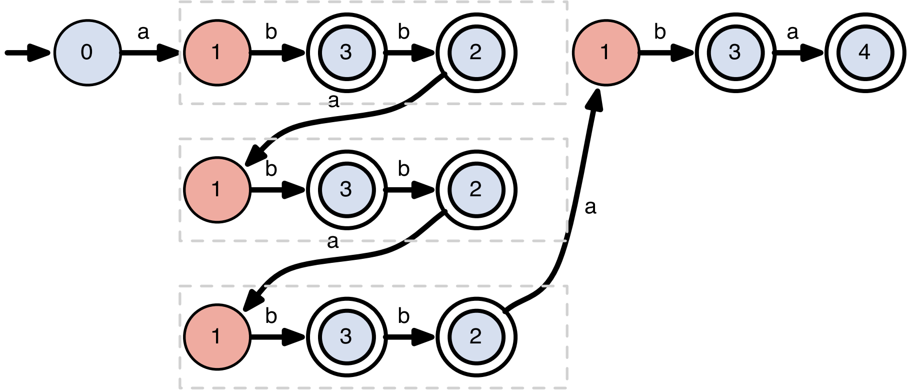
Berechenbarkeitstheorie

14. Vorlesung



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

Erinnerung - Abzählbarkeit

Erinnerung - Abzählbarkeit

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

Erinnerung - Abzählbarkeit

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

Definition

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

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

Erinnerung - Abzählbarkeit

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

Definition

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

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

Beispiel: $\{0, 1\}^*$ ist abzählbar.

Erinnerung - Abzählbarkeit

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

Definition

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

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

Beispiel: $\{0, 1\}^*$ ist abzählbar.

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

Erinnerung - Abzählbarkeit

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

Definition

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

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

Beispiel: $\{0, 1\}^*$ ist abzählbar.

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

Satz 18

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

Erinnerung - Abzählbarkeit

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

Definition

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

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

Beispiel: $\{0,1\}^*$ ist abzählbar.

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

Satz 18

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

- **Idee:** Diagonalisierung

Korollar

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

Korollar

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

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

Korollar

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

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

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Korollar

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

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

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Vorüberlegung :

- eine TM kann durch einen endliches Wort kodiert werden

Korollar

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

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

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Vorüberlegung :

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

Korollar

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

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

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Vorüberlegung :

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

Korollar

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

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

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Vorüberlegung :

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

Korollar

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

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

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Vorüberlegung :

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

0 entspricht a ,00 entspricht b, usw.

1 entspricht L
11 entspricht R

Korollar

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

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

Satz 19

Es gibt eine Sprache, die nicht erkennbar ist.

Vorüberlegung :

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

    0 entspricht a ,00 entspricht b, usw.

    1 entspricht L
    11 entspricht R

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

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

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

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

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

- jede TM erscheint als Kodierung T_w für ein Wort w

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

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

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

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

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

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

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

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

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

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

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

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

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

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

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

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

Beweis Satz 19:

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

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

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

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

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

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

Beweis Satz 19:

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

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

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

- jede TM erscheint als Kodierung T_w für ein Wort w

Beweis Satz 19:

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

Zusammenfassung

- wir können jeder TM ein Wort über $\{0, 1\}^*$ eindeutig zuordnen

Zusammenfassung

- wir können jeder TM ein Wort über $\{0, 1\}^*$ eindeutig zuordnen
- also gibt es nur abzählbar viele TM, und auch nur abzählbar viele Sprachen die von TM erkannt werden

Zusammenfassung

- wir können jeder TM ein Wort über $\{0, 1\}^*$ eindeutig zuordnen
- also gibt es nur abzählbar viele TM, und auch nur abzählbar viele Sprachen die von TM erkannt werden
- es gibt aber überabzählbar viel Funktionen von $\{0, 1\}^* \to \{0, 1\}$

	ε	0	1	00	01	10	11	000	
f_0	0	1	0	0	0	1	1	1	
f_1	0	0	1	1	0	0	1	1	$\ldots$
f_2	0	0	0	1	0	0	1	1	
f_3	1	1	0	1	0	0	0	1	
$\vdots$					$\vdots$				
g	1	1	1	0	0	1	0	1	

Zusammenfassung

- wir können jeder TM ein Wort über $\{0,1\}^*$ eindeutig zuordnen
- also gibt es nur abzählbar viele TM, und auch nur abzählbar viele Sprachen die von TM erkannt werden
- es gibt aber überabzählbar viel Funktionen von $\{0,1\}^* \to \{0,1\}$

	ε	0	1	00	01	10	11	000	
f_0	0	1	0	0	0	1	1	1	
f_1	0	0	1	1	0	0	1	1	$\dots$
f_2	0	0	0	1	0	0	1	1	
f_3	1	1	0	1	0	0	0	1	
$\vdots$					$\vdots$				
g	1	1	1	0	0	1	0	1	

- diese Funktionen kodieren (bijektiv) Sprachen über $\{0,1\}$

Zusammenfassung

- wir können jeder TM ein Wort über $\{0,1\}^*$ eindeutig zuordnen
- also gibt es nur abzählbar viele TM, und auch nur abzählbar viele Sprachen die von TM erkannt werden
- es gibt aber überabzählbar viel Funktionen von $\{0,1\}^* \to \{0,1\}$

	ε	0	1	00	01	10	11	000	
f_0	0	1	0	0	0	1	1	1	
f_1	0	0	1	1	0	0	1	1	$\dots$
f_2	0	0	0	1	0	0	1	1	
f_3	1	1	0	1	0	0	0	1	
$\vdots$					$\vdots$				
g	1	1	1	0	0	1	0	1	

- diese Funktionen kodieren (bijektiv) Sprachen über $\{0,1\}$
- es gibt mehr Sprachen als erkennbare Sprachen, also gibt es Sprachen die nicht erkennbar sind

Simulation

- gesucht ist TM U, die andere TM simulieren kann

Simulation

- gesucht ist TM U, die andere TM simulieren kann
- $U(\langle M, w\rangle)$ soll genau dann akzeptieren, wenn $M(w)$ akzeptiert

Simulation

- gesucht ist TM U, die andere TM simulieren kann
- $U(\langle M, w\rangle)$ soll genau dann akzeptieren, wenn $M(w)$ akzeptiert

Aufbau von U

- U besitzt Simulationband,welches die aktuelle Konfiguration von M enthält

Simulation

- gesucht ist TM U, die andere TM simulieren kann
- $U(\langle M, w\rangle)$ soll genau dann akzeptieren, wenn $M(w)$ akzeptiert

