Lösung für  $P'$  transformiert würde
  - $P$  kann also nicht lösbar sein

# PROBLEMREDUKTION

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit
  - Unlösbar  $\hat{=}$  unentscheidbar, nicht aufzählbar, nicht in Zeit  $t$  lösbar, ...
- **Methodik zum Nachweis der Unlösbarkeit**
  - Transformiere  $P$  in ein anderes Problem  $P'$ , das als unlösbar bekannt ist
  - Zeige, daß jede Lösung für  $P$  in eine Lösung für  $P'$  transformiert würde
  - $P$  kann also nicht lösbar sein
- **Methodik zur Konstruktion einer Lösung**
  - Transformiere ein anderes Problem  $P'$ , das als lösbar bekannt ist, in  $P$

# PROBLEMREDUKTION

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit
  - Unlösbar  $\hat{=}$  unentscheidbar, nicht aufzählbar, nicht in Zeit  $t$  lösbar, ...
- **Methodik zum Nachweis der Unlösbarkeit**
  - Transformiere  $P$  in ein anderes Problem  $P'$ , das als unlösbar bekannt ist
  - Zeige, daß jede Lösung für  $P$  in eine Lösung für  $P'$  transformiert würde
  - $P$  kann also nicht lösbar sein
- **Methodik zur Konstruktion einer Lösung**
  - Transformiere ein anderes Problem  $P'$ , das als lösbar bekannt ist, in  $P$
  - Zeige, wie eine Lösung für  $P'$  in eine Lösung für  $P$  transformiert wird

# PROBLEMREDUKTION

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit
  - Unlösbar  $\hat{=}$  unentscheidbar, nicht aufzählbar, nicht in Zeit  $t$  lösbar, ...
- **Methodik zum Nachweis der Unlösbarkeit**
  - Transformiere  $P$  in ein anderes Problem  $P'$ , das als unlösbar bekannt ist
  - Zeige, daß jede Lösung für  $P$  in eine Lösung für  $P'$  transformiert würde
  - $P$  kann also nicht lösbar sein
- **Methodik zur Konstruktion einer Lösung**
  - Transformiere ein anderes Problem  $P'$ , das als lösbar bekannt ist, in  $P$
  - Zeige, wie eine Lösung für  $P'$  in eine Lösung für  $P$  transformiert wird
- **Hilfsmittel: Reduzierbarkeit  $P' \leq P$** 
  - $P' \leq P$ , falls  $P' = f^{-1}(P) = \{x \mid f(x) \in P\}$  für ein total-berechenbares  $f$
  - “ $P'$  ist reduzierbar auf  $P$ ”

# PROBLEMREDUKTION

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit
  - Unlösbar  $\hat{=}$  unentscheidbar, nicht aufzählbar, nicht in Zeit  $t$  lösbar, ...
- **Methodik zum Nachweis der Unlösbarkeit**
  - Transformiere  $P$  in ein anderes Problem  $P'$ , das als unlösbar bekannt ist
  - Zeige, daß jede Lösung für  $P$  in eine Lösung für  $P'$  transformiert würde
  - $P$  kann also nicht lösbar sein
- **Methodik zur Konstruktion einer Lösung**
  - Transformiere ein anderes Problem  $P'$ , das als lösbar bekannt ist, in  $P$
  - Zeige, wie eine Lösung für  $P'$  in eine Lösung für  $P$  transformiert wird
- **Hilfsmittel: Reduzierbarkeit  $P' \leq P$** 
  - $P' \leq P$ , falls  $P' = f^{-1}(P) = \{x \mid f(x) \in P\}$  für ein total-berechenbares  $f$
  - “ $P'$  ist reduzierbar auf  $P$ ” (Begriff gilt für Teilmengen von Zahlen, Worten, ...)

# BEWEISFÜHRUNG DURCH REDUKTION

- $P' \leq P$  bedeutet “ $P'$  ist leichter als  $P$ ”

# BEWEISFÜHRUNG DURCH REDUKTION

- $P' \leq P$  bedeutet “ $P'$  ist leichter als  $P$ ”
  - Ist  $P$  lösbar, dann kann  $P'$  wie folgt gelöst werden

# BEWEISFÜHRUNG DURCH REDUKTION

- $P' \leq P$  bedeutet “ $P'$  ist leichter als  $P$ ”
  - Ist  $P$  lösbar, dann kann  $P'$  wie folgt gelöst werden
    - Bei Eingabe  $x$  bestimme  $f(x)$  ( $f$  ist die Reduktionsfunktion)

# BEWEISFÜHRUNG DURCH REDUKTION

- $P' \leq P$  bedeutet “ $P'$  ist leichter als  $P$ ”
  - Ist  $P$  lösbar, dann kann  $P'$  wie folgt gelöst werden
    - Bei Eingabe  $x$  bestimme  $f(x)$  ( $f$  ist die Reduktionsfunktion)
    - Löse  $f(x)$  mit der Lösungsmethode für  $P$

