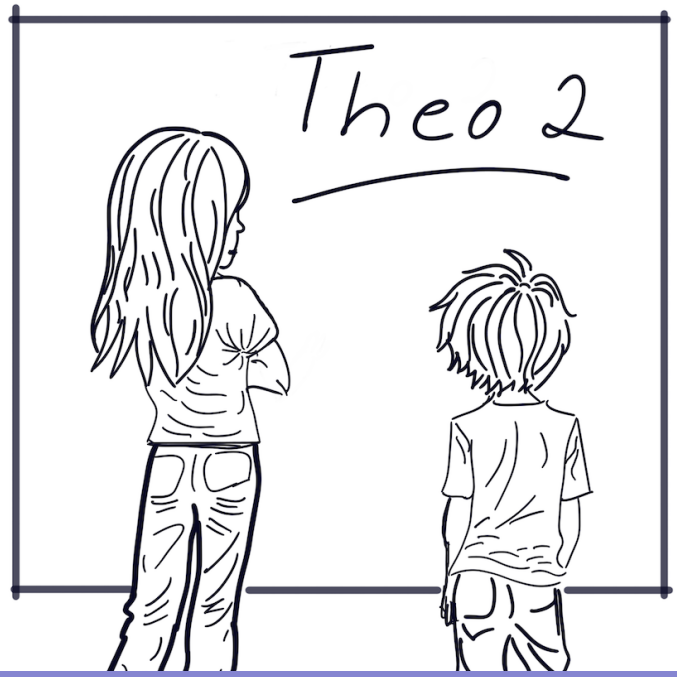
Theoretische Informatik 2

SoSe 2024

Berechenbarkeit und Komplexität

Prof. Dr. Sebastian Siebertz
AG Theoretische Informatik
MZH, Raum 3160
siebertz@uni-bremen.de

Universität Bremen



Church-Turing These

Satz

Die folgenden Klassen von Funktionen stimmen überein.

- Turing-berechenbare Funktionen
- WHILE-berechenbare Funktionen, C/Java/Python/...-berechenbare Funktionen
- μ-rekursive Funktionen, ...

Church-Turing These

Die Klasse der Turing-berechenbaren Funktionen stimmt mit der Klasse der intuitiv berechenbaren Funktionen überein.

Berechenbar heißt Turing-berechenbar.

Wiederholung Entscheidungsprobleme, entscheidbar, semi-entscheidbar

- Entscheidungsproblem, Sprache über Σ: $L \subseteq \Sigma^*$.
- L berechenbar, rekursiv, entscheidbar: es existiert Turingmaschine M, die auf jedem $w \in \Sigma^*$ terminiert und $L(M) = L$
  (d.h. M akzeptiert w wenn $w \in L$ und verwirft wenn $w \notin L$).
- L rekursiv aufzählbar, semi-entscheidbar: es existiert Turingmaschine M mit $L(M) = L$
  (d.h. M akzeptiert w wenn $w \in L$ und verwirft oder terminiert nicht wenn $w \notin L$).

Unentscheidbare Probleme

Satz

Es existieren unentscheidbare Probleme.

Beweis.

- Es gibt nur abzählbar viele Turingmaschinen (bis auf Umbenennung der Zustände und Symbole des Arbeitsalphabets), also auch nur abzählbar viele entscheidbare Sprachen.
  - Abzählbar: wir können alle Maschinen auflisten als $M_1, M_2, \ldots$
- Es gibt überabzählbar viele Sprachen.
  - Es gibt keine injektive Abbildung $f : \{L \ : \ L \text{ Sprache über } \Sigma\} \to \mathbb{N}$.

$\Rightarrow$ Es gibt eine Sprache, die von keiner Turingmaschine entschieden wird.