Aufbau von U

- U besitzt Simulationband,welches die aktuelle Konfiguration von M enthält
- Befehle werden auf einem Befehlsband dekodiert (als Sequenz)

Simulation

- gesucht ist TM U, die andere TM simulieren kann
- $U(\langle M, w\rangle)$ soll genau dann akzeptieren, wenn $M(w)$ akzeptiert

Aufbau von U

- U besitzt Simulationband,welches die aktuelle Konfiguration von M enthält
- Befehle werden auf einem Befehlsband dekodiert (als Sequenz)

Simulation eines Schrittes von $M(w)$

- suche auf dem Befehlsband nach einem ausführbaren Befehl

Simulation

- gesucht ist TM U, die andere TM simulieren kann
- $U(\langle M, w\rangle)$ soll genau dann akzeptieren, wenn $M(w)$ akzeptiert

Aufbau von U

- U besitzt Simulationband,welches die aktuelle Konfiguration von M enthält
- Befehle werden auf einem Befehlsband dekodiert (als Sequenz)

Simulation eines Schrittes von $M(w)$

- suche auf dem Befehlsband nach einem ausführbaren Befehl
- Modifiziere entsprechend des Befehls das Simulationsband (ersetzen durch Folgekonfiguration)

Simulation

- gesucht ist TM U, die andere TM simulieren kann
- $U(\langle M, w\rangle)$ soll genau dann akzeptieren, wenn $M(w)$ akzeptiert

Aufbau von U

- U besitzt Simulationband,welches die aktuelle Konfiguration von M enthält
- Befehle werden auf einem Befehlsband dekodiert (als Sequenz)

Simulation eines Schrittes von $M(w)$

- suche auf dem Befehlsband nach einem ausführbaren Befehl
- Modifiziere entsprechend des Befehls das Simulationsband (ersetzen durch Folgekonfiguration)

Programm von U

- Schreibe Startkonfiguration aufs Simulationsband

Simulation

- gesucht ist TM U, die andere TM simulieren kann
- $U(\langle M, w\rangle)$ soll genau dann akzeptieren, wenn $M(w)$ akzeptiert

Aufbau von U

- U besitzt Simulationband,welches die aktuelle Konfiguration von M enthält
- Befehle werden auf einem Befehlsband dekodiert (als Sequenz)

Simulation eines Schrittes von $M(w)$

- suche auf dem Befehlsband nach einem ausführbaren Befehl
- Modifiziere entsprechend des Befehls das Simulationsband (ersetzen durch Folgekonfiguration)

Programm von U

- Schreibe Startkonfiguration aufs Simulationsband
- Simuliere einen Schritt von $M(w)$

Simulation

- gesucht ist TM U, die andere TM simulieren kann
- $U(\langle M, w\rangle)$ soll genau dann akzeptieren, wenn $M(w)$ akzeptiert

Aufbau von U

- U besitzt Simulationband,welches die aktuelle Konfiguration von M enthält
- Befehle werden auf einem Befehlsband dekodiert (als Sequenz)

Simulation eines Schrittes von $M(w)$

- suche auf dem Befehlsband nach einem ausführbaren Befehl
- Modifiziere entsprechend des Befehls das Simulationsband (ersetzen durch Folgekonfiguration)

Programm von U

- Schreibe Startkonfiguration aufs Simulationsband
- Simuliere einen Schritt von $M(w)$
- Wiederhole, bis eine akzeptierende oder verwerfende Konfiguration vorliegt

Simulation

Universelle Turingmaschine

- gesucht ist TM U, die andere TM simulieren kann
- $U(\langle M, w\rangle)$ soll genau dann akzeptieren, wenn $M(w)$ akzeptiert

Aufbau von U

- U besitzt Simulationband,welches die aktuelle Konfiguration von M enthält
- Befehle werden auf einem Befehlsband dekodiert (als Sequenz)

Simulation eines Schrittes von $M(w)$

- suche auf dem Befehlsband nach einem ausführbaren Befehl
- Modifiziere entsprechend des Befehls das Simulationsband (ersetzen durch Folgekonfiguration)

Programm von U

- Schreibe Startkonfiguration aufs Simulationsband
- Simuliere einen Schritt von $M(w)$
- Wiederhole, bis eine akzeptierende oder verwerfende Konfiguration vorliegt

Satz 20

Die Sprache $A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\}$ ist **nicht entscheidbar**.

Satz 20

Die Sprache $A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\}$ ist **nicht entscheidbar**.

Beweis

Angenommen $A_{\mathsf{TM}} \in \mathbb{E}$, dann existiert für A_{TM} ein Entscheider H.

$$H(\langle M, w\rangle) = \begin{cases} \text{akzeptiert} & \text{M akzeptiert } w \\ \text{verwirft} & \text{M akzeptiert } w \text{ nicht} \end{cases}$$

Satz 20

Die Sprache $A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\}$ ist **nicht entscheidbar**.

Beweis

Angenommen $A_{\mathsf{TM}} \in \mathbb{E}$, dann existiert für A_{TM} ein Entscheider H.

$$H(\langle M, w\rangle) = \begin{cases} \text{akzeptiert} & \text{M akzeptiert } w \\ \text{verwirft} & \text{M akzeptiert } w \text{ nicht} \end{cases}$$

Wir konstruieren die Diagonalisierungsmaschine $D(\langle M\rangle)$ wie folgt:

Satz 20

Die Sprache $A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\}$ ist **nicht entscheidbar**.

Beweis

Angenommen $A_{\mathsf{TM}} \in \mathbb{E}$, dann existiert für A_{TM} ein Entscheider H.

$$H(\langle M, w\rangle) = \begin{cases} \text{akzeptiert} & \text{M akzeptiert } w \\ \text{verwirft} & \text{M akzeptiert } w \text{ nicht} \end{cases}$$

Wir konstruieren die Diagonalisierungsmaschine $D(\langle M\rangle)$ wie folgt:

1. D simuliert $H(\langle M, \langle M\rangle\rangle)$
2. D gibt das entgegengesetzte Ergebnis der Simulation zurück

Satz 20

Die Sprache $A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\}$ ist **nicht entscheidbar**.

