

(Master) Berechenbarkeit

## **v4.0.5.3.4 Entscheidbar, rekursiv aufzählbar**

**Kategory GmbH & Co. KG**

Präsentiert von Jörg Kunze

Copyright (C) 2024 Kategory GmbH & Co. KG

## BESCHREIBUNG

**Inhalt.** Entscheidbar ist eine Teilmenge der Menge der natürlichen Zahlen, wenn es eine totale  $\mu$ -rekursive Funktion gibt, die 1 ausgibt, wenn der Eingabewert in der Menge ist, und 0, wenn er es nicht ist.

Es ist schon ein kleiner Skandal, dass wir uns die Frage stellen, ob es überhaupt andere Mengen gibt. Gibt es Mengen, zu denen wir, egal was wir anstellen, keinen Algorithmus finden, mit dem entscheiden können, ob eine Zahl enthalten ist oder nicht?

Und wenn nicht, was bedeutet dann für diese Menge dieses „geben“ in „es gibt eine Menge ...“?

Eine andere Art von Mengen, sind die, die wir mit Hilfe eines Programms nach und nach auflisten können. Daraus folgt nicht, dass wir für eine gegebene Zahl, z. B. 6354981034340343034891784, entscheiden können, ob sie dazugehört.

Eine dritte Art sind Mengen, die wir mithilfe eines Prädikats aus ZFC, also mit einer Formel definieren können: die definierbaren Mengen.

Es wird sich herausstellen, dass entscheidbar  $\subsetneq$  listbar  $\subsetneq$  definierbar  $\subsetneq$  Teilmenge von  $\mathbb{N}$ . Intuitiv sind Mengen von links nach rechts immer weniger greifbar. Einzelne nicht-definierbare Mengen verschwinden im Nebel und sind nur durch ihre Wirkung im Rudel zu erkennen.

Das ein Programm eine Endlosschleife hat, ist normal. Dass es aber Funktionen gibt, zu denen es kein Programm ohne Endlosschleifen gibt, ist ein wenig unerträglich.

Beispiele: entscheidbar:  $p$  ist prim, „...“ ist eine Aussage listbar: Das Programm  $P$  hält. „...“ ist beweisbar definierbar: Das Programm  $P$  hält nicht. „...“ ist nicht beweisbar Teilmenge: indirekt nachgewiesen dadurch, dass es abzählbare viele Definitionen aber überabzählbar viele Teilmengen gibt.

Entscheidbare Mengen sind Urbilder, Listbare sind die Bilder von totalen  $\mu$ -rekursiven Funktionen. Listbare sind darüber hinaus die Definitionsbereiche von partiellen  $\mu$ -rekursiven Funktionen.

**Präsentiert.** Von Jörg Kunze

**Voraussetzungen.** Berechenbare Funktionen, Alphabet

**Text.** Der Begleittext als PDF und als LaTeX findet sich unter <https://github.com/kategory/kategoryMathematik/tree/main/v4%20Master/v4.0%20Berechenbarkeit/v4.0.5.3.4%20Entscheidbar%2C%20rekursiv%20aufz%C3%A4hlbare>

**Meine Videos.** Siehe auch in den folgenden Videos:

v4.0.2 (Master) Berechenbarkeit - Alphabete und Sprachen  
<https://youtu.be/cw1U8m9ldbA>

v4.0.4 (Master) Berechenbarkeit - Berechenbare Funktionen  
<https://youtu.be/tARmHFIP32o>

**Quellen.** Siehe auch in den folgenden Seiten:

<https://de.wikipedia.org/wiki/Entscheidbarkeit>  
[https://de.wikipedia.org/wiki/Rekursiv\\_aufz%C3%A4hlbare\\_Menge](https://de.wikipedia.org/wiki/Rekursiv_aufz%C3%A4hlbare_Menge)  
<https://de.wikipedia.org/wiki/Halteproblem>  
[https://de.wikipedia.org/wiki/Universelle\\_Turingmaschine](https://de.wikipedia.org/wiki/Universelle_Turingmaschine)  
<https://de.wikipedia.org/wiki/Entscheidbarkeit>  
[https://en.wikipedia.org/wiki/Definable\\_set](https://en.wikipedia.org/wiki/Definable_set)  
[https://de.wikipedia.org/wiki/Cantors\\_zweites\\_Diagonalargument](https://de.wikipedia.org/wiki/Cantors_zweites_Diagonalargument)  
[https://de.wikipedia.org/wiki/Cantors\\_erstes\\_Diagonalargument](https://de.wikipedia.org/wiki/Cantors_erstes_Diagonalargument)

**Buch.** Grundlage ist folgendes Buch:

Computability

A Mathematical Sketchbook

Douglas S. Bridges

Springer-Verlag New York Inc. 2013

978-1-4612-6925-0 (ISBN)

**Lizenz.** Dieser Text und das Video sind freie Software. Sie können es unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation veröffentlicht, weitergeben und/oder modifizieren, entweder gemäß Version 3 der Lizenz oder (nach Ihrer Option) jeder späteren Version.

