

Das Halteproblem

Dr. Eva Richter

11. Mai 2012

Fixpunkte von Funktionen

Definition

Sei $f : X \rightarrow X$ eine Funktion. Ein Element $x \in X$ heißt **Fixpunkt von f** , wenn gilt $f(x) = x$.

Beispiele:

- 1 für $id : \mathbb{R} \rightarrow \mathbb{R}$ mit $x \mapsto x$ ist jede reelle Zahl ein Fixpunkt
- 2 für $sqr : \mathbb{R} \rightarrow \mathbb{R}$ mit $x \mapsto x^2$ sind $x = 1$ und $x = 0$ die einzigen Fixpunkte

- beschreiben Bedingungen an Funktionen, unter denen Fixpunkte existieren
- werden in der Mathematik oft gebraucht, um die Eindeutigkeit von Lösungen zu beweisen
- berühmte Fixpunktsätze sind: **Browerscher Fixpunktsatz**, **Banachscher Fixpunktsatz**, **Fixpunktsatz von Knaster-Tarski**

Fixpunktsatz von Banach

Satz

Sei (M, d) ein nichtleerer vollständiger metrischer Raum, dann besitzt jede Kontraktion $\varphi : M \rightarrow M$ einen Fixpunkt.

Definition

Ein Paar (M, d) , wobei M eine Menge ist und $d : M \times M \rightarrow \mathbb{R}$ mit

- 1 $\forall x \in M \quad d(x, x) = 0$
- 2 $\forall x, y \in M \quad d(x, y) = d(y, x)$
- 3 $\forall x, y, z \in M \quad d(x, y) + d(y, z) \geq d(x, z)$

heißt **metrischer Raum**.

Vollständigkeit: jede Cauchyfolge konvergiert

Kontraktion: $\exists \lambda \in (0, 1]$ mit $d(\varphi(x), \varphi(y)) \leq \lambda d(x, y)$

Babylonisches Wurzelziehen für natürliche Zahlen

$$f(x) = \frac{1}{2}\left(x + \frac{a}{x}\right)$$

1 f ist Kontraktion mit $\lambda = 0.5$ auf dem Intervall $[w, w + 1]$,
wobei $w = \max\{n \in \mathbb{N} \mid n^2 < a\}$

2 Fixpunkt von f ist \sqrt{a}

3 sei x_0 ein Startwert, die Folge $(x_n)_{n \in \mathbb{N}}$ berechnet sich durch
 $x_n = f^n(x_0)$

4

$$\frac{1}{x_n} \leq \sqrt{a} \leq x_n$$

5 Folge $(x_n)_{n \in \mathbb{N}}$ konvergiert gegen \sqrt{a}

Fixpunktsatz von Knaster-Tarski

Satz

Sei L ein vollständiger Verband und $f : L \rightarrow L$ eine ordnungserhaltende Funktion. Dann ist die Menge der Fixpunkte von f ebenfalls ein vollständiger Verband.

- da vollständige Verbände nicht leer sind, hat f mindestens einen Fixpunkt
- kleinster Fixpunkt von f ist $\lim_{n \rightarrow \infty} f^n(0)$
- in der Informatik braucht man kleinste Fixpunkte um Semantik von Programmiersprachen zu definieren

Das Halteproblem

- eine der philosophisch wichtigsten Aussagen der Berechenbarkeitstheorie
- spezielles Problem, das algorithmisch nicht lösbar ist
- zeigen, dass Computer in einer sehr grundlegenden Weise beschränkt sind
- Art der unlösbaren Probleme sind weder erkünstelt noch weltfremd
- das allgemeine Problem der Software-Verifikation ist nicht durch einen Computer lösbar

Formulierung des Halteproblems

$ATM = \{(M, w) \mid M \text{ ist Turingmaschine und } M \text{ akzeptiert } w\}$

Satz

ATM ist unentscheidbar.

Akzeptierer für ATM :

$U =$

„auf Eingabe $\langle M, w \rangle$, mit Turingmaschine M und Wort w

- 1 Simuliere M auf w .
- 2 Falls M das Wort w akzeptiert, **accept**; falls M ablehnt, **reject**.