# BEWEISFÜHRUNG DURCH REDUKTION

- $P' \leq P$  bedeutet “ $P'$  ist leichter als  $P$ ”
  - Ist  $P$  lösbar, dann kann  $P'$  wie folgt gelöst werden
    - Bei Eingabe  $x$  bestimme  $f(x)$  ( $f$  ist die Reduktionsfunktion)
    - Löse  $f(x)$  mit der Lösungsmethode für  $P$
    - Es gilt  $x \in P' \Leftrightarrow f(x) \in P$ , also überträgt sich das Ergebnis

# BEWEISFÜHRUNG DURCH REDUKTION

- $P' \leq P$  bedeutet “ $P'$  ist leichter als  $P$ ”
  - Ist  $P$  lösbar, dann kann  $P'$  wie folgt gelöst werden
    - Bei Eingabe  $x$  bestimme  $f(x)$  ( $f$  ist die Reduktionsfunktion)
    - Löse  $f(x)$  mit der Lösungsmethode für  $P$
    - Es gilt  $x \in P' \Leftrightarrow f(x) \in P$ , also überträgt sich das Ergebnis
- Aus  $P' \leq P$  und  $P$  entscheidbar folgt  $P'$  entscheidbar
  - Übertragung von Entscheidbarkeit:  $\chi_{P'}(x) = \chi_{f^{-1}(P)}(x) = \chi_P(f(x))$

# BEWEISFÜHRUNG DURCH REDUKTION

- $P' \leq P$  bedeutet “ $P'$  ist leichter als  $P$ ”
  - Ist  $P$  lösbar, dann kann  $P'$  wie folgt gelöst werden
    - Bei Eingabe  $x$  bestimme  $f(x)$  ( $f$  ist die Reduktionsfunktion)
    - Löse  $f(x)$  mit der Lösungsmethode für  $P$
    - Es gilt  $x \in P' \Leftrightarrow f(x) \in P$ , also überträgt sich das Ergebnis
- Aus  $P' \leq P$  und  $P$  entscheidbar folgt  $P'$  entscheidbar
  - Übertragung von Entscheidbarkeit:  $\chi_{P'}(x) = \chi_{f^{-1}(P)}(x) = \chi_P(f(x))$
- Aus  $P' \leq P$  und  $P$  aufzählbar folgt  $P'$  aufzählbar
  - Übertragung von Aufzählbarkeit:  $\psi_{P'}(x) = \psi_{f^{-1}(P)}(x) = \psi_P(f(x))$

## BEISPIELE VON PROBLEMREDUKTION

- $S = \{i \mid \varphi_i(i) \neq \perp\} \leq H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$

“Das Selbstanwendbarkeitsproblem ist leichter als das Halteproblem”

## BEISPIELE VON PROBLEMREDUKTION

- $S = \{i \mid \varphi_i(i) \neq \perp\} \leq H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$

“Das Selbstanwendbarkeitsproblem ist leichter als das Halteproblem”

– Es gilt  $i \in S \Leftrightarrow (i, i) \in H$ .

## BEISPIELE VON PROBLEMREDUKTION

- $S = \{i \mid \varphi_i(i) \neq \perp\} \leq H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$

“Das Selbstanwendbarkeitsproblem ist leichter als das Halteproblem”

– Es gilt  $i \in S \Leftrightarrow (i, i) \in H$ .

– Wähle  $f(i) := (i, i)$ . Dann ist  $f$  total-berechenbar und  $S = f^{-1}(H)$

## BEISPIELE VON PROBLEMREDUKTION

●  $S = \{i \mid \varphi_i(i) \neq \perp\} \leq H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$

“Das Selbstanwendbarkeitsproblem ist leichter als das Halteproblem”

- Es gilt  $i \in S \Leftrightarrow (i, i) \in H$ .
- Wähle  $f(i) := (i, i)$ . Dann ist  $f$  total-berechenbar und  $S = f^{-1}(H)$
- Man kann auch  $H$  auf  $S$  reduzieren (aufwendig)

## BEISPIELE VON PROBLEMREDUKTION

- $S = \{i \mid \varphi_i(i) \neq \perp\} \leq H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$ 

“Das Selbstanwendbarkeitsproblem ist leichter als das Halteproblem”

  - Es gilt  $i \in S \Leftrightarrow (i, i) \in H$ .
  - Wähle  $f(i) := (i, i)$ . Dann ist  $f$  total-berechenbar und  $S = f^{-1}(H)$
  - Man kann auch  $H$  auf  $S$  reduzieren (aufwendig)
- $\overline{H} = \{(i, n) \mid \varphi_i(n) = \perp\} \leq \text{PROG}_z = \{i \mid \varphi_i = z\}$

## BEISPIELE VON PROBLEMREDUKTION

- $S = \{i \mid \varphi_i(i) \neq \perp\} \leq H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$

“Das Selbstanwendbarkeitsproblem ist leichter als das Halteproblem”

– Es gilt  $i \in S \Leftrightarrow (i, i) \in H$ .