Beweis

Angenommen $A_{\mathsf{TM}} \in \mathbb{E}$, dann existiert für A_{TM} ein Entscheider H.

$$H(\langle M, w\rangle) = \begin{cases} \text{akzeptiert} & \text{M akzeptiert } w \\ \text{verwirft} & \text{M akzeptiert } w \text{ nicht} \end{cases}$$

Wir konstruieren die Diagonalisierungsmaschine $D(\langle M\rangle)$ wie folgt:

1. D simuliert $H(\langle M, \langle M\rangle\rangle)$
2. D gibt das entgegengesetzte Ergebnis der Simulation zurück

Also: $$D(\langle M\rangle) = \begin{cases} \text{akzeptiert} & \text{wenn } M(\langle M\rangle) \text{ verwirft} \\ \text{verwirft} & \text{wenn } M(\langle M\rangle) \text{ akzeptiert} \end{cases}$$

Satz 20

Die Sprache $A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\}$ ist **nicht entscheidbar**.

Beweis

Angenommen $A_{\mathsf{TM}} \in \mathbb{E}$, dann existiert für A_{TM} ein Entscheider H.

$$H(\langle M, w\rangle) = \begin{cases} \text{akzeptiert} & \text{M akzeptiert } w \\ \text{verwirft} & \text{M akzeptiert } w \text{ nicht} \end{cases}$$

Wir konstruieren die Diagonalisierungsmaschine $D(\langle M\rangle)$ wie folgt:

1. D simuliert $H(\langle M, \langle M\rangle\rangle)$
2. D gibt das entgegengesetzte Ergebnis der Simulation zurück

Also: $$D(\langle M\rangle) = \begin{cases} \text{akzeptiert} & \text{wenn } M(\langle M\rangle) \text{ verwirft} \\ \text{verwirft} & \text{wenn } M(\langle M\rangle) \text{ akzeptiert} \end{cases}$$

D.h.: $$D(\langle D\rangle) = \begin{cases} \text{akzeptiert} & \text{wenn } D(\langle D\rangle) \text{ verwirft} \\ \text{verwirft} & \text{wenn } D(\langle D\rangle) \text{ akzeptiert} \end{cases}$$

Satz 20

Die Sprache $A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\}$ ist **nicht entscheidbar**.

Beweis

Angenommen $A_{\mathsf{TM}} \in \mathbb{E}$, dann existiert für A_{TM} ein Entscheider H.

$$H(\langle M, w\rangle) = \begin{cases} \text{akzeptiert} & \text{M akzeptiert } w \\ \text{verwirft} & \text{M akzeptiert } w \text{ nicht} \end{cases}$$

Wir konstruieren die Diagonalisierungsmaschine $D(\langle M\rangle)$ wie folgt:

1. D simuliert $H(\langle M, \langle M\rangle\rangle)$
2. D gibt das entgegengesetzte Ergebnis der Simulation zurück

Also: $$D(\langle M\rangle) = \begin{cases} \text{akzeptiert} & \text{wenn } M(\langle M\rangle) \text{ verwirft} \\ \text{verwirft} & \text{wenn } M(\langle M\rangle) \text{ akzeptiert} \end{cases}$$

D.h.: $$D(\langle D\rangle) = \begin{cases} \text{akzeptiert} & \text{wenn } D(\langle D\rangle) \text{ verwirft} \\ \text{verwirft} & \text{wenn } D(\langle D\rangle) \text{ akzeptiert} \end{cases}$$

$\rightarrow$ Widerspruch, D, und somit H kann nicht existieren

Anmerkung zum Beweis

Anmerkung zum Beweis

Wir benutzen das Prinzip der Diagonalisierung

Anmerkung zum Beweis

Wir benutzen das Prinzip der Diagonalisierung

Aufzählung der TM

	$\langle T_1 \rangle$	$\langle T_2 \rangle$	$\langle T_3 \rangle$	$\langle T_4 \rangle$	$\langle T_5 \rangle$	$\langle T_6 \rangle$	$\langle T_7 \rangle$	$\langle T_8 \rangle$	
T_1	A	A	A	A	A	V	V	V	
T_2	A	V	V	V	A	A	V	V	...
T_3	A	A	A	V	A	A	V	V	
T_4	V	V	A	V	A	A	A	V	

Anmerkung zum Beweis

Wir benutzen das Prinzip der Diagonalisierung

Aufzählung der TM

	$\langle T_1 \rangle$	$\langle T_2 \rangle$	$\langle T_3 \rangle$	$\langle T_4 \rangle$	$\langle T_5 \rangle$	$\langle T_6 \rangle$	$\langle T_7 \rangle$	$\langle T_8 \rangle$	
T_1	A	A	A	A	A	V	V	V	
T_2	A	V	V	V	A	A	V	V	...
T_3	A	A	A	V	A	A	V	V	
T_4	V	V	A	V	A	A	A	V	
$H(\langle M, \langle M \rangle \rangle)$	A	V	A	V	V	A	A	V	

Anmerkung zum Beweis

Wir benutzen das Prinzip der Diagonalisierung

Aufzählung der TM

	$\langle T_1\rangle$	$\langle T_2\rangle$	$\langle T_3\rangle$	$\langle T_4\rangle$	$\langle T_5\rangle$	$\langle T_6\rangle$	$\langle T_7\rangle$	$\langle T_8\rangle$	
T_1	A	A	A	A	A	V	V	V	
T_2	A	V	V	V	A	A	V	V	$\dots$
T_3	A	A	A	V	A	A	V	V	
T_4	V	V	A	V	A	A	A	V	
$H(\langle M, \langle M\rangle\rangle)$	A	V	A	V	V	A	A	V	
$D(\langle M\rangle)$	V	A	V	A	A	V	V	A	

Anmerkung zum Beweis

Wir benutzen das Prinzip der Diagonalisierung

Aufzählung der TM

	$\langle T_1\rangle$	$\langle T_2\rangle$	$\langle T_3\rangle$	$\langle T_4\rangle$	$\langle T_5\rangle$	$\langle T_6\rangle$	$\langle T_7\rangle$	$\langle T_8\rangle$	
T_1	A	A	A	A	A	V	V	V	
T_2	A	V	V	V	A	A	V	V	$\dots$
T_3	A	A	A	V	A	A	V	V	
T_4	V	V	A	V	A	A	A	V	
$H(\langle M, \langle M\rangle\rangle)$	A	V	A	V	V	A	A	V	
$D(\langle M\rangle)$	V	A	V	A	A	V	V	A	

- D **anti-simuliert** H

Anmerkung zum Beweis

Wir benutzen das Prinzip der Diagonalisierung

Aufzählung der TM

	$\langle T_1\rangle$	$\langle T_2\rangle$	$\langle T_3\rangle$	$\langle T_4\rangle$	$\langle T_5\rangle$	$\langle T_6\rangle$	$\langle T_7\rangle$	$\langle T_8\rangle$	
T_1	A	A	A	A	A	V	V	V	
T_2	A	V	V	V	A	A	V	V	...
T_3	A	A	A	V	A	A	V	V	
T_4	V	V	A	V	A	A	A	V	
$H(\langle M, \langle M\rangle\rangle)$	A	V	A	V	V	A	A	V	
$D(\langle M\rangle)$	V	A	V	A	A	V	V	A	

