

# Berechenbare Analysis

## SoSe 19

Benedikt Lücken-Winkels

May 16, 2019

### Contents

<b>1</b>	<b>Vorlesung</b>	<b>2</b>
<b>2</b>	<b>Vorlesung</b>	<b>2</b>
2.1	Berechenbarkeit . . . . .	2
2.2	Entscheidbarkeit . . . . .	2
2.3	Berechenbare Reelle Zahlen . . . . .	2
<b>3</b>	<b>Vorlesung</b>	<b>3</b>
3.1	Binary Sequence . . . . .	4
<b>4</b>	<b>Vorlesung</b>	<b>4</b>
<b>5</b>	<b>Vorlesung</b>	<b>4</b>
<b>6</b>	<b>Vorlesung</b>	<b>5</b>
<b>7</b>	<b>General Stuff</b>	<b>5</b>
<b>8</b>	<b>1. Übung</b>	<b>5</b>

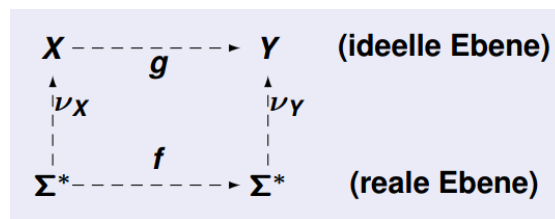
# 1 Vorlesung

## 2 Vorlesung

### 2.1 Berechenbarkeit

Es gibt einen Algorithmus, der die Zahl angeben kann (es gibt nur eine abzählbar unendliche Anzahl an Algorithmen, aber überabzählbar viele reelle Zahlen)

Figure 1:  $g$  ist  $(\nu_x, \nu_y)$  berechenbar, wenn  $g$  von einer berechenbaren Funktion  $f$  realisiert wird



### 2.2 Entscheidbarkeit

**Diagonalisierung** Wären die Reellen Zahlen abzählbar, wäre die Diagonalzahl darin enthalten (!Widerspruch).

Table 1: Diagonalisierungsbeispiel:  $x_\infty$  kann nicht in der Liste enthalten sein

$x_0$	0.500000
$x_1$	0.411110
$x_2$	0.312110
$x_3$	0.222220
$x_4$	0.233330
...	...
$x_\infty$	0.067785....

**Definition** Menge  $A$  Entscheidbar, wenn eine Funktion  $f_A(x)$ , die entscheidet, ob  $x \in A$  berechenbar ist.

### 2.3 Berechenbare Reelle Zahlen

**Konstruktive Mathematik** Formulierung algorithmischen Rechnens: zB  $\exists$  neu definiert als "es existiert ein Algorithmus". Nicht mehr für "klassische Mathematiker" lesbar

**Definition** Für  $x \in \mathbb{R}$  sind die Bedingungen äquivalent (wenn eine Bedingung erfüllt ist, sind alle Erfüllt):

1. Eine TM erzeugt eine unendlich lange binäre Representation von  $x$  auf dem Ausgabeband
2. **Fehlerabschätzung** Es gibt eine TM, die Approximationen liefert. Formal:  $q : \mathbb{N} \rightarrow \mathbb{Q}$   $(q_i)_{i \in \mathbb{N}}$  ist Folge rationaler Zahlen, die gegen  $x$  konvergiert. Bedeutet, dass alle  $q_i$  innerhalb eines bestimmten beliebig kleinen Bereichs um  $x$  liegen. Größter möglicher Fehler  $2^0 = 1$
3. **Intervallschachtelung** Es gibt eine berechenbare Intervallschachtelung: Angabe zweier Folgen rationaler Zahlen mit der Aussage, dass  $x$  dazwischen liegt. Ziel: Abstände von linker und rechter Schranke soll gegen null gehen.
4. **Dedekindscher Schnitt** Menge  $\{q \in \mathbb{Q} | q < x\}$  ist entscheidbar. Beispiel  $\sqrt{2}$  ist berechenbar.  $\{q | q < \sqrt{2}\} = \{q | q^2 < 2\}$ .  $\Rightarrow$  Es gibt einen Test, ob die Zahl kleiner ist.
5.  $z \in \mathbb{Z}$   $A \subseteq \mathbb{N}$ ,  $x_A = \sum_{i \in A} 2^{-i}$ ,  $x = z + x_A$
6. Es existiert eine Kettenbruchentwicklung

### Folgerungen / Beispiele

- $\Rightarrow$  Für Berechenbarkeit muss nur eine der Bedingungen bewiesen werden. Menge der berechenbaren reellen Zahlen  $= \mathbb{R}_c$
- Nicht berechenbare reelle Zahlen durch Diagonalisierung konstruierbar
- $e$  berechenbar, weil die Fehlerabschätzung (2) existiert
- $\pi$  (Notiert als alternierende Reihe) berechenbar, weil Intervallschachtelung existiert
- $\sqrt{2}$  berechenbar, weil Dedekindscher Schnitt existiert.