– Wähle  $f(i) := (i, i)$ . Dann ist  $f$  total-berechenbar und  $S = f^{-1}(H)$

– Man kann auch  $H$  auf  $S$  reduzieren (aufwendig)

- $\overline{H} = \{(i, n) \mid \varphi_i(n) = \perp\} \leq \text{PROG}_z = \{i \mid \varphi_i = z\}$

– Es gilt  $(i, n) \in \overline{H} \Leftrightarrow \forall t \in \mathbb{N}. \Phi_i(n) \neq t$ .

# BEISPIELE VON PROBLEMREDUKTION

- $S = \{i \mid \varphi_i(i) \neq \perp\} \leq H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$

“Das Selbstanwendbarkeitsproblem ist leichter als das Halteproblem”

- Es gilt  $i \in S \Leftrightarrow (i, i) \in H$ .
- Wähle  $f(i) := (i, i)$ . Dann ist  $f$  total-berechenbar und  $S = f^{-1}(H)$
- Man kann auch  $H$  auf  $S$  reduzieren (aufwendig)

- $\overline{H} = \{(i, n) \mid \varphi_i(n) = \perp\} \leq \text{PROG}_z = \{i \mid \varphi_i = z\}$

- Es gilt  $(i, n) \in \overline{H} \Leftrightarrow \forall t \in \mathbb{N}. \Phi_i(n) \neq t$ .
- Da  $\Phi = \{(i, n, t) \mid \Phi_i(n) = t\}$  entscheidbar ist, gibt es ein  $j$  mit

$$\varphi_j(i, n, t) = \chi_{\Phi}(i, n, t) = \begin{cases} 1 & \text{falls } (i, n, t) \in \Phi \\ 0 & \text{sonst} \end{cases}$$

## BEISPIELE VON PROBLEMREDUKTION

- $S = \{i \mid \varphi_i(i) \neq \perp\} \leq H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$

“Das Selbstanwendbarkeitsproblem ist leichter als das Halteproblem”

- Es gilt  $i \in S \Leftrightarrow (i, i) \in H$ .
- Wähle  $f(i) := (i, i)$ . Dann ist  $f$  total-berechenbar und  $S = f^{-1}(H)$
- Man kann auch  $H$  auf  $S$  reduzieren (aufwendig)

- $\overline{H} = \{(i, n) \mid \varphi_i(n) = \perp\} \leq \text{PROG}_z = \{i \mid \varphi_i = z\}$

- Es gilt  $(i, n) \in \overline{H} \Leftrightarrow \forall t \in \mathbb{N}. \Phi_i(n) \neq t$ .
- Da  $\Phi = \{(i, n, t) \mid \Phi_i(n) = t\}$  entscheidbar ist, gibt es ein  $j$  mit
$$\varphi_j(i, n, t) = \chi_{\Phi}(i, n, t) = \begin{cases} 1 & \text{falls } (i, n, t) \in \Phi \\ 0 & \text{sonst} \end{cases}$$
- Nach dem SMN Theorem gibt es eine total-berechenbare Funktion  $f$  mit  $\varphi_{f(i,n)}(t) = \varphi_j(i, n, t)$

## BEISPIELE VON PROBLEMREDUKTION

- $S = \{i \mid \varphi_i(i) \neq \perp\} \leq H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$

“Das Selbstanwendbarkeitsproblem ist leichter als das Halteproblem”

- Es gilt  $i \in S \Leftrightarrow (i, i) \in H$ .
- Wähle  $f(i) := (i, i)$ . Dann ist  $f$  total-berechenbar und  $S = f^{-1}(H)$
- Man kann auch  $H$  auf  $S$  reduzieren (aufwendig)

- $\overline{H} = \{(i, n) \mid \varphi_i(n) = \perp\} \leq PROG_z = \{i \mid \varphi_i = z\}$

- Es gilt  $(i, n) \in \overline{H} \Leftrightarrow \forall t \in \mathbb{N}. \Phi_i(n) \neq t$ .
- Da  $\Phi = \{(i, n, t) \mid \Phi_i(n) = t\}$  entscheidbar ist, gibt es ein  $j$  mit
$$\varphi_j(i, n, t) = \chi_{\Phi}(i, n, t) = \begin{cases} 1 & \text{falls } (i, n, t) \in \Phi \\ 0 & \text{sonst} \end{cases}$$
- Nach dem SMN Theorem gibt es eine total-berechenbare Funktion  $f$  mit  $\varphi_{f(i,n)}(t) = \varphi_j(i, n, t)$
- Es folgt  $(i, n) \in \overline{H} \Leftrightarrow \forall t \in \mathbb{N}. \Phi_i(n) \neq t \Leftrightarrow \forall t \in \mathbb{N}. \varphi_{f(i,n)}(t) = 0 \Leftrightarrow \varphi_{f(i,n)} = z$   
 $\Leftrightarrow f(i, n) \in PROG_z$