Turing-Akzeptierbarkeit des Halteproblems

- U gerät in eine Schleife, wenn M bei Eingabe von w in eine Schleife gerät (kein Entscheider)
- falls Algorithmus festzustellen könnte, dass M auf w niemals anhält, würde er das Paar ablehnen
- daher heißt A_{TM} auch das **Halteproblem**
- werden zeigen, dass es keinen Algorithmus geben kann, der feststellt, ob eine beliebige Maschine auf einer Eingabe jemals anhalten wird
- U ist Beispiel für eine **universelle Turingmaschine**; U kann die Arbeit einer beliebigen Turingmaschine aus deren Beschreibung simulieren
- UTM spielte wichtige Rolle als Anstoß für die Entwicklung von Computern mit gespeicherten Programmen

Die Methode der Diagonalisierung

- geht auf eine Technik von Georg Cantor zurück
- Cantor wollte Größe von unendlichen Mengen messen, suchte Methode um festzustellen, welche von zwei unendlich großen Mengen „größer“ ist
- bei endlichen Mengen kann man Elemente zählen, bei unendlichen Mengen wird man damit nicht fertig
- endliche Mengen haben genau dann die gleiche Größe, wenn man ihre Elemente paarweise einander zuordnen kann
- Cantor übertrug diese Idee auf unendliche Mengen

Definition

Seien A und B Mengen und $f : A \rightarrow B$ eine Funktion.

- 1 f heißt **eindeutig** oder **injektiv**, wenn es keine Elemente von A gibt, die unter f dasselbe Bild haben, $f(a) \neq f(a')$ für $a \neq a'$.
- 2 f heißt Funktion **auf B** oder **surjektiv**, wenn für alle $b \in B$ ein Element $a \in A$ existiert, so dass $f(a) = b$.
- 3 A und B **haben dieselbe Größe**, wenn es eine eindeutige Funktion f von A auf B gibt. f heißt dann auch **Bijektion** oder **Korrespondenz**.

Beispiele für Korrespondenzen

- \mathbb{N} , Menge der natürlichen Zahlen und \mathbb{E} , Menge geraden Zahlen sind gleich groß

$f : \mathbb{N} \rightarrow \mathbb{E}$ mit $n \mapsto 2n$ ist eine Bijektion

- $\mathbb{Q}^+ = \{\frac{m}{n} \mid m, n \in \mathbb{N} \setminus \{0\}\}$ und \mathbb{N} sind gleich groß

Liste der Zahlen entlang der Nebendiagonalen unter Auslassen von Wiederholungen:

1 2 1 3 1 4 3 2 1
 $\frac{1}{1}$, $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$, $\frac{1}{3}$, $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$, \dots

Definition

Eine Menge A heißt **abzählbar**, wenn sie endlich ist oder dieselbe Größe wie \mathbb{N} hat.

- Anstelle von $A \rightarrow B$ beweist man $(A \wedge \neg B) \rightarrow \mathbf{f}$.
- $C \rightarrow \mathbf{f}$ ist genau dann wahr, wenn C falsch ist.
- Wir müssen also zeigen, dass $A \wedge \neg B$ falsch ist.

Anwendungsbereich:

- zeigen, dass eine gewisse Eigenschaft **nicht** vorliegt (x ist ungerade),
- man hat keine positiven Kriterien (x ist **nicht** durch 2 teilbar),
- kann aber das Gegenteil zuverlässig erkennen.

Wurzel aus 2 ist irrational

Annahme	$\sqrt{2}$ ist rational.
Umformungen	\Leftrightarrow es gibt teilerfremde $a, b \in \mathbb{Z}$ mit $\sqrt{2} = \frac{a}{b}$ $\Leftrightarrow 2b^2 = a^2$ $\Rightarrow a^2$ ist gerade $\Rightarrow a$ ist gerade \Rightarrow es gibt ein $s \in \mathbb{Z}$ mit $a = 2s$, $\Rightarrow b^2 = 2s^2$ $\Rightarrow b$ ist gerade
Widerspruch	a und b haben gemeinsamen Teiler (nämlich 2)
Konklusion	$\sqrt{2}$ ist nicht rational, daher gilt $\sqrt{2}$ ist irrational.

Satz

\mathbb{R} ist überabzählbar.

Beweis:

- mit Hilfe der Diagonalisierungsmethode, indirekt
- Annahme: es gibt eine Bijektion $f : \mathbb{N} \rightarrow \mathbb{R}$ mit der Liste

$$(1, r_1), (2, r_2), (3, r_3), (4, r_4) \dots$$

- konstruieren eine Zahl x , die nicht in der Liste auftaucht
- die i -te Nachkommastelle von $x = 0.a_1a_2 \dots a_i \dots$ wird definiert durch:

$$a_i := \begin{cases} 2 & \text{falls die } i\text{-te Stelle von } f(i) \text{ ungleich } 2 \text{ ist} \\ 3 & \text{sonst.} \end{cases}$$

- x unterscheidet sich von jeder Zahl in der Liste an mindestens einer Stelle, kann also selbst nicht in der Liste gewesen sein

□

Folgerung

Es gibt Sprachen, die nicht Turing-akzeptierbar sind.