**Implementierung** Ziel: zB Berechnung von Differentialgleichungen

## 3 Vorlesung

**Implementierung in C++** Ziel: shared pointer für temporäre Variablen verstecken (durch wrapper)

- (binary sequence) bs: ein Bit nach dem anderen wird ausgegeben. binseq gibt zur natürlichen Zahl  $n$  und liefert das  $n$ -te Bit der reellen Zahl (Vorzeichen, 0 oder 1).
- (rational approximations) ra: Fehler beliebiger Größe (Ganze Zahlen). approx rationale Approximation mit einem beliebig großem Fehler. (Abänderung der Definition, weil ganze Zahlen zulässig)

- ni: Untere und obere Schranke. lower/upperbound gibt n-te Schranke
- (Dedekind cut) dc: Ist eine Zahl kleiner. smaller entscheidet, ob die angegebene Rationale Zahl kleiner ist.
- ds: decide ist das n-te Bit gesetzt oder nicht
- cf: cont-fraction n-tes Folgenglied

### 3.1 Binary Sequence

- make-node erzeugt den shared pointer auf das node Objekt
- DAG (directed acyclic graph) als Struktur für Operatoren

## 4 Vorlesung

Programmierung

## 5 Vorlesung

(2)  $\Rightarrow$  (1)

Umsetzung von Approximation zur Binärfolge für die gesuchte Zahl x:

- Bereich zwischen 2 ganzen Zahlen aproximieren (ist x eine 2er-Potenz, schlägt dieser Schritt fehl). Fallunterscheidung:
  - Ist die Zahl ein endlicher Binärbruch schreibe diesen auf
  - ,sonst appoximiere und schreibe dann den endlichen Binärbruch
- Binärsequenzen eignen sich nicht zum Rechnen

$\mathbb{R}_c$  ist ein Körper

- Sind 2 Zahlen berechenbar, so auch das Ergebnis aus  $+$   $-$   $*$   $/$   $\Rightarrow$  gilt für Intervallschachtelungen (Lemma 3.8)
  - $+$  : untere/obere Grenze addieren
  - $-$  : untere/obere Grenze subtrahieren
  - $*$  ,  $/$  : min und max des Kreuzproduktes
- Ein Polynom mit berechenbaren Koeffizienten hat berechenbare Nullstellen

## 6 Vorlesung

### DAG

Interne Datenstruktur der Zahlen

- Auswertung der Zahlenwerte nur bei Bedarf (lazy eval)
- Bei einer Berechnung wird ein neuer "Rechenknoten" mit Pointer auf die Variable erstellt
  - Ein Knoten pro Operation (sehr Speicherintensiv)
  - Lösung: Komplexere Rechenknoten

### Berechenbare reelle Folgen

#### Berechenbarkeit einer Folge

- Berechenbare Folge berechenbarer Zahlen
- Das  $n$ -te Folgenglied der Folge kann mit Fehler  $2^{-i}$  durch eine berechenbare Folge rationaler Zahlen approximiert werden
- Nicht alle reellen Zahlen können durch eine berechenbare reelle Folge berechnet werden
  - Wähle eine rationale Folge  $q_n$ , die  $x_n$  approximiert
  - Wähle  $x_n$  so, dass es außerhalb dem approximierten Bereich von  $q_n$  liegt (Diagonalisierung)

## 7 General Stuff

- Abzählbar unendlich, wenn Bijektion zu  $\mathbb{N} \Rightarrow$  So viele berechenbare reelle Zahlen, wie Programme
- Entscheidbar:
- Berechenbar:
- Diagonalisierung

## 8 1. Übung

### Aufgabe 1

- Ziel: Finden des richtigen  $n$  für den Fehler
- Die Größe des Unterschieds zwischen  $x$  und  $y$  muss größer sein, als die Summe der Fehler
- Gleichheit testen geht nicht mit einer totalen Funktion

## Aufgabe 2

(4)  $\Rightarrow$  (3)

- Menge der kleineren Zahlen ist entscheidbar
- Durchtesten der Integers ob die Zahlen innerhalb oder außerhalb der Menge liegen
- Aus der Entscheidbarkeit der Menge werden die Folgen für die Schranken

(3)  $\Rightarrow$  (2)

- Differenz zwischen den Schranken ergibt Fehlergröße
- Folge  $q$  ist die Folge, die sich aus der Mitte  $\frac{a+b}{2}$

(2)  $\Rightarrow$  (3)

- Schranken  $a, b$  ergeben sich aus Folge  $+/-$  Fehler
- $a_k = \max(q_k + 2^{-k} - \frac{1}{k}, a_k - 1)$
- $b_k = \min(q_k - 2^{-k} - \frac{1}{k}, a_k - 1)$

(2/3)  $\Rightarrow$  (4)

Zusätzlicher Test, wenn die Zahl rational ist, weil der Test auf Gleichheit eine Endlosschleife

## Aufgabe 3

$x_A bb \Rightarrow A_{entscheidbar}$

Tablemakers dilemma