1.3 Ist jede Funktion berechenbar?

Die Methode der Diagonalisierung

Inhalt

- Klärung der Frage vom Beginn, ob jedes Problem durch einen Computer lösbar ist
- um diese Frage zu beantworten – Technik der Diagonalisierung kennenlernen
- Anwendung der Diagonalisierung in 2 Varianten:
  1. um zu zeigen, dass es nicht-berechenbare Funktionen geben muss
  2. um eine konkrete nicht-berechenbare Funktion zu finden
- dazu benötigen wir 2 Werkzeuge: die Begriffe der Abzählbarkeit und Aufzählbarkeit

Die Existenz nicht-berechenbarer Funktionen

Abzählbarkeit, Überabzählbarkeit

- wir wollen zeigen, dass es weit mehr Funktionen gibt als Algorithmen
- dazu: Einführung des Konzeptes verschiedener Arten von Unendlichkeit

Definition 1.13 (Abzählbarkeit, Überabzählbarkeit)

Eine Menge M heißt **abzählbar**, falls

- $M = \emptyset$ oder
- es eine surjektive Funktion $f\colon \mathbb{N} \to M$ gibt.

Wir nennen M **überabzählbar**, falls M *nicht abzählbar* ist.

- informell: eine Menge ist abzählbar, wenn es eine Darstellung der Menge als (unendliche) Liste gibt
- anders ausgedrückt: eine Menge ist abzählbar, wenn sie höchstens so groß ist wie die natürlichen Zahlen (diese Einsicht führt zu äquivalenten Definitionen der Abzählbarkeit als Bijektion oder Injektion auf die natürlichen Zahlen)

Nützliche Eigenschaften:

Lemma 1.14

Die folgenden Aussagen sind für eine Menge M äquivalent:

(a) *M ist abzählbar*

(b) *es gibt eine injektive Funktion $f\colon M \to \mathbb{N}$*

(c) *M ist endlich oder es gibt eine bijektive Funktion $f\colon M \to \mathbb{N}$*

Beweis. Übung

Lemma 1.15

Wenn M abzählbar ist, dann ist auch $N \subseteq M$ abzählbar.

Beweis. Übung

Beispiele abzählbarer Mengen:

- $\mathbb{Z}$ ist abzählbar durch $f(n) := \begin{cases} -\frac{n}{2}, & \text{falls } n \text{ gerade;} \\ \frac{n+1}{2}, & \text{sonst.} \end{cases}$
- $\mathbb{N} \times \mathbb{N}$ ist abzählbar durch die bijektive Funktion $\mathrm{pc}\colon \mathbb{N}^2 \to \mathbb{N}$ mit:

$$\mathrm{pc}(x, y) := y + \sum_{i=0}^{x+y} i.$$

pc	0	1	2	3	4 $\cdots$	y
0	0	2	5	9	14	
1	1	4	8	13		
2	3	7	12	$\cdots$		
3	6	11				
4	10					
$\vdots$						
x						

  - pc ist die sogenannte Cantorsche Paarkodierungsfunktion
  - beachte: hier wurde Lemma 1.14 ausgenutzt
  - Übung: Wie können die 2 Umkehrfunktionen definiert werden, die eine natürliche Zahl auf die jeweilige x bzw. y-Komponente des Paares abbilden, die sie codiert?

Theorem 1.16

Sei Σ ein Alphabet, dann ist Σ^* abzählbar.

Beweis. Folgt sofort aus der Codierung von Zeichenketten als Zahlen auf Folie 47.

- Wiederholung: Sei $\Sigma = \{x_1, \dots, x_n\}$ und $b = n + 1$, dann ist die Abzählung $f\colon \Sigma^* \to \mathbb{N}$ für alle $w = x_{i_1} x_{i_2} \dots x_{i_s}$ über Σ wie folgt definiert:

$$f(w) := \begin{cases} 0, & \text{falls } w = \varepsilon; \\ (i_1 \dots i_s)_b, & \text{sonst.} \end{cases}$$

  mit $(i_1 \dots i_s)_b := \sum_{j=1}^{s} i_j \cdot b^{s-j}$

- es folgt, dass f injektiv ist

Übung: Konstruieren Sie eine surjektive bzw. bijektive Abzählung aller Wörter aus Σ^*.

Beispiele überabzählbarer Mengen:

Theorem 1.17

Die Menge aller (partiellen) Funktionen $f : \mathbb{N} \to \mathbb{N}$ ist überabzählbar.

- wir zeigen eine stärkere Aussage: Bereits die Menge aller totalen Funktionen $f : \mathbb{N} \to \{0, 1\}$ ist überabzählbar. Ferner nutzen wir dann Lemma 1.15, um die Aussage dieses Lemmas zu folgern.
- dazu nutzen wir die **Technik der Diagonalisierung**
- im speziellen *Georg Cantors* zweites Diagonalenargument (das erste wurde in Paarcodierungsfunktion verwendet)

Beweis des Theorems 1.17 durch Widerspruch:

Annahme: Die Menge aller totalen Funktionen $f\colon \mathbb{N} \to \{0,1\}$ ist abzählbar.