## BEISPIELE VON PROBLEMREDUKTION

- $S = \{i \mid \varphi_i(i) \neq \perp\} \leq H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$

“Das Selbstanwendbarkeitsproblem ist leichter als das Halteproblem”

- Es gilt  $i \in S \Leftrightarrow (i, i) \in H$ .
- Wähle  $f(i) := (i, i)$ . Dann ist  $f$  total-berechenbar und  $S = f^{-1}(H)$
- Man kann auch  $H$  auf  $S$  reduzieren (aufwendig)

- $\overline{H} = \{(i, n) \mid \varphi_i(n) = \perp\} \leq PROG_z = \{i \mid \varphi_i = z\}$

- Es gilt  $(i, n) \in \overline{H} \Leftrightarrow \forall t \in \mathbb{N}. \Phi_i(n) \neq t$ .
- Da  $\Phi = \{(i, n, t) \mid \Phi_i(n) = t\}$  entscheidbar ist, gibt es ein  $j$  mit
$$\varphi_j(i, n, t) = \chi_{\Phi}(i, n, t) = \begin{cases} 1 & \text{falls } (i, n, t) \in \Phi \\ 0 & \text{sonst} \end{cases}$$
- Nach dem SMN Theorem gibt es eine total-berechenbare Funktion  $f$  mit  $\varphi_{f(i,n)}(t) = \varphi_j(i, n, t)$
- Es folgt  $(i, n) \in \overline{H} \Leftrightarrow \forall t \in \mathbb{N}. \Phi_i(n) \neq t \Leftrightarrow \forall t \in \mathbb{N}. \varphi_{f(i,n)}(t) = 0 \Leftrightarrow \varphi_{f(i,n)} = z$   
 $\Leftrightarrow f(i, n) \in PROG_z$

also  $\overline{H} = f^{-1}(PROG_z)$

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit

# PROBLEMREDUKTION MIT ABSCHLUSSEIGENSCHAFTEN

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit
- **Methodik**
  - Zeige, daß Lösung für  $P$  ein unlösbares Problem  $P'$  lösen würde

# PROBLEMREDUKTION MIT ABSCHLUSSEIGENSCHAFTEN

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit
- **Methodik**
  - Zeige, daß Lösung für  $P$  ein unlösbares Problem  $P'$  lösen würde
  - Zeige, wie Lösung eines bekannten Problems  $P'$  zur Lösung von  $P$  führt

# PROBLEMREDUKTION MIT ABSCHLUSSEIGENSCHAFTEN

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit
- **Methodik**
  - Zeige, daß Lösung für  $P$  ein unlösbares Problem  $P'$  lösen würde
  - Zeige, wie Lösung eines bekannten Problems  $P'$  zur Lösung von  $P$  führt
- **Hilfsmittel: Abschlußigenschaften**
  - $M, M'$  entscheidbar, dann auch  $M \cup M', M \cap M', M \setminus M', \overline{M}, f^{-1}(M)$
  - $M, M'$  aufzählbar, dann auch  $M \cup M', M \cap M', g(M), g^{-1}(M)$
  - $M$  entscheidbar  $\Leftrightarrow M$  und  $\overline{M}$  aufzählbar
  - Reduzierbarkeit ist ein besonders mächtiger Spezialfall

# PROBLEMREDUKTION MIT ABSCHLUSSEIGENSCHAFTEN

- **Ziel: Wiederverwendung bekannter Ergebnisse**
  - Zur Lösung eines Problems  $P$  bzw. zum Nachweis seiner Unlösbarkeit
- **Methodik**
  - Zeige, daß Lösung für  $P$  ein unlösbares Problem  $P'$  lösen würde
  - Zeige, wie Lösung eines bekannten Problems  $P'$  zur Lösung von  $P$  führt
- **Hilfsmittel: Abschlußigenschaften**
  - $M, M'$  entscheidbar, dann auch  $M \cup M', M \cap M', M \setminus M', \overline{M}, f^{-1}(M)$
  - $M, M'$  aufzählbar, dann auch  $M \cup M', M \cap M', g(M), g^{-1}(M)$
  - $M$  entscheidbar  $\Leftrightarrow M$  und  $\overline{M}$  aufzählbar
  - Reduzierbarkeit ist ein besonders mächtiger Spezialfall
- **Umkehrung der Abschlußigenschaften**
  - $M$  nicht entscheidbar  $\Rightarrow \overline{M}$  nicht entscheidbar
  - $M$  aufzählbar, nicht entscheidbar  $\Rightarrow \overline{M}$  weder aufzählbar noch entscheidbar
  - $M$  entscheidbar,  $M \cup M'$  nicht entscheidbar  $\Rightarrow M'$  nicht entscheidbar
  - $M$  entscheidbar,  $M \setminus M'$  nicht entscheidbar  $\Rightarrow M'$  nicht entscheidbar
  - ⋮

