

Entscheidungsprobleme

- übliche Formulierung
 - gegeben:** Eingabe x aus einer Grundmenge M
 - Frage:** Hat x eine bestimmte Eigenschaft P ?
- Beispiel:
 - gegeben:** $n \in \mathbb{N}$
 - Frage:** Ist n eine Primzahl?
- **Formalisierung:**
Grundmenge ist die Menge aller Wörter über einem Alphabet Σ .
Menge aller Eingaben mit Antwort "JA" ist Sprache $L \subseteq \Sigma^*$

Definitionen Entscheidbarkeit und Semi-Entscheidbarkeit

Definition: Eine Menge $A \subseteq \Sigma^*$ heißt **entscheidbar**, falls die charakteristische Funktion von A , nämlich $\chi_A: \Sigma^* \rightarrow \{0, 1\}$, berechenbar ist. Hierbei ist für alle $w \in \Sigma^*$:

$$\chi_A(w) = \begin{cases} 1 & \text{falls } w \in A, \\ 0 & \text{falls } w \notin A, \end{cases}$$

Definition: Eine Menge $A \subseteq \Sigma^*$ heißt **semi-entscheidbar**, falls die “halbe” charakteristische Funktion von A , nämlich $\chi'_A: \Sigma^* \rightarrow \{0, 1\}$, berechenbar ist. Hierbei ist für alle $w \in \Sigma^*$:

$$\chi'_A(w) = \begin{cases} 1 & \text{falls } w \in A, \\ \text{nicht definiert} & \text{falls } w \notin A, \end{cases}$$

Beziehung zwischen Entscheidbarkeit und Semi-Entscheidbarkeit

Satz. Eine Menge $A \subseteq \Sigma^*$ ist entscheidbar genau dann, wenn sowohl die Menge A als auch ihr Komplement $\bar{A} = \Sigma^* \setminus A$ semi-entscheidbar sind.

Beweis.

- Aus Entscheidbarkeit von A folgt sofort Semi-Entscheidbarkeit von A und \bar{A} .
- Sind A und \bar{A} jeweils semi-entscheidbar,
 - so lasse für eine Eingabe w gleichzeitig Algorithmen M_A und $M_{\bar{A}}$ zur Berechnung von χ'_A und $\chi'_{\bar{A}}$ laufen, bis einer stoppt.
Wegen der Semi-Entscheidbarkeit von A und \bar{A} muss einer der Algorithmen stoppen!
 - Stoppt M_A : Ausgabe 1; stoppt $M_{\bar{A}}$: Ausgabe 0.

Rekursiv aufzählbare Mengen

Definition. Eine Menge $A \subseteq \Sigma^*$ heißt **rekursiv aufzählbar**, falls $A = \emptyset$ oder falls es eine totale und **berechenbare** Funktion $f: \mathbb{N} \rightarrow \Sigma^*$ gibt, so dass

$$A = \{f(0), f(1), f(2), \dots\}$$

gilt. Sprechweise: f zählt A auf.

Man beachte: $f(i) = f(j)$ für $i \neq j$ ist zulässig.

Satz. Eine Menge ist genau dann **rekursiv aufzählbar**, wenn sie **semi-entscheidbar** ist.

Folgerung

Sei $A \subseteq \Sigma^*$ eine Menge. Dann sind folgende Aussagen äquivalent.

- (i) A ist rekursiv aufzählbar.
- (ii) A ist semi-entscheidbar.
- (iii) χ'_A ist berechenbar.
- (iv) A ist Definitionsbereich einer berechenbaren Funktion.
- (v) A ist Wertebereich einer berechenbaren Funktion.

Nicht-entscheidbare Probleme

1. Probleme mit Turingmaschinen als Eingaben (**Halteproblem**)
 - Turingmaschinen werden durch Wörter codiert.
 - Anwendung der **Diagonalisierung**
2. **Reduktion** als Beweistechnik für Nicht-Entscheidbarkeit
3. Weitere nicht-entscheidbare Probleme

Codierung von Turing-Maschinen

Gegeben: TM $M = (Z, \{0, 1\}, \Gamma, \delta, z_0, \square, E)$

Ziel: Codierung von M durch ein Wort $w \in \{0, 1\}^*$

$Z = \{z_0, z_1, \dots, z_k\}$, $\Gamma = \{a_0, a_1, \dots, a_k\}$ wobei $a_0 = 0, a_1 = 1, a_2 = \square$,
 $E = \{z_{i_1}, \dots, z_{i_\ell}\}$.