- D **anti-simuliert** H
- D kann nicht in der Aufzählung der T_i enthalten sein

Anmerkung zum Beweis

Wir benutzen das Prinzip der Diagonalisierung

Aufzählung der TM	$\langle T_1\rangle$	$\langle T_2\rangle$	$\langle T_3\rangle$	$\langle T_4\rangle$	$\langle T_5\rangle$	$\langle T_6\rangle$	$\langle T_7\rangle$	$\langle T_8\rangle$	
T_1	A	A	A	A	A	V	V	V	
T_2	A	V	V	V	A	A	V	V	$\dots$
T_3	A	A	A	V	A	A	V	V	
T_4	V	V	A	V	A	A	A	V	
$H(\langle M, \langle M\rangle\rangle)$	A	V	A	V	V	A	A	V	
$D(\langle M\rangle)$	V	A	V	A	A	V	V	A	

- D **anti-simuliert** H
- D kann nicht in der Aufzählung der T_i enthalten sein
- Antisimulation funktioniert nur, wenn H ein Entscheider ist

Definition

Sei $\mathcal{L}$ eine Menge von Sprachen über dem Alphabet Σ, dann bezeichnen wir mit $\mathcal{L}^{\text{co}}$ die Menge der Komplementsprachen aus $\mathcal{L}$. Also

$$\mathcal{L}^{\text{co}} := \{\bar{L} \mid L \in \mathcal{L}\}$$

mit $\bar{L} = \Sigma^* \setminus L$.

Definition

Sei $\mathcal{L}$ eine Menge von Sprachen über dem Alphabet Σ, dann bezeichnen wir mit $\mathcal{L}^{\text{co}}$ die Menge der Komplementsprachen aus $\mathcal{L}$. Also

$$\mathcal{L}^{\text{co}} := \{\bar{L} \mid L \in \mathcal{L}\}$$

mit $\bar{L} = \Sigma^* \setminus L$.

- Achtung: im Allgemeinen gilt $\mathcal{L}^{\text{co}} \neq \bar{\mathcal{L}}$

Definition

Sei $\mathcal{L}$ eine Menge von Sprachen über dem Alphabet Σ, dann bezeichnen wir mit $\mathcal{L}^{\text{co}}$ die Menge der Komplementsprachen aus $\mathcal{L}$. Also

$$\mathcal{L}^{\text{co}} := \{\bar{L} \mid L \in \mathcal{L}\}$$

mit $\bar{L} = \Sigma^* \setminus L$.

- Achtung: im Allgemeinen gilt $\mathcal{L}^{\text{co}} \neq \bar{\mathcal{L}}$
- Beispiel: die co-regulären Sprachen REG^{co}

Definition

Sei $\mathcal{L}$ eine Menge von Sprachen über dem Alphabet Σ, dann bezeichnen wir mit $\mathcal{L}^{\text{co}}$ die Menge der Komplementsprachen aus $\mathcal{L}$. Also

$$\mathcal{L}^{\text{co}} := \{\bar{L} \mid L \in \mathcal{L}\}$$

mit $\bar{L} = \Sigma^* \setminus L$.

- Achtung: im Allgemeinen gilt $\mathcal{L}^{\text{co}} \neq \bar{\mathcal{L}}$
- Beispiel: die co-regulären Sprachen $\text{REG}^{\text{co}} \to \text{REG}^{\text{co}} = \text{REG}$

Definition

Sei $\mathcal{L}$ eine Menge von Sprachen über dem Alphabet Σ, dann bezeichnen wir mit $\mathcal{L}^{\text{co}}$ die Menge der Komplementsprachen aus $\mathcal{L}$. Also

$$\mathcal{L}^{\text{co}} := \{\bar{L} \mid L \in \mathcal{L}\}$$

mit $\bar{L} = \Sigma^* \setminus L$.

- Achtung: im Allgemeinen gilt $\mathcal{L}^{\text{co}} \neq \bar{\mathcal{L}}$
- Beispiel: die co-regulären Sprachen $\text{REG}^{\text{co}} \to \text{REG}^{\text{co}} = \text{REG}$
- $\mathbb{A}^{\text{co}}$ die Menge der co-aufzählbaren Sprachen

Definition

Sei $\mathcal{L}$ eine Menge von Sprachen über dem Alphabet Σ, dann bezeichnen wir mit $\mathcal{L}^{\text{co}}$ die Menge der Komplementsprachen aus $\mathcal{L}$. Also

$$\mathcal{L}^{\text{co}} := \{\bar{L} \mid L \in \mathcal{L}\}$$

mit $\bar{L} = \Sigma^* \setminus L$.

- Achtung: im Allgemeinen gilt $\mathcal{L}^{\text{co}} \neq \bar{\mathcal{L}}$
- Beispiel: die co-regulären Sprachen $\text{REG}^{\text{co}} \to \text{REG}^{\text{co}} = \text{REG}$
- $\mathbb{A}^{\text{co}}$ die Menge der co-aufzählbaren Sprachen

Satz 21

$$\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\text{co}}$$

Definition

Sei $\mathcal{L}$ eine Menge von Sprachen über dem Alphabet Σ, dann bezeichnen wir mit $\mathcal{L}^{\text{co}}$ die Menge der Komplementsprachen aus $\mathcal{L}$. Also

$$\mathcal{L}^{\text{co}} := \{\bar{L} \mid L \in \mathcal{L}\}$$

mit $\bar{L} = \Sigma^* \setminus L$.

- Achtung: im Allgemeinen gilt $\mathcal{L}^{\text{co}} \neq \bar{\mathcal{L}}$
- Beispiel: die co-regulären Sprachen $\text{REG}^{\text{co}} \to \text{REG}^{\text{co}} = \text{REG}$
- $\mathbb{A}^{\text{co}}$ die Menge der co-aufzählbaren Sprachen

Satz 21

$$\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\text{co}}$$

Vorüberlegung zum Beweis: $\mathbb{E}$ ist abgeschlossen unter Komplement

Definition

Sei $\mathcal{L}$ eine Menge von Sprachen über dem Alphabet Σ, dann bezeichnen wir mit $\mathcal{L}^{\text{co}}$ die Menge der Komplementsprachen aus $\mathcal{L}$. Also

$$\mathcal{L}^{\text{co}} := \{\bar{L} \mid L \in \mathcal{L}\}$$

mit $\bar{L} = \Sigma^* \setminus L$.