Teilziele für den Beweis:

- 1 Die Menge aller Turingmaschinen ist abzählbar.
- 2 Die Menge aller Turingmaschinen ist aufzählbar.
- 3 Die Menge aller Sprachen ist überabzählbar.

- Turingmaschine, die möglichst viele Einsen aufs Band schreibt und anhält
- sei k_n die maximale Anzahl der Einsen, die eine Turingmaschine mit n Zuständen, die anhält aufs Band schreiben kann
- $\Sigma : \mathbb{N} \rightarrow \mathbb{N}$ mit $\Sigma(n) = k_n$ ist nicht berechenbar
- $S : \mathbb{N} \rightarrow \mathbb{N}$ mit $S(n)$ ist maximale Anzahl der Schritte, die eine Turingmaschine, die immer anhält, ausführen kann ist nicht berechenbar.

Zustände n	Anzahl TM	$\Sigma(n)$
1	64	1
2	20736	4
3	16777216	6
4	$25,6 \cdot 10^9$	13
5	$\approx 63,4 \cdot 10^{12}$	≥ 4098
6	$\approx 232 \cdot 10^{15}$	$> 3,514 \cdot 10^{18267}$

Existenz nicht Turing-akzeptierbarer Sprachen

- 1 Menge aller Turingmaschinen ist abzählbar, da Σ^* über einem Alphabet Σ abzählbar ist: man zählt der Reihe nach alle Zeichenketten der Länge 0, der Länge 1, der Länge 2 usw. auf
- 2 Menge der Turingmaschinen ist abzählbar, da sich jede Turingmaschine M durch eine endliche Zeichenkette $\langle M \rangle$ darstellen lässt
- 3 zähle alle Zeichenketten auf, entferne ungültige Kodierungen

Existenz nicht Turing-akzeptierbarer Sprachen

- Menge \mathbb{B} aller unendlichen binären ist Folgen überabzählbar (Übungsaufgabe).
- geben Bijektion von \mathcal{L} , Menge aller Sprachen über Σ zu \mathbb{B} an
- $\Sigma^* = \{s_1, s_2, \dots\}$ Menge der Wörter über Σ
- zu jeder Sprache $A \in \mathcal{L}$ gibt es **charakteristische Folge** χ_A mit:

$$\chi_A(s_i) = \begin{cases} 1 & \text{falls } s_i \in A \\ 0 & \text{sonst} \end{cases}$$

- $f : \mathcal{L} \rightarrow \mathbb{B}$ mit $A \mapsto \chi_A$ ist injektiv und surjektiv, also eine Bijektion.
- Da \mathbb{B} überabzählbar ist, ist auch \mathcal{L} überabzählbar.

Es gibt aber nur abzählbar viele Turingmaschinen!

Unentscheidbarkeit des Halteproblems

Satz

$A_{TM} = \{ \langle M, w \rangle \mid M \text{ ist Turingmaschine und } M \text{ akzeptiert } w \}$
ist nicht entscheidbar

- nehme an, A_{TM} wäre entscheidbar mit einem Entscheider H
- H akzeptiert $\langle M, w \rangle$, falls M auf w anhält, lehnt ab bei reject oder wenn M in eine Schleife gerät
- konstruiere Turingmaschine D , die H als Unterprogramm enthält

Die seltsame Maschine D

$D =$

„auf Eingabe von $\langle M \rangle$ für Turingmaschine M

- 1 Starte H auf der Eingabe $\langle M, \langle M \rangle \rangle$
- 2 Gebe das Gegenteil von der Ausgabe von H aus, d.h. wenn H akzeptiert—**reject** wenn H ablehnt—**accept**.“

Verhalten von D :

$$D(\langle M \rangle) = \begin{cases} \text{accept} & \text{falls } M \text{ die Eingabe } \langle M \rangle \text{ nicht akzeptiert} \\ \text{reject} & \text{falls } M \text{ die Eingabe } \langle M \rangle \text{ akzeptiert} \end{cases}$$

$$D(\langle D \rangle) = \begin{cases} \text{accept} & \text{falls } D \text{ die Eingabe } \langle D \rangle \text{ nicht akzeptiert} \\ \text{reject} & \text{falls } D \text{ die Eingabe } \langle D \rangle \text{ akzeptiert} \end{cases}$$

Was auch immer D tut, nach ihrer Definition muss sie das genaue Gegenteil tun.