Codierung von Z, Γ, E durch Wort $u \in \{0, 1, \#\}^*$:

$u = \#\#\text{bin}(k)\#\#\text{bin}(n)\#\#\text{bin}(i_1)\#\text{bin}(i_2) \cdots \text{bin}(i_\ell)$

Jeder δ -Regel der Form $\delta(z_i, a_j) = (z_{i'}, a_{j'}, y)$ entspricht das Wort

$$w_{i,j} = \#\#\text{bin}(i)\#\text{bin}(j)\#\text{bin}(i')\#\text{bin}(j')\#\text{bin}(y),$$

wobei $\text{bin}(L) = 00, \text{bin}(R) = 01, \text{bin}(N) = 10$

Codierung von δ durch Wort v , das durch Konkatenation der Wörter $w_{i,j}$ entsteht.

Codierung von M durch $uv \in \{0, 1, \#\}^*$.

Codierung von Turing-Maschinen – Fortsetzung

Jedes Wort über $\{0, 1, \#\}^*$ kann durch ein Wort über $\{0, 1\}$ codiert werden, indem man folgende Codierung vornimmt:

$$0 \mapsto 00, \quad 1 \mapsto 01, \quad 2 \mapsto 10, \quad \# \mapsto 11.$$

Nicht jedes Wort in $\{0, 1\}^*$ ist die Codierung einer Turingmaschine. Sei aber M_0 irgendeine beliebige feste Turingmaschine, dann können wir *für jedes* $w \in \{0, 1\}^*$ festlegen, dass M_w eine bestimmte Turingmaschine bezeichnet, nämlich

$$M(w) = M_w = \begin{cases} M & \text{falls } w \text{ Codewort von } M \text{ ist,} \\ M_0 & \text{sonst.} \end{cases}$$

Spezielles Halteproblem für Turingmaschinen

Definition. Das **spezielle Halteproblem** für Turingmaschinen ist die Menge

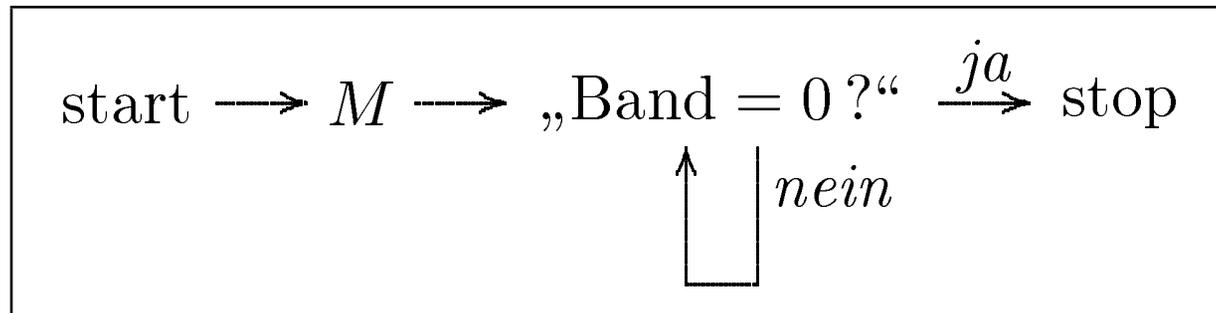
$$K = \{w \in \{0, 1\}^* \mid M_w \text{ angesetzt auf } w \text{ hält}\}.$$

Satz.

Das spezielle Halteproblem für Turingmaschinen (K) ist nicht entscheidbar.

Unentscheidbarkeit des speziellen Halteproblems – Beweis I

Angenommen, K wäre entscheidbar. Dann wäre χ_K berechenbar mittels einer Turingmaschine M . Diese Maschine M könnte nun leicht zu einer Turingmaschine M' umgebaut werden, die durch folgende Abbildung definiert ist.



Das heißt, M' stoppt genau dann, wenn M den Wert 0 ausgeben würde. Falls M den Wert 1 ausgibt, gerät M' in eine Endlosschleife.

Unentscheidbarkeit des speziellen Halteproblems – Beweis II

Sei w' ein Codewort der Maschine M' . Nun gilt

M' angesetzt auf w' hält

- $\Leftrightarrow M$ angesetzt auf w' gibt 0 aus (wegen Definition von M'),
- $\Leftrightarrow \chi_K(w') = 0$ (da M die Menge K entscheidet),
- $\Leftrightarrow w' \notin K$ (wegen Definition von χ_K),
- $\Leftrightarrow M_{w'}$ angesetzt auf w' hält nicht (wegen Definition von K),
- $\Leftrightarrow M'$ angesetzt auf w' hält nicht (da w' Code von M' ist).

Dieser Widerspruch beweist, dass die Eingangsannahme falsch war, also ist K nicht entscheidbar.

Reduzierbarkeit von Sprachen

Definition.