Die Veröffentlichung von Text und Video erfolgt in der Hoffnung, dass es Ihnen von Nutzen sein wird, aber OHNE IRGENDNEINE GARANTIE, sogar ohne die implizite Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK. Details finden Sie in der GNU General Public License.

Sie sollten ein Exemplar der GNU General Public License zusammen mit diesem Text erhalten haben (zu finden im selben Git-Projekt). Falls nicht, siehe <http://www.gnu.org/licenses/>.

**Das Video.** Das Video hierzu ist zu finden unter [hhh](#)

#### 1. v4.0.5.3.4 ENTSCHEIDBAR, REKURSIV AUFGÄHLLBAR

**1.1. Entscheidbar.** Entscheidbar ist eine Teilmenge der Menge der natürlichen Zahlen, wenn es eine totale (also normale, nicht partielle, auf allen natürlichen Zahlen definierte)  $\mu$ -rekursive Funktion gibt, die 1 ausgibt, wenn der Eingabewert in der Menge ist, und 0, wenn er es nicht ist. Mit anderen Worten, ist eine Menge entscheidbar, wenn ihre charakteristische Funktion total und berechenbar ist. Noch anders: wir können mit Hilfe eines Computerprogramms entscheiden, ob ein Wert drinne ist oder nicht.

Eine entscheidbare Menge ist das Urbild einer einelementigen Menge unter einer totalen Funktion.

Entscheidbar: die totale berechenbare Funktion muss nicht *primitiv* rekursiv sein. Die Ackermann-Funktion ist ein Beispiel einer totalen berechenbaren nicht primitiv-rekursiven Funktion. Wir könnten also z. B. Ackermann einbauen in eine Funktion die 1 ist, wenn Ackermann eine durch 17 teilbare Zahl liefert und 0 sonst.

**1.2. Listbar.** Eine Menge ist rekursiv aufzählbar (ich hätte „listbar“ genommen), wenn wir sie mit Hilfe eines Programms nach und nach auflisten können, z. B. in einer Schleife.

Präziser ist eine listbare Menge die leere Menge oder das Bild einer totalen Funktion  $f$ . Für die eben angedeutete Schleife rufen wir dann  $f$  nacheinander für die Werte  $0, 1, 2, \dots$  auf und geben den Wert aus.

```
for( let i = 0; ; ++i ) console.log( f(i) );
```

Die leere Menge nehmen wir mit rein, weil sie zwar nicht das Bild einer totalen Funktion sein kann aber das Urbild einer endlichen Menge. Und wir wollen, dass alle entscheidbaren Mengen auch listbar sind. Ein Programm, welches die leere Menge auflistet, ist z. B.

```
return;
```

weil hier kein „console.log(...)“ stattfindet.

**1.3. Entscheidbar vs listbar.** Jede entscheidbare Menge ist listbar. Beweis: Sei  $f$  die totale  $\mu$ -rekursive charakteristische Funktion der Menge. Falls die Menge leer ist sind wir fertig. Falls nicht sei  $m$  Element der Menge. Dann definieren wir  $g(n) = n$ , falls  $f(m) = 1$  und  $g(n) = m$  sonst.

```
function( n ) {
  let m = 0;
  for( ; !f(m); ++m ); // Schleife endet, da die Menge nicht leer ist

  return f(n) ? n : m;
```

```
}

```

Gibt es listbare Mengen, die gleichzeitig nicht entscheidbar sind? Ja! Wir werden in diesem Kurs erkennen, dass die Menge der haltenden Turingmaschinen nicht entscheidbar.

Statt  $\mathbb{N}$  können wir, wie immer, auch die Menge der endlichen (Zeichen-) Ketten betrachten. Dort wollen wir, dass die Mengen der Terme, Formeln, Aussagen und Beweise entscheidbar sind. In der mathematischen Logik wird gezeigt werden, dass die Menge der beweisbaren Aussagen nicht entscheidbar aber immerhin listbar ist. Alan Turing und Alonzo Church haben für 1936 festgestellt, dass die Menge der beweisbaren Aussagen nicht entscheidbar ist.

Um einmal ein „normales“ Thema zu nehmen: Die Menge der Diophantischen Gleichungen ist nicht entscheidbar.

**1.4. Listbar = Definitionsbereich einer  $\mu$ -rekursiven Funktion.** Sei  $M$  listbar. Falls  $M$  leer, nehmen wir als  $f$  die Funktion, die nirgendwo definiert ist:

```
function( n ) {
  while( true );
}
```

Falls  $M$  nicht leer, geben wir 1 zurück, wenn der Eingabewert gelistet wird:

```
function( n ) {
  for( let i = 0; f(i) != n; ++i );
  return 1;
}
```

Sei nun  $M$  der Definitionsbereich einer  $\mu$ -rekursiven Funktion  $g$ . Falls ein  $n$  nicht im Definitionsbereich liegt, können wir in unserem zu konstruierenden  $f$  nicht  $g(n)$  aufrufen, da wir dann in einer Endlosschleife landen. Wir gehen statt dessen so vor, dass wir  $g$  nach einander für alle Zahlen als Eingabe Stück für Stück ausführen und dabei schauen, ob das Programm zum Halten kommt. Wenn ja, geben wir den Wert aus.