## RESULTATE AUS ABSCHLUSSEIGENSCHAFTEN

- $\{(i, n) \mid \varphi_i(n) = \perp\}$  ist nicht aufzählbar
  - $\{(i, n) \mid \varphi_i(n) = \perp\}$  ist das Komplement von  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$
  - $H$  ist aufzählbar aber unentscheidbar, also kann  $\overline{H}$  nicht aufzählbar sein

## RESULTATE AUS ABSCHLUSSEIGENSCHAFTEN

- $\{(i, n) \mid \varphi_i(n) = \perp\}$  ist nicht aufzählbar
  - $\{(i, n) \mid \varphi_i(n) = \perp\}$  ist das Komplement von  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$
  - $H$  ist aufzählbar aber unentscheidbar, also kann  $\overline{H}$  nicht aufzählbar sein
- $\{i \mid \varphi_i(i) = \perp\}$  ist nicht aufzählbar
  - $\{i \mid \varphi_i(i) = \perp\}$  ist das Komplement von  $S = \{i \mid \varphi_i(i) \neq \perp\}$
  - $S$  ist aufzählbar aber unentscheidbar, also kann  $\overline{S}$  nicht aufzählbar sein

## RESULTATE AUS ABSCHLUSSEIGENSCHAFTEN

- $\{(i, n) \mid \varphi_i(n) = \perp\}$  ist nicht aufzählbar
  - $\{(i, n) \mid \varphi_i(n) = \perp\}$  ist das Komplement von  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$
  - $H$  ist aufzählbar aber unentscheidbar, also kann  $\overline{H}$  nicht aufzählbar sein
- $\{i \mid \varphi_i(i) = \perp\}$  ist nicht aufzählbar
  - $\{i \mid \varphi_i(i) = \perp\}$  ist das Komplement von  $S = \{i \mid \varphi_i(i) \neq \perp\}$
  - $S$  ist aufzählbar aber unentscheidbar, also kann  $\overline{S}$  nicht aufzählbar sein
- $PROG_z = \{i \mid \varphi_i = z\}$  ist nicht aufzählbar
  - Es gilt  $\overline{H} \leq PROG_z$  und  $\overline{H}$  ist nicht aufzählbar

## RESULTATE AUS ABSCHLUSSEIGENSCHAFTEN

- $\{(i, n) \mid \varphi_i(n) = \perp\}$  ist nicht aufzählbar
  - $\{(i, n) \mid \varphi_i(n) = \perp\}$  ist das Komplement von  $H = \{(i, n) \mid \varphi_i(n) \neq \perp\}$
  - $H$  ist aufzählbar aber unentscheidbar, also kann  $\overline{H}$  nicht aufzählbar sein
- $\{i \mid \varphi_i(i) = \perp\}$  ist nicht aufzählbar
  - $\{i \mid \varphi_i(i) = \perp\}$  ist das Komplement von  $S = \{i \mid \varphi_i(i) \neq \perp\}$
  - $S$  ist aufzählbar aber unentscheidbar, also kann  $\overline{S}$  nicht aufzählbar sein
- $PROG_z = \{i \mid \varphi_i = z\}$  ist nicht aufzählbar
  - Es gilt  $\overline{H} \leq PROG_z$  und  $\overline{H}$  ist nicht aufzählbar
- $PF_\varphi = \{i \mid \varphi_i \text{ partiell}\}$  ist unentscheidbar
  - $PF_\varphi$  ist das Komplement von  $\mathcal{R}_\varphi = \{i \mid \varphi_i \text{ total}\}$
  - $\mathcal{R}_\varphi$  ist nicht aufzählbar, also kann  $PF_\varphi = \overline{\mathcal{R}_\varphi}$  nicht entscheidbar sein

# WELCHE VERIFIKATIONSPROBLEME SIND ENTSCHIEDBAR?

- **Halteproblem**

- Kann man von einem beliebigen Programm entscheiden, ob es bei bestimmten Eingaben hält oder nicht?
- Bereits als unentscheidbar nachgewiesen

# WELCHE VERIFIKATIONSPROBLEME SIND ENTSCHIEDBAR?

- **Halteproblem**

- Kann man von einem beliebigen Programm entscheiden, ob es bei bestimmten Eingaben hält oder nicht?
- Bereits als unentscheidbar nachgewiesen

- **Korrektheitsproblem**

- Kann man von einem beliebigen Programm entscheiden, ob es eine bestimmte Funktion berechnet oder nicht?
- Für die Nullfunktion bereits als unentscheidbar nachgewiesen
- Gilt ähnliches für andere Funktionen?

# WELCHE VERIFIKATIONSPROBLEME SIND ENTSCHIEDBAR?

- **Halteproblem**

- Kann man von einem beliebigen Programm entscheiden, ob es bei bestimmten Eingaben hält oder nicht?
- Bereits als unentscheidbar nachgewiesen

- **Korrektheitsproblem**

- Kann man von einem beliebigen Programm entscheiden, ob es eine bestimmte Funktion berechnet oder nicht?
- Für die Nullfunktion bereits als unentscheidbar nachgewiesen
- Gilt ähnliches für andere Funktionen?