Das ist Widerspruch zur Annahme, also können D und demzufolge auch H nicht existieren. \square

- 1 nehme an, wir hätten Entscheider H für A_{TM}
- 2 konstruiere D , die bei Eingabe von $\langle M \rangle$ genau dann akzeptiert, wenn M bei Eingabe von $\langle M \rangle$ nicht akzeptiert
- 3 lasse D mit der Eingabe $\langle D \rangle$ laufen und erhalte ein Ergebnis, das der Definition von D widerspricht

Was hat das alles mit der Diagonalisierungsmethode zu tun?

Verhalten aller Turingmaschinen

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	\dots
M_1	accept		accept		
M_2	accept	accept	accept	accept	
M_3					\dots
M_4	accept	accept			
\vdots		\vdots			

Verhalten von H mit den Eingaben $\langle M_i, \langle M_j \rangle \rangle$

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	\dots
M_1	accept	reject	accept	reject	
M_2	accept	accept	accept	accept	
M_3	reject	reject	reject	reject	
M_4	accept	accept	reject	reject	
\vdots		\vdots			

Was macht D ?

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	\dots	$\langle D \rangle$	\dots
M_1	<u>accept</u>	reject	accept	reject			
M_2	accept	<u>accept</u>	accept	accept	\dots		
M_3	reject	reject	<u>reject</u>	reject			
M_4	accept	accept	reject	<u>reject</u>			
\vdots		\vdots					
D	reject	reject	accept	accept		<u>?</u>	
\vdots		\vdots					

D berechnet stets das Gegenteil von den Einträgen in der Diagonale.

Widerspruch taucht an der Stelle mit dem Fragezeichen auf, wo der Eintrag das Gegenteil von sich selbst sein muss.

Nicht-Turing-akzeptierbare Sprachen

- es gibt nicht entscheidbare Sprachen, z. B. A_{TM}
- es gibt auch Sprachen, die nicht einmal Turing-akzeptierbar sind (A_{TM} gehört nicht dazu)
- werden zeigen, dass wenn eine Menge A und auch ihr Komplement \bar{A} Turing-akzeptierbar sind, dann ist A entscheidbar
- da es nicht entscheidbare Mengen gibt, folgt daraus, dass entweder diese Mengen oder deren Komplemente nicht Turing-akzeptierbar sind
- eine Menge ist **co-Turing-akzeptierbar**, wenn sie das Komplement einer Turing-akzeptierbaren Menge ist

Satz

Eine Menge A ist genau dann entscheidbar, wenn A und \bar{A} Turing-akzeptierbar sind.

Wir müssen zwei Richtungen beweisen:

- 1 Wenn A entscheidbar ist, dann sind sowohl A als auch \bar{A} Turing-akzeptierbar.
- 2 Wenn sowohl A als auch \bar{A} Turing-akzeptierbar sind, dann ist A entscheidbar.

- jede entscheidbare Sprache ist Turing-akzeptierbar, da ein Entscheider insbesondere auch ein Akzeptierer ist
- aus dem Entscheider für A baut man einen Akzeptierer für \overline{A} , indem man die negativen Antworten des Entscheiders als *accept* für das Komplement betrachtet

seien M_1 und M_2 Akzeptierer für A bzw. \overline{A} , konstruiere M :

$M =$

„Auf Eingabe von w

- ➊ Starte M_1 und M_2 parallel auf w .
 - ➋ Wenn M_1 akzeptiert, **accept**; wenn M_2 akzeptiert, **reject**.“
- paralleles Starten von M_1 und M_2 wird durch 2-Band-TM simuliert
 - entweder M_1 oder M_2 akzeptieren w , da $w \in A$ oder $w \in \overline{A}$
- M hält in jedem Fall an, ist also ein Entscheider, akzeptiert alle Strings aus A und lehnt alle Strings aus \overline{A} ab

Also ist A entscheidbar. □

Folgerung

$\overline{A_{TM}}$ ist nicht Turing-akzeptierbar.

Beweis:

- A_{TM} ist Turing-akzeptierbar
- wäre $\overline{A_{TM}}$ auch Turing-akzeptierbar, dann wäre A_{TM} entscheidbar
- wäre Widerspruch zum Satz über Unentscheidbarkeit von A_{TM}
- also ist $\overline{A_{TM}}$ nicht Turing-akzeptierbar.

□