- Achtung: im Allgemeinen gilt $\mathcal{L}^{\text{co}} \neq \bar{\mathcal{L}}$
- Beispiel: die co-regulären Sprachen $\text{REG}^{\text{co}} \to \text{REG}^{\text{co}} = \text{REG}$
- $\mathbb{A}^{\text{co}}$ die Menge der co-aufzählbaren Sprachen

Satz 21

$$\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\text{co}}$$

Vorüberlegung zum Beweis: $\mathbb{E}$ ist abgeschlossen unter Komplement

- $L \in \mathbb{E} \Rightarrow$ es gibt einen Entscheider M_L für L

Definition

Sei $\mathcal{L}$ eine Menge von Sprachen über dem Alphabet Σ, dann bezeichnen wir mit $\mathcal{L}^{\text{co}}$ die Menge der Komplementsprachen aus $\mathcal{L}$. Also

$$\mathcal{L}^{\text{co}} := \{\bar{L} \mid L \in \mathcal{L}\}$$

mit $\bar{L} = \Sigma^* \setminus L$.

- Achtung: im Allgemeinen gilt $\mathcal{L}^{\text{co}} \neq \bar{\mathcal{L}}$
- Beispiel: die co-regulären Sprachen $\text{REG}^{\text{co}} \rightarrow \text{REG}^{\text{co}} = \text{REG}$
- $\mathbb{A}^{\text{co}}$ die Menge der co-aufzählbaren Sprachen

Satz 21

$$\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\text{co}}$$

Vorüberlegung zum Beweis: $\mathbb{E}$ ist abgeschlossen unter Komplement

- $L \in \mathbb{E} \Rightarrow$ es gibt einen Entscheider M_L für L
- Tausch von q_A und q_V gibt Entscheider für $\bar{L} \rightarrow \bar{L} \in \mathbb{E}$

Beweis $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\mathbf{co}}$:

Beweis $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\text{co}}$:

1. Teil $L \in \mathbb{E} \Rightarrow L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$

Beweis $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\textbf{co}}$:

1. Teil $L \in \mathbb{E} \Rightarrow L \in \mathbb{A} \cap \mathbb{A}^{\textbf{co}}$

- Wenn $L \in \mathbb{E}$, dann erkennt der Entscheider zu L die Sprache L, also $L \in \mathbb{A}$.

Beweis $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\text{co}}$:

1. Teil $L \in \mathbb{E} \Rightarrow L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$

- Wenn $L \in \mathbb{E}$, dann erkennt der Entscheider zu L die Sprache L, also $L \in \mathbb{A}$.
- Wenn $L \in \mathbb{E}$, dann $\bar{L} \in \mathbb{E}$ und der Entscheider zu $\bar{L}$ ekennt $\bar{L}$, also $\bar{L} \in \mathbb{A}$ und somit $L \in \mathbb{A}^{\text{co}}$.

Beweis $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\mathbf{co}}$:

1. Teil $L \in \mathbb{E} \Rightarrow L \in \mathbb{A} \cap \mathbb{A}^{\mathbf{co}}$

- Wenn $L \in \mathbb{E}$, dann erkennt der Entscheider zu L die Sprache L, also $L \in \mathbb{A}$.
- Wenn $L \in \mathbb{E}$, dann $\bar{L} \in \mathbb{E}$ und der Entscheider zu $\bar{L}$ ekennt $\bar{L}$, also $\bar{L} \in \mathbb{A}$ und somit $L \in \mathbb{A}^{\text{co}}$.
- $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$

Beweis $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\mathbf{co}}$:

1. **Teil** $L \in \mathbb{E} \Rightarrow L \in \mathbb{A} \cap \mathbb{A}^{\mathbf{co}}$
   - Wenn $L \in \mathbb{E}$, dann erkennt der Entscheider zu L die Sprache L, also $L \in \mathbb{A}$.
   - Wenn $L \in \mathbb{E}$, dann $\bar{L} \in \mathbb{E}$ und der Entscheider zu $\bar{L}$ ekennt $\bar{L}$, also $\bar{L} \in \mathbb{A}$ und somit $L \in \mathbb{A}^{\text{co}}$.
   - $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$
2. **Teil** $L \in \mathbb{A} \cap \mathbb{A}^{\mathbf{co}} \Rightarrow L \in \mathbb{E}$

Beweis $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\textbf{co}}$:

1. **Teil** $L \in \mathbb{E} \Rightarrow L \in \mathbb{A} \cap \mathbb{A}^{\textbf{co}}$
   - Wenn $L \in \mathbb{E}$, dann erkennt der Entscheider zu L die Sprache L, also $L \in \mathbb{A}$.
   - Wenn $L \in \mathbb{E}$, dann $\bar{L} \in \mathbb{E}$ und der Entscheider zu $\bar{L}$ ekennt $\bar{L}$, also $\bar{L} \in \mathbb{A}$ und somit $L \in \mathbb{A}^{\text{co}}$.
   - $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$
2. **Teil** $L \in \mathbb{A} \cap \mathbb{A}^{\textbf{co}} \Rightarrow L \in \mathbb{E}$
   - Sei M eine TM die L erkennt und $\bar{M}$ eine Turingmaschine die $\bar{L}$ erkennt

Beweis $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\text{co}}$:

1. **Teil** $L \in \mathbb{E} \Rightarrow L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$
   - Wenn $L \in \mathbb{E}$, dann erkennt der Entscheider zu L die Sprache L, also $L \in \mathbb{A}$.
   - Wenn $L \in \mathbb{E}$, dann $\bar{L} \in \mathbb{E}$ und der Entscheider zu $\bar{L}$ ekennt $\bar{L}$, also $\bar{L} \in \mathbb{A}$ und somit $L \in \mathbb{A}^{\text{co}}$.
   - $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$
2. **Teil** $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}} \Rightarrow L \in \mathbb{E}$
   - Sei M eine TM die L erkennt und $\bar{M}$ eine Turingmaschine die $\bar{L}$ erkennt
   - Baue neue TM M', die abwechselnd bei Eingabe w, jeweils $M(w)$ und $\bar{M}(w)$ einen Schritt simuliert

Beweis $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\text{co}}$:

1. Teil $L \in \mathbb{E} \Rightarrow L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$

- Wenn $L \in \mathbb{E}$, dann erkennt der Entscheider zu L die Sprache L, also $L \in \mathbb{A}$.
- Wenn $L \in \mathbb{E}$, dann $\bar{L} \in \mathbb{E}$ und der Entscheider zu $\bar{L}$ ekennt $\bar{L}$, also $\bar{L} \in \mathbb{A}$ und somit $L \in \mathbb{A}^{\text{co}}$.
- $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$

2. Teil $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}} \Rightarrow L \in \mathbb{E}$

- Sei M eine TM die L erkennt und $\bar{M}$ eine Turingmaschine die $\bar{L}$ erkennt
- Baue neue TM M', die abwechselnd bei Eingabe w, jeweils $M(w)$ und $\bar{M}(w)$ einen Schritt simuliert
- Akzeptiere, wenn $M(w)$ akzeptiert.