$\Rightarrow$ es gibt surjektive Abbildung $g\colon \mathbb{N} \to \{f \mid f\colon \mathbb{N} \to \{0,1\}\}$

Gibt es eine solche Abzählung(**!**) g, dann können wir die Funktionen

- nummerieren mit $g(i) = f_i$ $(i \in \mathbb{N})$
- und damit samt ihrer Funktionswerte auflisten:

	0	1	2	$\cdots$
f_0	$f_0(0)$	$f_0(1)$	$f_0(2)$	
f_1	$f_1(0)$	$f_1(1)$	$f_1(2)$	
f_2	$f_2(0)$	$f_2(1)$	$f_2(2)$	
$\vdots$				$\ddots$

Beweis (fortgesetzt): **Erste Anwendung des Diagonalenarguments**

Idee: Können wir eine totale Funktion $h\colon \mathbb{N} \to \{0,1\}$ finden, die in der Abzählung nicht vorkommt, kann es eine solche Abzählung nicht geben. (Warum nicht?)

$$\begin{array}{c|cccc} & 0 & 1 & 2 & \cdots \\ \hline f_0 & f_0(0) & f_0(1) & f_0(2) & \\ f_1 & f_1(0) & f_1(1) & f_1(2) & \\ f_2 & f_2(0) & f_2(1) & f_2(2) & \\ \vdots & & & & \ddots \end{array}$$

Definiere neue Funktion $h\colon \mathbb{N} \to \{0,1\}$ mit

$$h(n) := \begin{cases} 1, & \text{falls } f_n(n) = 0; \\ 0, & \text{falls } f_n(n) = 1; \end{cases}$$

- da $h \in \{f \mid f : \mathbb{N} \to \{0,1\}\}$, muss es auch in der Abzählung vorkommen, d.h. es gibt ein j so, dass $h = f_j$
- also gilt: $h(n) = f_j(n)$ für alle $n \in \mathbb{N}$ und insbesondere auch $f_j(j) = h(j)$
- nach Definition von h gilt aber: $\begin{array}{lcl} f_j(j) = 1 & \Rightarrow & h(j) = 0 \\ f_j(j) = 0 & \Rightarrow & h(j) = 1 \end{array}$
- **Widerspruch!** zu $f_j(j) = h(j)$ und damit zur Annahme, dass $\{f \mid f : \mathbb{N} \to \{0,1\}\}$ abzählbar ist
- aus $\{f \mid f : \mathbb{N} \to \{0,1\}\} \subseteq \{f \mid f : \mathbb{N} \to \mathbb{N}\}$ und der Kontraposition von Lemma 1.15 folgt dann die zu beweisende Aussage

Theorem 1.18

Folgende Mengen sind überabzählbar:

(a) *Die Menge aller Funktionen* $f\colon \Sigma^* \to \Sigma^*$ *für ein Alphabet* Σ.

(b) $\mathbb{R}$

(c) $\mathcal{P}(\mathbb{N})$

(d) $\mathcal{P}(\Sigma^*)$

$\ldots$

Beweisidee.

Cantors zweites Diagonalenargument kann hier genauso ausgenutzt werden, wie im Beweis von Theorem 1.17 gesehen. Aussage (a) kann auch aus dem Zusammenhang zwischen Zeichenketten und natürlichen Zahlen gefolgert werden. (Übung)

Nur „wenige" berechenbare Funktionen

Theorem 1.19

Es gibt nicht-berechenbare Funktionen.

Beweis. Die Aussage folgt aus Theorem 1.17 und dem folgenden Lemma:

Lemma 1.20

Die Menge aller berechenbaren (partiellen) Funktionen ist abzählbar.

Beweis Lemma 1.20. Nach der Church-Turing-These genügt es zu zeigen, dass die Menge aller Instanzen eines Berechnungsmodells (miniPy, TM, ...) abzählbar ist. Wir zeigen hier, dass die Menge aller miniPy-Programme abzählbar ist.

- aus Theorem 1.16 und Lemma 1.15 folgt: jede Sprache aus Σ^* (für ein Alphabet Σ) ist abzählbar
- aus Definition 1.1 (Syntax miniPy) folgt: die Menge aller miniPy-Programme ist eine Sprache L_{miniPy} über einem Alphabet Σ (Übung: Wie ist Σ definiert?)
- damit ist gezeigt, dass die L_{miniPy} abzählbar ist und damit auch $\mathbb{F}_{\text{ber}}$

Beweis zu Theorem 1.19 (cont.).

- Theorem 1.17 und Lemma 1.20 besagen nun: es gibt überabzählbar viele Funktionen aber nur abzählbar viele berechenbare
- daraus folgt: es muss nicht-berechenbare Funktionen geben
- Damit ist das Theorem bewiesen.

- aus Theorem 1.17 folgt weiterhin, dass die überwiegende Mehrheit der Funktionen nicht-berechenbar ist
- obiges Theorem 1.19 sagt nur die Existenz solcher voraus
- im Folgenden: konkrete nicht-berechenbare Funktionen