Unentscheidbare Probleme

- Ziel in dieser Vorlesung:
  Es gibt **wichtige** Sprachen, die unentscheidbar sind.
  - Hält eine Turingmaschine auf einer/jeder Eingabe?
  - Akzeptiert eine Turingmaschine ein gegebenes Wort?
  - ...
- Übersicht
  - Turingmaschinen können als Strings kodiert werden: Gödelisierung.
  - Turingmaschinen können ihre eigenen Kodierungen als Eingaben erhalten: Selbstreferenz.
  - Widerspruchsbeweis mit Diagonalisierung.
  - Wenn ein erstes konkretes unentscheidbares Problem bekannt ist: Reduktionen um für weitere Probleme zu zeigen, dass sie unentscheidbar sind.

Kodierung von Turingmaschinen

- Kodiere Turingmaschine $M = (Q, \Sigma, \Gamma, [, \sqcup, \delta, q_s, q_a, q_r)$ als String $\langle M \rangle$ über $\{0, 1\}$.
- Annahmen:
  - Jede TM mit n Zuständen benutzt genau die Zustände $q_1, \ldots, q_n$.
  - $\Sigma = \{0, 1\}$.
  - Jede TM mit Arbeitsalphabet der Größe m benutzt $\Gamma = \{0, 1, \ldots, m-1\}$.

Kodierung von Turingmaschinen

- Kodiere Turingmaschine $M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, q_s, q_a, q_r)$ als String $\langle M \rangle$ über $\{0, 1\}$.
- $\langle M \rangle$ beginnt mit

$$10^n 10^m 10^\ell 10^b 10^s 10^a 10^r$$

  - $n = |Q|$,
  - $m = |\Gamma|$,
  - ℓ und b: Indizes des linken Endmarkers und des Blanksymbols in der Menge $\Gamma = \{0, \ldots, m-1\}$,
  - s, a und r: Indizes des Start-, akzeptierenden und verwerfenden Zustandes in der Menge $\{q_1, \ldots, q_n\}$.

Kodierung von Turingmaschinen

- Deterministische Maschine: δ bildet genau $|Q| \times |\Gamma|$ Paare ab.
- Hänge für jedes Paar den Teilstring

$$10^u 10^v 10^w 10^x 10^{d+2}$$

an, falls $\delta(q_u, a_v) = (q_w, a_x, d)$ mit $d \in \{-1, 0, 1\}$.

- Interpretiere $\langle M \rangle$ als Binärdarstellung einer Zahl $n \in \mathbb{N}$.
  - Die Abbildung g mit $\langle M \rangle \mapsto n$ wird Gödelisierung genannt.

$\Rightarrow$ die Menge der Turingmaschinen ist abzählbar.

  - "Schließen der Lücken" in der Abbildung g:
    Aufzählung der Turingmaschinen als

$$M_1, M_2, M_3 \ldots$$

Universelle Turingmaschine

- Turingmaschinen können selbst als Eingaben benutzt werden.
- Die Universelle Turingmaschine M_U simuliert eine gegebene Turingmaschine auf einem gegebenen Wort.
- Formal:

$$L(M_U) = \{\langle M \rangle w \ : \ M \text{ Turingmaschine und } w \in L(M)\}.$$

  - Beachte: im String $\langle M \rangle w$ ist eindeutig bestimmt (und berechenbar), wo die Kodierung von M aufhört und wo w anfängt.
  - Die Universelle Turingmaschine erhält Kodierung einer Turingmaschine M und Wort $w \in \Sigma^*$ als Eingabe und akzeptiert genau dann, wenn M das Wort w akzeptiert.
  - Dazu simuliert die universelle Turingmaschine die Maschine M auf w Schritt für Schritt.

Diagonalisierung

- Diagonalisierung: ein vorbereitender Satz.

Satz von Cantor

Es gibt es keine Bijektion zwischen $\mathbb{N}$ und der Potenzmenge von $\mathbb{N}$.

Beweis.