- **Spezifikationsproblem**

- Kann man von einem beliebigen Programm entscheiden, ob es eine gegebene Spezifikation erfüllt oder nicht?

# WELCHE VERIFIKATIONSPROBLEME SIND ENTSCHIEDBAR?

## ● Halteproblem

- Kann man von einem beliebigen Programm entscheiden, ob es bei bestimmten Eingaben hält oder nicht?
- Bereits als unentscheidbar nachgewiesen

## ● Korrektheitsproblem

- Kann man von einem beliebigen Programm entscheiden, ob es eine bestimmte Funktion berechnet oder nicht?
- Für die Nullfunktion bereits als unentscheidbar nachgewiesen
- Gilt ähnliches für andere Funktionen?

## ● Spezifikationsproblem

- Kann man von einem beliebigen Programm entscheiden, ob es eine gegebene Spezifikation erfüllt oder nicht?

## ● Äquivalenzproblem

- Kann man entscheiden, ob zwei beliebige Programme die gleiche Funktion berechnen oder nicht?

# WELCHE VERIFIKATIONSPROBLEME SIND ENTSCHIEDBAR?

- **Halteproblem**

- Kann man von einem beliebigen Programm entscheiden, ob es bei bestimmten Eingaben hält oder nicht?
- Bereits als unentscheidbar nachgewiesen

- **Korrektheitsproblem**

- Kann man von einem beliebigen Programm entscheiden, ob es eine bestimmte Funktion berechnet oder nicht?
- Für die Nullfunktion bereits als unentscheidbar nachgewiesen
- Gilt ähnliches für andere Funktionen?

- **Spezifikationsproblem**

- Kann man von einem beliebigen Programm entscheiden, ob es eine gegebene Spezifikation erfüllt oder nicht?

- **Äquivalenzproblem**

- Kann man entscheiden, ob zwei beliebige Programme die gleiche Funktion berechnen oder nicht?

**Gibt es eine allgemeine Antwort?**

# DER SATZ VON RICE

Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

# DER SATZ VON RICE

Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

# DER SATZ VON RICE

Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

– Betrachte  $g \equiv \lambda x. \perp$

# DER SATZ VON RICE

Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

- Betrachte  $g \equiv \lambda x. \perp$
- Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

# DER SATZ VON RICE

**Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar**

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

- Betrachte  $g \equiv \lambda x. \perp$
- Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

- Dann ist  $h'$  berechenbar und nach dem SMN Theorem gibt es ein total-berechenbares  $f$  mit  $h'(i, x) = \varphi_{f(i)}(x)$

# DER SATZ VON RICE

Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

- Betrachte  $g \equiv \lambda x. \perp$
- Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

- Dann ist  $h'$  berechenbar und nach dem SMN Theorem gibt es ein total-berechenbares  $f$  mit  $h'(i, x) = \varphi_{f(i)}(x)$
- Es folgt:  $i \in S$

# DER SATZ VON RICE

**Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar**

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

- Betrachte  $g \equiv \lambda x. \perp$
- Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

- Dann ist  $h'$  berechenbar und nach dem SMN Theorem gibt es ein total-berechenbares  $f$  mit  $h'(i, x) = \varphi_{f(i)}(x)$
- Es folgt:  $i \in S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = h(x)$

# DER SATZ VON RICE

**Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar**

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

- Betrachte  $g \equiv \lambda x. \perp$
- Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

- Dann ist  $h'$  berechenbar und nach dem SMN Theorem gibt es ein total-berechenbares  $f$  mit  $h'(i, x) = \varphi_{f(i)}(x)$
- Es folgt:  $i \in S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = h(x) \Rightarrow \varphi_{f(i)} = h \in P$

# DER SATZ VON RICE

**Keine nichttriviale extensionale Eigenschaft berechenbarer Funktionen ist entscheidbar**

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

– Betrachte  $g \equiv \lambda x. \perp$

– Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

– Dann ist  $h'$  berechenbar und nach dem SMN Theorem gibt es ein total-berechenbares  $f$  mit  $h'(i, x) = \varphi_{f(i)}(x)$

– Es folgt:  $i \in S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = h(x) \Rightarrow \varphi_{f(i)} = h \in P \Rightarrow f(i) \in \mathcal{L}_P$

# DER SATZ VON RICE

Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

– Betrachte  $g \equiv \lambda x. \perp$

– Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

– Dann ist  $h'$  berechenbar und nach dem SMN Theorem gibt es ein total-berechenbares  $f$  mit  $h'(i, x) = \varphi_{f(i)}(x)$

– Es folgt:  $i \in S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = h(x) \Rightarrow \varphi_{f(i)} = h \in P \Rightarrow f(i) \in \mathcal{L}_P$   
 $i \notin S$