Sei dazu  $h_g(n, k)$  ein Programm, welches  $g$ , genauer ein Programm, welches  $g$  implementiert, mit der Eingabe  $n$  für  $k$  Schritte laufen lässt und dann 1 zurückgibt, wenn das Programm mit der Eingabe nach  $\leq k$  Schritten zum Halten gekommen ist, 0 sonst. Dieses Programm nennen wir **beschränkte Haltefunktion**. Dann berechnen wir nacheinander  $h_f$  auf den Paaren  $(n, k)$  in  $(0, 1), (1, 1), (0, 2), (2, 1), (1, 2), (0, 3), (3, 1), (2, 2), \dots$  an. Immer, wenn 1 rauskommt, geben wir  $n$  aus.

Das ist ein Durchschlängeln, mit dem wir auch alle Brüche aufzählen können. Übrigens ist die Menge aller Brüche Listbar. Siehe Cantors erstes Diagonalargument.

Dieses  $h$  ist gar nicht so leicht zu bekommen: Ein Weg dieses  $h$  zu bekommen, ist das ursprüngliche Programm umzuschreiben, so dass es nur eine Schleife enthält (ja das geht) und diese nach  $k$  Durchläufen abbricht.

Ein zweiter Weg ist sich einen JavaScript-Interpreter in JavaScript zu schreiben, mit dem das Programm für  $g$  Schritt für Schritt ausgeführt wird. Dieser Interpreter bekommt als Parameter das ursprüngliche Programm, die Eingabe und unser  $k$ . Dies ist die universelle Turingmaschine, die wir in einem der nächsten Videos kennenlernen.

**1.5. Nicht listbare Mengen.** Falls eine Menge  $M$  und ihr Komplement  $\overline{M}$  listbar sind, so ist die Menge sogar entscheidbar.

Beweis: Seien  $f, g$   $\mu$ -rekursive Funktionen, die Mengen  $M, \overline{M}$  listen. Für ein gegebenes  $n$  berechnen wir, ähnlich wie mit dem  $h$  oben, schrittweise und nach und nach für alle  $i$  die Werte  $f(i)$  und  $g(i)$ . Egal, ob  $n \in M$  oder  $n \in \overline{M}$ , wird es irgendwann auftauchen und wir wissen dann, ob es in  $M$  ist.

```
function( n ) {
  for( let i = 0;; ++i ) {
    if( f(i) == n ) return 1;
    if( g(i) == n ) return 0;
  }
}
```

---

Da es nicht entscheidbare Mengen gibt, gibt es automatisch auch nicht listbare: Die Menge aller nicht haltenden Programme. Die Menge aller nicht beweisbaren Aussagen.

**1.6. Definierbar.** Eine Menge ist definierbar, wenn sie durch ein Prädikat von ZFC definiert werden kann. Die Komplemente nicht listbarer Mengen sind offensichtlich definierbar: Die Menge aller nicht haltenden Programme. Die Menge aller nicht beweisbaren Aussagen.

**1.7. Nicht-definierbar.** Dadurch, dass es abzählbare viele Definitionen aber überabzählbar viele Teilmengen gibt (siehe ), muss es auch sehr viele nicht-definierbare Mengen geben. Siehe Cantors zweites Diagonalargument.

**1.8. Die Hierarchie der Greifbarkeit.** Also:

entscheidbar  $\subsetneq$  listbar  $\subsetneq$  definierbar  $\subsetneq$  Teilmenge von  $\mathbb{N}$ .

Intuitiv sind Mengen von links nach rechts immer weniger greifbar. Einzelne nicht-definierbare Mengen verschwinden im Nebel und sind nur durch ihre Wirkung im Rudel zu erkennen. Auf diese Weise wirken sie allerdings mit Macht.

Gefühlt haben wir entscheidbare Mengen im Griff. Listbare Mengen haben wir ein wenig im Griff. Immerhin können wir alle Elemente nach und nach generieren. Nicht listbare Mengen sind dagegen mysteriös. Wir können vielleicht für einzelne Elemente beweisen, dass sie dazugehören oder nicht, aber das war es dann auch schon.

#### LITERATUR

[Douglas2013] Douglas S. Bridges, *Computability, A Mathematical Sketchbook*, Springer, Berlin Heidelberg New York 2013, ISBN 978-1-4612-6925-0 (ISBN).

#### SYMBOLVERZEICHNIS

$\mathbb{N}$	Die Menge der natürlichen Zahlen (mit Null): $\{0, 1, 2, 3, \dots\}$
$n, k, l, s, i, x_i, q$	Natürliche Zahlen
$A(k, n)$	Ackermannfunktion
$f, g, h$	Funktionen
$s()$	Nachfolgerfunktion: $s(n) := n + 1$