Beweis $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\text{co}}$:

1. **Teil** $L \in \mathbb{E} \Rightarrow L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$
   - Wenn $L \in \mathbb{E}$, dann erkennt der Entscheider zu L die Sprache L, also $L \in \mathbb{A}$.
   - Wenn $L \in \mathbb{E}$, dann $\bar{L} \in \mathbb{E}$ und der Entscheider zu $\bar{L}$ ekennt $\bar{L}$, also $\bar{L} \in \mathbb{A}$ und somit $L \in \mathbb{A}^{\text{co}}$.
   - $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$
2. **Teil** $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}} \Rightarrow L \in \mathbb{E}$
   - Sei M eine TM die L erkennt und $\bar{M}$ eine Turingmaschine die $\bar{L}$ erkennt
   - Baue neue TM M', die abwechselnd bei Eingabe w, jeweils $M(w)$ und $\bar{M}(w)$ einen Schritt simuliert
   - Akzeptiere, wenn $M(w)$ akzeptiert.
   - Verwerfe, wenn $\bar{M}(w)$ akzeptiert.

Beweis $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\text{co}}$:

1. Teil $L \in \mathbb{E} \Rightarrow L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$

- Wenn $L \in \mathbb{E}$, dann erkennt der Entscheider zu L die Sprache L, also $L \in \mathbb{A}$.
- Wenn $L \in \mathbb{E}$, dann $\bar{L} \in \mathbb{E}$ und der Entscheider zu $\bar{L}$ ekennt $\bar{L}$, also $\bar{L} \in \mathbb{A}$ und somit $L \in \mathbb{A}^{\text{co}}$.
- $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$

2. Teil $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}} \Rightarrow L \in \mathbb{E}$

- Sei M eine TM die L erkennt und $\bar{M}$ eine Turingmaschine die $\bar{L}$ erkennt
- Baue neue TM M', die abwechselnd bei Eingabe w, jeweils $M(w)$ und $\bar{M}(w)$ einen Schritt simuliert
- Akzeptiere, wenn $M(w)$ akzeptiert.
- Verwerfe, wenn $\bar{M}(w)$ akzeptiert.
- $M(w)$ oder $\bar{M}(w)$ muss akzeptieren $\rightarrow$ M' ist Entscheider

Beweis $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\text{co}}$:

1. **Teil** $L \in \mathbb{E} \Rightarrow L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$
   - Wenn $L \in \mathbb{E}$, dann erkennt der Entscheider zu L die Sprache L, also $L \in \mathbb{A}$.
   - Wenn $L \in \mathbb{E}$, dann $\bar{L} \in \mathbb{E}$ und der Entscheider zu $\bar{L}$ ekennt $\bar{L}$, also $\bar{L} \in \mathbb{A}$ und somit $L \in \mathbb{A}^{\text{co}}$.
   - $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}}$
2. **Teil** $L \in \mathbb{A} \cap \mathbb{A}^{\text{co}} \Rightarrow L \in \mathbb{E}$
   - Sei M eine TM die L erkennt und $\bar{M}$ eine Turingmaschine die $\bar{L}$ erkennt
   - Baue neue TM M', die abwechselnd bei Eingabe w, jeweils $M(w)$ und $\bar{M}(w)$ einen Schritt simuliert
   - Akzeptiere, wenn $M(w)$ akzeptiert.
   - Verwerfe, wenn $\bar{M}(w)$ akzeptiert.
   - $M(w)$ oder $\bar{M}(w)$ muss akzeptieren $\to$ M' ist Entscheider
   - $L \in \mathbb{E}$

$\square$

Satz 22

$$\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$$

$(A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\} \notin \mathbb{E})$

Satz 22

$$\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$$

$(A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w)$ akzeptiert$\} \notin \mathbb{E})$

Beweis

Satz 22

$$\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$$

$(A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w)$ akzeptiert$\} \notin \mathbb{E})$

Beweis

Die Sprache A_{TM} ist erkennbar ($A \in \mathbb{A}$), durch folgende TM:

Satz 22

$$\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$$

$(A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\} \notin \mathbb{E})$

Beweis

Die Sprache A_{TM} ist erkennbar ($A \in \mathbb{A}$), durch folgende TM:

- bei Eingabe $\langle M, w\rangle$ simuliere $M(w)$

Satz 22

$$\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$$

$(A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\} \notin \mathbb{E})$

Beweis

Die Sprache A_{TM} ist erkennbar ($A \in \mathbb{A}$), durch folgende TM:

- bei Eingabe $\langle M, w\rangle$ simuliere $M(w)$
- stoppt die Simulation, gib das Ergebnis der Simulation weiter

Satz 22

$$\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$$

$(A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\} \notin \mathbb{E})$

Beweis

Die Sprache A_{TM} ist erkennbar ($A \in \mathbb{A}$), durch folgende TM:

- bei Eingabe $\langle M, w\rangle$ simuliere $M(w)$
- stoppt die Simulation, gib das Ergebnis der Simulation weiter

$\rightarrow$ alle Eingaben $\langle M, w\rangle \in A_{\mathsf{TM}}$ werden akzeptiert

Satz 22

$$\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$$

$(A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\} \notin \mathbb{E})$

Beweis

Die Sprache A_{TM} ist erkennbar ($A \in \mathbb{A}$), durch folgende TM: **(UTM)**

- bei Eingabe $\langle M, w\rangle$ simuliere $M(w)$
- stoppt die Simulation, gib das Ergebnis der Simulation weiter

$\rightarrow$ alle Eingaben $\langle M, w\rangle \in A_{\mathsf{TM}}$ werden akzeptiert

Satz 22

$$\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$$

$(A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\} \notin \mathbb{E})$

Beweis

Die Sprache A_{TM} ist erkennbar ($A \in \mathbb{A}$), durch folgende TM: **(UTM)**

- bei Eingabe $\langle M, w\rangle$ simuliere $M(w)$
- stoppt die Simulation, gib das Ergebnis der Simulation weiter