- Angenommen, es gibt doch eine Bijektion $f: \mathbb{N} \to 2^{\mathbb{N}}$.
- Betrachten die folgende unendliche zweidimensionale Matrix:
  - Zeilen und Spalten mit den natürlichen Zahlen indiziert.
  - Eintrag an Position (i, j) ist 1 falls $j \in f(i)$ und 0 falls $j \notin f(i)$.

Diagonalisierung

	1	2	3	4	5	6	7	8	9	$\cdots$
$f(1)$	0	0	0	0	1	1	0	0	0	
$f(2)$	1	0	0	1	1	0	0	0	1	
$f(3)$	0	0	1	0	1	1	1	1	0	
$f(4)$	1	0	0	0	0	0	0	1	1	
$f(5)$	1	0	0	0	0	0	1	0	0	
$f(6)$	0	0	1	0	1	0	0	0	1	$\cdots$
$f(7)$	0	1	1	1	1	0	0	0	1	
$f(8)$	0	0	0	0	1	0	0	1	1	
$f(9)$	0	1	1	0	1	1	0	0	1	
$\vdots$					$\vdots$					

- $i$te Zeile der Matrix beschreibt die Menge $f(i)$.
- Z.B. $f(4) = \{1, 8, 9, \ldots\}$.

Diagonalisierung

	1	2	3	4	5	6	7	8	9	$\cdots$
$f(1)$	0	0	0	0	1	1	0	0	0	
$f(2)$	1	0	0	1	1	0	0	0	1	
$f(3)$	0	0	1	0	1	1	1	1	0	
$f(4)$	1	0	0	0	0	0	0	1	1	
$f(5)$	1	0	0	0	0	0	1	0	0	
$f(6)$	0	0	1	0	1	0	0	0	1	$\cdots$
$f(7)$	0	1	1	1	1	0	0	0	1	
$f(8)$	0	0	0	0	1	0	0	1	1	
$f(9)$	0	1	1	0	1	1	0	0	1	
$\vdots$					$\vdots$					

- f bijektiv $\Rightarrow$ jede Teilmenge von $\mathbb{N}$ taucht in genau einer Zeile auf.
- Betrachte das binäre Komplement der Diagonalen: $1, 1, 0, 1, 1, 1, 1, 0, 0, \ldots$
- Kann in keiner Zeile stehen: Widerspruch an Position (i, i).

Diagonalisierung

- Formal: das Komplement der Diagonalen:

$$D = \{i \in \mathbb{N} \ : \ i \notin f(i)\}$$

- Angenommen $D = f(j)$ für ein $j \in \mathbb{N}$.
  - Ein solches j muss existieren, da f nach Annahme bijektiv ist.
- Dann gilt

$$\begin{aligned} j \in D &\Leftrightarrow j \notin f(j) && \text{Def. } D \\ &\Leftrightarrow j \notin D && f(j) = D, \end{aligned}$$

- Widerspruch, also kann keine Bijektion $f : \mathbb{N} \to 2^{\mathbb{N}}$ existieren.

Das (spezielle) Halteproblem

- Das Halteproblem ist das Problem:
  - Gegeben eine Turingmaschine M und ein Wort w, hält M auf w?
  - Formal: $H = \{\langle M \rangle w \ : \ M \text{ Turingmaschine und } M \text{ hält auf } w\}$.
- Das spezielle Halteproblem ist das Problem:
  - Gegeben eine Turingmaschine M, hält M auf ihrer eigenen Kodierung $\langle M \rangle$?
  - Formal: $H_s = \{\langle M \rangle \langle M \rangle \ : \ M \text{ Turingmaschine und } M \text{ hält auf } \langle M \rangle\}$.

Satz

Das spezielle Halteproblem ist unentscheidbar.

Beweis

- Betrachten die folgende unendliche zweidimensionale Matrix:
  - Zeilen und Spalten mit den natürlichen Zahlen indiziert.
  - Eintrag an Position (i, j) ist 1 falls M_i auf der Kodierung $\langle M_j \rangle$ hält, und 0 sonst.

