

# Berechenbare Analysis

## SoSe 19

Benedikt Lücken-Winkels

April 18, 2019

### Contents

<b>1</b>	<b>1. Vorlesung</b>	<b>2</b>
<b>2</b>	<b>2. Vorlesung</b>	<b>2</b>
2.1	Berechenbarkeit . . . . .	2
2.2	Entscheidbarkeit . . . . .	2
2.3	Berechenbare Reelle Zahlen . . . . .	2
<b>3</b>	<b>3. Vorlesung</b>	<b>3</b>
3.1	Binary Sequence . . . . .	4

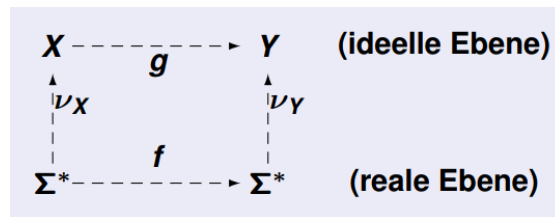
## 1 1. Vorlesung

## 2 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
...	...
<hr/>	
$x_\infty$	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.  $z \in \mathbb{Z} \ A \subseteq \mathbb{N}, x_A = \sum i \in A 2^{-1-i}, x = z + x_A$
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 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 achiclic graph) als Stuktur für Operatoren