$\rightarrow$ alle Eingaben $\langle M, w\rangle \in A_{\mathsf{TM}}$ werden akzeptiert

Nach Satz 20 ist jedoch A_{TM} nicht entscheidbar

Satz 22

$$\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$$

$(A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\} \notin \mathbb{E})$

Beweis

Die Sprache A_{TM} ist erkennbar ($A \in \mathbb{A}$), durch folgende TM: **(UTM)**

- bei Eingabe $\langle M, w\rangle$ simuliere $M(w)$
- stoppt die Simulation, gib das Ergebnis der Simulation weiter

$\rightarrow$ alle Eingaben $\langle M, w\rangle \in A_{\mathsf{TM}}$ werden akzeptiert

Nach Satz 20 ist jedoch A_{TM} nicht entscheidbar

Da nach Satz 21: $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\mathsf{co}}$, folgt dass $A_{\mathsf{TM}} \notin \mathbb{A}^{\mathsf{co}}$.

Satz 22

$$\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$$

$(A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\} \notin \mathbb{E})$

Beweis

Die Sprache A_{TM} ist erkennbar ($A \in \mathbb{A}$), durch folgende TM: **(UTM)**

- bei Eingabe $\langle M, w\rangle$ simuliere $M(w)$
- stoppt die Simulation, gib das Ergebnis der Simulation weiter

$\rightarrow$ alle Eingaben $\langle M, w\rangle \in A_{\mathsf{TM}}$ werden akzeptiert

Nach Satz 20 ist jedoch A_{TM} nicht entscheidbar

Da nach Satz 21: $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\mathsf{co}}$, folgt dass $A_{\mathsf{TM}} \notin \mathbb{A}^{\mathsf{co}}$.

Also: $\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$. $\square$

Satz 22

$$\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$$

$(A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\} \notin \mathbb{E})$

Beweis

Die Sprache A_{TM} ist erkennbar ($A \in \mathbb{A}$), durch folgende TM: **(UTM)**

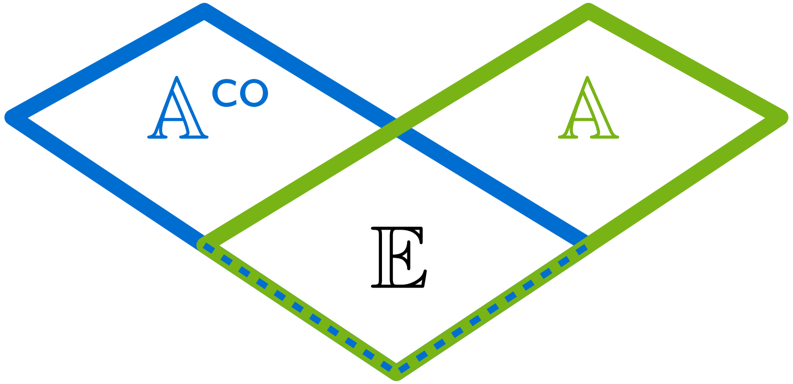
- bei Eingabe $\langle M, w\rangle$ simuliere $M(w)$
- stoppt die Simulation, gib das Ergebnis der Simulation weiter

$\rightarrow$ alle Eingaben $\langle M, w\rangle \in A_{\mathsf{TM}}$ werden akzeptiert

Nach Satz 20 ist jedoch A_{TM} nicht entscheidbar

Da nach Satz 21: $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\mathsf{co}}$, folgt dass $A_{\mathsf{TM}} \notin \mathbb{A}^{\mathsf{co}}$.

Also: $\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$. $\square$



Satz 22

$$\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$$

$(A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M(w) \text{ akzeptiert}\} \notin \mathbb{E})$

Beweis

Die Sprache A_{TM} ist erkennbar ($A \in \mathbb{A}$), durch folgende TM: **(UTM)**

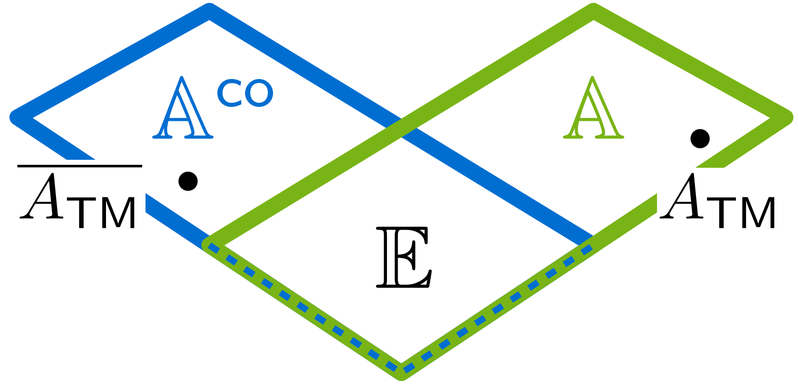
- bei Eingabe $\langle M, w\rangle$ simuliere $M(w)$
- stoppt die Simulation, gib das Ergebnis der Simulation weiter

$\rightarrow$ alle Eingaben $\langle M, w\rangle \in A_{\mathsf{TM}}$ werden akzeptiert

Nach Satz 20 ist jedoch A_{TM} nicht entscheidbar

Da nach Satz 21: $\mathbb{E} = \mathbb{A} \cap \mathbb{A}^{\mathsf{co}}$, folgt dass $A_{\mathsf{TM}} \notin \mathbb{A}^{\mathsf{co}}$.

Also: $\overline{A_{\mathsf{TM}}} \notin \mathbb{A}$. □



Reduktionen

Reduktionen

- wir haben bereits die Nichtentscheidbarkeit einer Sprache durch Diagonalisierung gezeigt

Reduktionen

- wir haben bereits die Nichtentscheidbarkeit einer Sprache durch Diagonalisierung gezeigt
- Idee lässt sich abwandeln, z.B. für HALT $:= \{\langle M, w\rangle \mid M(w)\downarrow\}$

Reduktionen

- wir haben bereits die Nichtentscheidbarkeit einer Sprache durch Diagonalisierung gezeigt
- Idee lässt sich abwandeln, z.B. für $\text{HALT} := \{\langle M, w\rangle \mid M(w)\downarrow\}$

	1	2	3	4
T_1	$\uparrow$	$\downarrow$	$\downarrow$	$\downarrow$
T_2	$\downarrow$	$\downarrow$	$\downarrow$	$\uparrow$
T_3	$\downarrow$	$\uparrow$	$\downarrow$	$\downarrow$
T_4	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$