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$	$\langle M_6 \rangle$	$\langle M_7 \rangle$	$\langle M_8 \rangle$	$\langle M_9 \rangle$	$\cdots$
M_1	0	1	0	1	1	1	0	1	0	
M_2	1	1	0	0	0	0	0	0	0	
M_3	1	0	0	0	0	1	1	1	0	
M_4	0	0	1	0	0	1	0	1	0	
M_5	1	0	0	1	1	1	0	0	0	
M_6	1	1	1	1	1	0	0	1	1	$\cdots$
M_7	1	1	0	1	1	0	1	0	0	
M_8	0	1	0	0	0	0	0	1	0	
M_9	0	1	0	0	0	1	1	0	0	
$\vdots$					$\vdots$					

Beweis

- Angenommen das spezielle Halteproblem ist entscheidbar.
- Dann existiert Turingmaschine M_s, die eine Eingabe $\langle M \rangle \langle M \rangle$ genau dann akzeptiert, wenn M auf dem Wort $\langle M \rangle$ hält.
- Sei M'_s die Maschine, die auf einer Eingabe $\langle M \rangle$
  - die Maschine M_s auf $\langle M \rangle \langle M \rangle$ simuliert und
  - falls M_s die Eingabe $\langle M \rangle \langle M \rangle$ akzeptiert, so geht M'_s in eine triviale Endlosschleife und
  - falls M_s die Eingabe $\langle M \rangle \langle M \rangle$ verwirft, so terminiert M'_s und akzeptiert.
- Es gilt also für $\langle M_i \rangle$

$$\begin{aligned} M'_s \text{ hält auf } \langle M_i \rangle &\Leftrightarrow M_s \text{ verwirft } \langle M_i \rangle \langle M_i \rangle \\ &\Leftrightarrow M_i \text{ hält auf } \langle M_i \rangle \text{ nicht.} \end{aligned}$$

→ Das Komplement der Diagonalen!

Beweis

- Wenn M_s existiert, so existiert auch M'_s und es gilt $M'_s = M_i$ für ein $i \in \mathbb{N}$.
- Dann gilt

$$\begin{aligned} M_i = M'_s \text{ hält auf } \langle M_i \rangle &\Leftrightarrow M_s \text{ verwirft } \langle M_i \rangle \langle M_i \rangle \\ &\Leftrightarrow M_i \text{ hält auf } \langle M_i \rangle \text{ nicht.} \end{aligned}$$

- Dies ist ein Widerspruch, also existiert M_s nicht und das spezielle Halteproblem ist unentscheidbar.

Das spezielle Halteproblem

- Das spezielle Halteproblem ist das Problem:
  - Gegeben eine Turingmaschine M, hält M auf ihrer eigenen Kodierung $\langle M \rangle$?
  - Formal: $H_s = \{\langle M \rangle \langle M \rangle \ : \ M \text{ Turingmaschine und } M \text{ hält auf } \langle M \rangle\}$.
- Das spezielle Halteproblem ist semi-entscheidbar.
  - Zu einer gegebenen Turingmaschine M, simuliere M auf $\langle M \rangle$.

Korollar

Das Komplement des speziellen Halteproblems ist nicht semi-entscheidbar.

Beweis.

- L ist entscheidbar genau dann, wenn L und $\bar{L}$ semi-entscheidbar sind.

Das Halteproblem

- Das Halteproblem ist das Problem:
  - Gegeben eine Turingmaschine M und ein Wort w, hält M auf w?
  - Formal: $H = \{\langle M \rangle w \ : \ M \text{ Turingmaschine und } M \text{ hält auf } w\}$.

Satz

Das Halteproblem ist unentscheidbar

Beweis.

- Angenommen M_H ist TM, die H entscheidet.
- Konstruiere TM M_s, die
  - auf der Eingabe $\langle M \rangle \langle M \rangle$ die Maschine M_H simuliert.
- M_s entscheidet H_s, ein Widerspruch!