Eine Sprache $A \subseteq \Sigma^*$ heißt **reduzierbar** auf die Sprache $B \subseteq \Gamma^*$, Notation $A \leq B$, wenn es eine totale und berechenbare Funktion $f : \Sigma^* \rightarrow \Gamma^*$ gibt, sodass

für alle $w \in \Sigma^*$: $w \in A$ genau dann, wenn $f(w) \in B$.

Satz.

Es seien A und B Sprachen mit $A \leq B$.

Ist B entscheidbar, so ist auch A entscheidbar.

Ist A nicht entscheidbar, so ist auch B nicht entscheidbar.

Das Halteproblem für Turingmaschinen

Definition. Das (allgemeine) **Halteproblem** für Turingmaschinen ist die Menge

$$H = \{w\#x \mid M_w \text{ angesetzt auf } x \text{ hält}\}.$$

Satz. Das Halteproblem für Turingmaschinen (H) ist nicht entscheidbar.

Beweis: durch **Reduktion** des speziellen Halteproblems K auf H

für alle $w \in \{0, 1\}^*$: $w \in K$ genau dann, wenn $w\#w \in H$

Funktion $f : \{0, 1\}^* \rightarrow \{0, 1, \#\}^*$ mit $f(w) = w\#w$ ist berechenbar.

Da K nicht entscheidbar ist, ist auch H nicht entscheidbar.

Das 10. Hilbertsche ³ Problem

Definition. Das 10. Hilbertsche Problem ist definiert durch:

Gegeben: $n \in \mathbb{N}$, ein Polynom $p(x_1, \dots, x_n)$ in n Unbekannten,

Frage: Besitzt p ganzzahlige Nullstellen?

Satz. Das 10. Hilbertsche Problem ist nicht entscheidbar.

³David Hilbert, deutscher Mathematiker, 1862-1943

Das Postsche ⁴ Korrespondenzproblem

Definition. Das Postsche Korrespondenzproblem (PKP) ist definiert durch:

Gegeben: Alphabet A , $k \in \mathbb{N}$ sowie die Folge von Wortpaaren

$(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$ mit $x_i, y_i \in A^+$ für $1 \leq i \leq k$.

Frage: Gibt es eine Folge von Indizes i_1, i_2, \dots, i_n mit $i_j \in \{1, 2, \dots, k\}$ für $1 \leq j \leq n$, $n \in \mathbb{N}$, so dass $x_{i_1}x_{i_2} \dots x_{i_n} = y_{i_1}y_{i_2} \dots y_{i_n}$ gilt?

Satz. Das Postsche Korrespondenzproblem ist nicht entscheidbar.

⁴Emil Post, amerikanischer Mathematiker, 1897-1954

Beispiel für Postsches Korrespondenzproblem

Das Korrespondenzproblem

$$A = \{0, 1\}, \quad k = 3, \quad K = ((1, 101), (10, 00), (011, 11)),$$

also

$$\begin{array}{lll} x_1 = 1 & x_2 = 10 & x_3 = 011 \\ y_1 = 101 & y_2 = 00 & y_3 = 11 \end{array}$$

besitzt die Lösung $(1, 3, 2, 3)$, denn es gilt

$$x_1 x_3 x_2 x_3 = 101110011 = 101110011 = y_1 y_3 y_2 y_3.$$

Weiteres Beispiel für Postsches Korrespondenzproblem

Gegeben ist folgende Belegung des PKP:

$$\begin{array}{cccc} x_1 = 001 & x_2 = 01 & x_3 = 01 & x_4 = 10 \\ y_1 = 0 & y_2 = 011 & y_3 = 101 & y_4 = 001. \end{array}$$

Dieses Problem besitzt eine Lösung, aber die kürzeste Lösung besteht aus 66 Indizes, nämlich

2, 4, 3, 4, 4, 2, 1, 2, 4, 3, 4, 3, 4, 4, 3, 4, 4, 2, 1, 4, 4, 2, 1, 3, 4, 1, 1, 3, 4, 4, 4, 2, 1,
2, 1, 1, 1, 3, 4, 3, 4, 1, 1, 1, 4, 4, 2, 1, 4, 1, 1, 3, 4, 1, 1, 3, 1, 1, 3, 1, 2, 1, 4, 1, 1, 3

Semi-Entscheidbarkeit des PKP

Der **naive Algorithmus**, der bei gegebener Eingabe

$$(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$$

systematisch alle immer länger werdende Indexfolgen i_1, i_2, \dots, i_n daraufhin untersucht, ob sie eine Lösung darstellen und im positiven Fall stoppt, demonstriert, dass das **PKP semi-entscheidbar** ist. Bei Eingaben, die keine Lösung besitzen, stoppt das Verfahren allerdings nicht.