Reduktionen

- wir haben bereits die Nichtentscheidbarkeit einer Sprache durch Diagonalisierung gezeigt
- Idee lässt sich abwandeln, z.B. für HALT $:= \{\langle M, w\rangle \mid M(w)\downarrow\}$

	1	2	3	4
T_1	$\uparrow$	$\downarrow$	$\downarrow$	$\downarrow$
T_2	$\downarrow$	$\downarrow$	$\downarrow$	$\uparrow$
T_3	$\downarrow$	$\uparrow$	$\downarrow$	$\downarrow$
T_4	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$

Falls HALT $\in \mathbb{E}$ kann ich eine Diagonalisierungsmaschine D bauen

1. entscheide ob Eingabe $\langle M, \langle M\rangle\rangle$ aus HALT
2. Wenn ja, gehe in Endlosschleife, wenn nein stoppe

Reduktionen

- wir haben bereits die Nichtentscheidbarkeit einer Sprache durch Diagonalisierung gezeigt
- Idee lässt sich abwandeln, z.B. für HALT $:= \{\langle M, w\rangle \mid M(w)\downarrow\}$

	1	2	3	4
T_1	$\uparrow$	$\downarrow$	$\downarrow$	$\downarrow$
T_2	$\downarrow$	$\downarrow$	$\downarrow$	$\uparrow$
T_3	$\downarrow$	$\uparrow$	$\downarrow$	$\downarrow$
T_4	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
D	$\downarrow$	$\uparrow$	$\uparrow$	$\uparrow$

Falls HALT $\in \mathbb{E}$ kann ich eine Diagonalisierungsmaschine D bauen

1. entscheide ob Eingabe $\langle M, \langle M\rangle\rangle$ aus HALT
2. Wenn ja, gehe in Endlosschleife, wenn nein stoppe

Reduktionen

- wir haben bereits die Nichtentscheidbarkeit einer Sprache durch Diagonalisierung gezeigt
- Idee lässt sich abwandeln, z.B. für HALT $:= \{\langle M, w\rangle \mid M(w)\downarrow\}$

	1	2	3	4
T_1	$\uparrow$	$\downarrow$	$\downarrow$	$\downarrow$
T_2	$\downarrow$	$\downarrow$	$\downarrow$	$\uparrow$
T_3	$\downarrow$	$\uparrow$	$\downarrow$	$\downarrow$
T_4	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
D	$\downarrow$	$\uparrow$	$\uparrow$	$\uparrow$

Falls HALT $\in \mathbb{E}$ kann ich eine Diagonalisierungsmaschine D bauen

1. entscheide ob Eingabe $\langle M, \langle M\rangle\rangle$ aus HALT
2. Wenn ja, gehe in Endlosschleife, wenn nein stoppe

- D kann nicht existieren $\rightarrow$ HALT ist nicht entscheidbar

Reduktionen

- wir haben bereits die Nichtentscheidbarkeit einer Sprache durch Diagonalisierung gezeigt
- Idee lässt sich abwandeln, z.B. für HALT $:= \{\langle M, w\rangle \mid M(w)\downarrow\}$

	1	2	3	4
T_1	$\uparrow$	$\downarrow$	$\downarrow$	$\downarrow$
T_2	$\downarrow$	$\downarrow$	$\downarrow$	$\uparrow$
T_3	$\downarrow$	$\uparrow$	$\downarrow$	$\downarrow$
T_4	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
D	$\downarrow$	$\uparrow$	$\uparrow$	$\uparrow$

Falls HALT $\in \mathbb{E}$ kann ich eine Diagonalisierungsmaschine D bauen

1. entscheide ob Eingabe $\langle M, \langle M\rangle\rangle$ aus HALT
2. Wenn ja, gehe in Endlosschleife, wenn nein stoppe

- D kann nicht existieren $\rightarrow$ HALT ist nicht entscheidbar
- gleiche Konstruktion zeigt, dass SHALT $= \{x \mid M_x(x)\downarrow\} \notin \mathbb{E}$

Reductionen

- wir haben bereits die Nichtentscheidbarkeit einer Sprache durch Diagonalisierung gezeigt
- Idee lässt sich abwandeln, z.B. für HALT $:= \{\langle M, w\rangle \mid M(w)\downarrow\}$

	1	2	3	4
T_1	$\uparrow$	$\downarrow$	$\downarrow$	$\downarrow$
T_2	$\downarrow$	$\downarrow$	$\downarrow$	$\uparrow$
T_3	$\downarrow$	$\uparrow$	$\downarrow$	$\downarrow$
T_4	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
D	$\downarrow$	$\uparrow$	$\uparrow$	$\uparrow$

Falls HALT $\in \mathbb{E}$ kann ich eine Diagonalisierungsmaschine D bauen

1. entscheide ob Eingabe $\langle M, \langle M\rangle\rangle$ aus HALT
2. Wenn ja, gehe in Endlosschleife, wenn nein stoppe

- D kann nicht existieren $\rightarrow$ HALT ist nicht entscheidbar
- gleiche Konstruktion zeigt, dass SHALT $= \{x \mid M_x(x)\downarrow\} \notin \mathbb{E}$

  (denn für die Diagonalisierung muss ich nur wissen ob $M_x(x)\uparrow$ oder $M_x(x)\downarrow$)

Reduktionen

- wir haben bereits die Nichtentscheidbarkeit einer Sprache durch Diagonalisierung gezeigt
- Idee lässt sich abwandeln, z.B. für HALT $:= \{\langle M, w\rangle \mid M(w)\downarrow\}$

	1	2	3	4
T_1	$\uparrow$	$\downarrow$	$\downarrow$	$\downarrow$
T_2	$\downarrow$	$\downarrow$	$\downarrow$	$\uparrow$
T_3	$\downarrow$	$\uparrow$	$\downarrow$	$\downarrow$
T_4	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
D	$\downarrow$	$\uparrow$	$\uparrow$	$\uparrow$

Falls HALT $\in \mathbb{E}$ kann ich eine Diagonalisierungsmaschine D bauen

1. entscheide ob Eingabe $\langle M, \langle M\rangle\rangle$ aus HALT
2. Wenn ja, gehe in Endlosschleife, wenn nein stoppe

- D kann nicht existieren $\to$ HALT ist nicht entscheidbar
- gleiche Konstruktion zeigt, dass SHALT $= \{x \mid M_x(x)\downarrow\} \notin \mathbb{E}$

  (denn für die Diagonalisierung muss ich nur wissen ob $M_x(x)\uparrow$ oder $M_x(x)\downarrow$)
- **Gesucht:** Methode zum Transfer dieser Ergebnisse