Turing-Reduktionen

- Im Beweis: Angenommen Maschine M_H existiert.
- M_H wird als nun als Blackbox benutzt und auf einem Wort simuliert.
- Wir erhalten eine Maschine für H_s, ein Widerspruch.
- Historisch: M_H wird als ein Orakel benutzt.
- Heute würden wir sagen: M_H wird als Unterprogramm aufgerufen.

Orakel-Turingmaschinen

- Sei $L \subseteq \Sigma^*$ eine Sprache über einem Alphabet Σ.
- Eine Orakel-Turingmaschine mit Orakel L ist eine Turingmaschine M
  - mit einem zusätzlichen Arbeitsband, dem Orakelband, und
  - drei ausgezeichneten Zuständen q_j, q_n und $q_?$.
  - Schreibt M ein Wort $w \in \Sigma^*$ auf das Orakelband und wechselt dann in den Zustand $q_?$, so "befragt M das Orakel":
    - Der Nachfolgezustand von $q_?$ ist q_j falls $w \in L$ und q_n falls $w \notin L$.
    - Der Inhalt des Orakelbandes verschwindet.
- Anschaulich ist ein Orakel eine Blackbox, die ein gegebenes Problem lösen kann.

Turing-Reduktionen

- Seien $L_1, L_2 \subseteq \Sigma^*$ Sprachen.
- L_1 ist Turing-reduzierbar auf L_2, geschrieben

$$L_1 \leq_T L_2,$$

wenn es eine Orakel-Turingmaschine mit Orakel L_2 gibt, die L_1 entscheidet.

Turing-Reduktionen

Lemma

Seien $L_1, L_2 \subseteq \Sigma^*$ mit $L_1 \leq_T L_2$.

- *L_2 entscheidbar $\Rightarrow$ L_1 entscheidbar.*
- *L_1 unentscheidbar $\Rightarrow$ L_2 unentscheidbar.*

Beweis.

- Sei L_2 entscheidbar und sei M_2 TM, die L_2 entscheidet.
- Da $L_1 \leq_T L_2$, existiert eine Orakelmaschine M mit Orakel L_2, die L_1 entscheidet.
- Erhalten aus M eine TM M_1, die L_1 entscheidet:
  - ersetze Orakelband durch normales Arbeitsband,
  - ersetze jeden Orakelaufruf von M durch Simulation von M_2.

Das Wortproblem

- Das Wortproblem ist das Problem:
  - Gegeben eine Turingmaschine M und ein Wort w, akzeptiert M das Wort w?
  - Formal: $W = \{\langle M \rangle w \ : \ M \text{ Turingmaschine und } w \in L(M)\}$.
- Es gilt $L(M_U) = W$.

Satz

Es gilt $H \leq_T W$.

- Da H unentscheidbar, ist nach Lemma das Wortproblem unentscheidbar.

Beweis

- Konstruiere eine Orakelmaschine M_H mit Orakel W, die H entscheidet.
- Auf Eingabe $\langle M \rangle w$ für eine Turingmaschine M und ein Wort w berechne die Kodierung $\langle M' \rangle$ einer Maschine M':
  - M' arbeitet genau wie M, aber wenn M terminiert, so akzeptiert M' die Eingabe (egal ob M verwirft oder akzeptiert).

$$w \in L(M') \Leftrightarrow M \text{ terminiert auf } w.$$

Beweis

$$w \in L(M') \Leftrightarrow M \text{ terminiert auf } w.$$

- Rufe das Orakel für W auf der Eingabe $\langle M' \rangle w$ auf.
  - Zustand $q_j \rightarrow$ akzeptiere.
  - Zustand $q_n \rightarrow$ verwerfe.