# DER SATZ VON RICE

**Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar**

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

– Betrachte  $g \equiv \lambda x. \perp$

– Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

– Dann ist  $h'$  berechenbar und nach dem SMN Theorem gibt es ein total-berechenbares  $f$  mit  $h'(i, x) = \varphi_{f(i)}(x)$

– Es folgt:  $i \in S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = h(x) \Rightarrow \varphi_{f(i)} = h \in P \Rightarrow f(i) \in \mathcal{L}_P$   
 $i \notin S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = \perp$

# DER SATZ VON RICE

**Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar**

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

– Betrachte  $g \equiv \lambda x. \perp$

– Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

– Dann ist  $h'$  berechenbar und nach dem SMN Theorem gibt es ein total-berechenbares  $f$  mit  $h'(i, x) = \varphi_{f(i)}(x)$

– Es folgt:  $i \in S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = h(x) \Rightarrow \varphi_{f(i)} = h \in P \Rightarrow f(i) \in \mathcal{L}_P$   
 $i \notin S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = \perp \Rightarrow \varphi_{f(i)} = g \notin P$

# DER SATZ VON RICE

**Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar**

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

- Betrachte  $g \equiv \lambda x. \perp$
- Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

- Dann ist  $h'$  berechenbar und nach dem SMN Theorem gibt es ein total-berechenbares  $f$  mit  $h'(i, x) = \varphi_{f(i)}(x)$
- Es folgt:  $i \in S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = h(x) \Rightarrow \varphi_{f(i)} = h \in P \Rightarrow f(i) \in \mathcal{L}_P$   
 $i \notin S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = \perp \Rightarrow \varphi_{f(i)} = g \notin P \Rightarrow f(i) \notin \mathcal{L}_P$

# DER SATZ VON RICE

**Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar**

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

– Betrachte  $g \equiv \lambda x. \perp$

– Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

– Dann ist  $h'$  berechenbar und nach dem SMN Theorem gibt es ein total-berechenbares  $f$  mit  $h'(i, x) = \varphi_{f(i)}(x)$

– Es folgt:  $i \in S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = h(x) \Rightarrow \varphi_{f(i)} = h \in P \Rightarrow f(i) \in \mathcal{L}_P$

$i \notin S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = \perp \Rightarrow \varphi_{f(i)} = g \notin P \Rightarrow f(i) \notin \mathcal{L}_P$

– Insgesamt  $i \in S \Leftrightarrow f(i) \in \mathcal{L}_P$ , also  $S \leq \mathcal{L}_P$

# DER SATZ VON RICE

**Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar**

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

– Betrachte  $g \equiv \lambda x. \perp$

– Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

– Dann ist  $h'$  berechenbar und nach dem SMN Theorem gibt es ein total-berechenbares  $f$  mit  $h'(i, x) = \varphi_{f(i)}(x)$

– Es folgt:  $i \in S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = h(x) \Rightarrow \varphi_{f(i)} = h \in P \Rightarrow f(i) \in \mathcal{L}_P$

$i \notin S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = \perp \Rightarrow \varphi_{f(i)} = g \notin P \Rightarrow f(i) \notin \mathcal{L}_P$

– Insgesamt  $i \in S \Leftrightarrow f(i) \in \mathcal{L}_P$ , also  $S \leq \mathcal{L}_P$

– Da  $S$  unentscheidbar ist, muß dies auch für  $\mathcal{L}_P$  gelten



# DER SATZ VON RICE

**Keine nichttriviale extensionale Eigenschaft  
berechenbarer Funktionen ist entscheidbar**

Für  $\emptyset \neq P \subset \mathcal{T}_\mu$  ist  $\mathcal{L}_P = \{i \mid \varphi_i \in P\}$  nicht entscheidbar

Beweis durch Reduktion auf  $S = \{i \mid \varphi_i(i) \neq \perp\}$

- Betrachte  $g \equiv \lambda x. \perp$
- Falls  $g \notin P$ , so wähle  $h \in P$  beliebig und definiere

$$h'(i, x) = \begin{cases} h(x) & \text{falls } \varphi_i(i) \neq \perp \\ \perp & \text{sonst} \end{cases}$$

- Dann ist  $h'$  berechenbar und nach dem SMN Theorem gibt es ein total-berechenbares  $f$  mit  $h'(i, x) = \varphi_{f(i)}(x)$
- Es folgt:  $i \in S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = h(x) \Rightarrow \varphi_{f(i)} = h \in P \Rightarrow f(i) \in \mathcal{L}_P$   
 $i \notin S \Rightarrow \forall x. h'(i, x) = \varphi_{f(i)}(x) = \perp \Rightarrow \varphi_{f(i)} = g \notin P \Rightarrow f(i) \notin \mathcal{L}_P$
- Insgesamt  $i \in S \Leftrightarrow f(i) \in \mathcal{L}_P$ , also  $S \leq \mathcal{L}_P$
- Da  $S$  unentscheidbar ist, muß dies auch für  $\mathcal{L}_P$  gelten ✓
- Falls  $g \in P$  wähle ein beliebiges  $h \notin P$  und zeige so  $S \leq \overline{P}$  ✓

## ANWENDUNGEN DES SATZES VON RICE

- $MON = \{i \mid \forall k. \varphi_i(k) < \varphi_i(k+1)\}$ 
  - Monotone Funktionen sind unentscheidbar

## ANWENDUNGEN DES SATZES VON RICE

- $MON = \{i \mid \forall k. \varphi_i(k) < \varphi_i(k+1)\}$ 
  - Monotone Funktionen sind unentscheidbar
- $EF = \{i \mid \forall j. \varphi_i(j) \in \{0, 1\}\}$ 
  - Entscheidungsfunktionen sind unentscheidbar

## ANWENDUNGEN DES SATZES VON RICE

- $MON = \{i \mid \forall k. \varphi_i(k) < \varphi_i(k+1)\}$ 
  - Monotone Funktionen sind unentscheidbar
- $EF = \{i \mid \forall j. \varphi_i(j) \in \{0, 1\}\}$ 
  - Entscheidungsfunktionen sind unentscheidbar
- $PROG_{spec} = \{i \mid \varphi_i \text{ erfüllt Spezifikation } spec\}$ 
  - Allgemeines Spezifikationsproblem ist unentscheidbar

## ANWENDUNGEN DES SATZES VON RICE

- $MON = \{i \mid \forall k. \varphi_i(k) < \varphi_i(k+1)\}$ 
  - Monotone Funktionen sind unentscheidbar
- $EF = \{i \mid \forall j. \varphi_i(j) \in \{0, 1\}\}$ 
  - Entscheidungsfunktionen sind unentscheidbar
- $PROG_{spec} = \{i \mid \varphi_i \text{ erfüllt Spezifikation } spec\}$ 
  - Allgemeines Spezifikationsproblem ist unentscheidbar
- $PROG_f = \{i \mid \varphi_i = f\}$ 
  - Korrektheitsproblem ist unentscheidbar

## ANWENDUNGEN DES SATZES VON RICE

- $MON = \{i \mid \forall k. \varphi_i(k) < \varphi_i(k+1)\}$ 
  - Monotone Funktionen sind unentscheidbar
- $EF = \{i \mid \forall j. \varphi_i(j) \in \{0, 1\}\}$ 
  - Entscheidungsfunktionen sind unentscheidbar
- $PROG_{spec} = \{i \mid \varphi_i \text{ erfüllt Spezifikation } spec\}$ 
  - Allgemeines Spezifikationsproblem ist unentscheidbar
- $PROG_f = \{i \mid \varphi_i = f\}$ 
  - Korrektheitsproblem ist unentscheidbar
- $EQ = \{(i, j) \mid \varphi_i = \varphi_j\}$ 
  - Äquivalenzproblem ist unentscheidbar

# ANWENDUNGEN DES SATZES VON RICE

- $MON = \{i \mid \forall k. \varphi_i(k) < \varphi_i(k+1)\}$ 
  - Monotone Funktionen sind unentscheidbar
- $EF = \{i \mid \forall j. \varphi_i(j) \in \{0, 1\}\}$ 
  - Entscheidungsfunktionen sind unentscheidbar
- $PROG_{spec} = \{i \mid \varphi_i \text{ erfüllt Spezifikation } spec\}$ 
  - Allgemeines Spezifikationsproblem ist unentscheidbar
- $PROG_f = \{i \mid \varphi_i = f\}$ 
  - Korrektheitsproblem ist unentscheidbar
- $EQ = \{(i, j) \mid \varphi_i = \varphi_j\}$ 
  - Äquivalenzproblem ist unentscheidbar
- $RG = \{(i, j) \mid j \in \text{range}(\varphi_i)\}$ 
  - Bildbereiche sind unentscheidbar

## ANWENDUNGEN DES SATZES VON RICE

- $MON = \{i \mid \forall k. \varphi_i(k) < \varphi_i(k+1)\}$ 
  - Monotone Funktionen sind unentscheidbar
- $EF = \{i \mid \forall j. \varphi_i(j) \in \{0, 1\}\}$ 
  - Entscheidungsfunktionen sind unentscheidbar
- $PROG_{spec} = \{i \mid \varphi_i \text{ erfüllt Spezifikation } spec\}$ 
  - Allgemeines Spezifikationsproblem ist unentscheidbar
- $PROG_f = \{i \mid \varphi_i = f\}$ 
  - Korrektheitsproblem ist unentscheidbar
- $EQ = \{(i, j) \mid \varphi_i = \varphi_j\}$ 
  - Äquivalenzproblem ist unentscheidbar
- $RG = \{(i, j) \mid j \in \text{range}(\varphi_i)\}$ 
  - Bildbereiche sind unentscheidbar

**Keine Programmeigenschaft kann getestet werden**  
**Beweise müssen von Hand geführt werden**  
**Rechnerunterstützung nur in Spezialfällen möglich**