M_H akzeptiert w

$\Leftrightarrow$ $\langle M' \rangle w \in W$

$\Leftrightarrow$ $w \in L(M')$

$\Leftrightarrow$ M terminiert auf w

$\Rightarrow$ M_H ist Orakelmaschine mit Orakel W, die H entscheidet.

(Many-one) Reduktionen

- Im Beweis: ein einziger Orakelaufruf und die Antwort wird sofort übernommen → Many-one-Reduktion, auf deutsch meistens einfach Reduktion.
- Eine Reduktion von $L_1 \subseteq \Sigma^*$ auf $L_2 \subseteq \Sigma^*$ ist eine (totale) berechenbare Funktion

$$f : \Sigma^* \to \Sigma^*,$$

so dass für alle $w \in \Sigma^*$:

$$w \in L_1 \Leftrightarrow f(w) \in L_2.$$

- L_1 ist auf L_2 reduzierbar, $L_1 \leq L_2$, falls es eine Reduktion von L_1 nach L_2 gibt.

Reduktionen

Lemma

Seien $L_1, L_2 \subseteq \Sigma^*$ *mit* $L_1 \leq L_2$.

- L_2 *entscheidbar* $\Rightarrow$ L_1 *entscheidbar.*
- L_1 *unentscheidbar* $\Rightarrow$ L_2 *unentscheidbar.*

Beweis

Beweis.

- $L_1 \leq L_2 \rightarrow$ berechenbare Funktion $f: \Sigma^* \rightarrow \Sigma^*$, so dass f.a. $w \in \Sigma^*$ gilt: $w \in L_1 \Leftrightarrow f(w) \in L_2$.
- L_2 entscheidbar $\rightarrow$ TM M_2, die L_2 entscheidet.
- Auf Eingabe $w \in \Sigma^*$: berechne $f(w)$ und simuliere M_2 auf $f(w)$.
- M_2 terminiert auf jeder Eingabe $\rightarrow$ übernehme Antwort.

- Berechnung terminiert auf jeder Eingabe w und w wird akzeptiert genau dann, wenn
  - $f(w) \in L(M_2) \Leftrightarrow$
  - $f(w) \in L_2 \Leftrightarrow$
  - $w \in L_1$.

Das Wortproblem

Lemma

$L_1 \leq L_2 \Rightarrow L_1 \leq_T L_2$.

Es gilt nicht nur $H \leq_T W$, sondern sogar

Satz

Es gilt $H \leq W$.

Beweis.

- Die Funktion

$$f\colon \Sigma^* \to \Sigma^*\colon y \mapsto \begin{cases} y & \text{falls } y \neq \langle M \rangle w \\ \langle M' \rangle w & \text{falls } y = \langle M \rangle w, \end{cases}$$

  ist eine totale berechenbare Funktion.

Turing-Reduktionen vs Reduktionen

- Eine Reduktion liefert stärkere Aussagen als eine Turing-Reduktion.

Lemma

Seien $L_1, L_2 \subseteq \Sigma^*$ mit $L_1 \leq L_2$.

- *L_2 semi-entscheidbar $\Rightarrow$ L_1 semi-entscheidbar*
- *L_1 nicht semi-entscheidbar $\Rightarrow$ L_2 nicht semi-entscheidbar.*

Beweis analog.

- Das Lemma gilt nicht für Turing-Reduktionen.
- Z.B. $H \leq_T \bar{H}$ und $\bar{H} \leq_T H$, aber $\bar{H}$ ist nicht semi-entscheidbar, da sonst H entscheidbar wäre.

Zusammenfassung

- Es gibt wichtige unentscheidbare Probleme, z.B. das Halteproblem und das Wortproblem.
- Gezeigt für ein erstes konkretes Problem: das spezielle Halteproblem mit Diagonalisierung.
- Dann mit Hilfe von Reduktionen bewiesen für das Halteproblem und das Wortproblem.
- Neue Unentscheidbarkeitsbeweise: meistens mit Reduktionen.