# Berechnung von Mengen

Hans Jürgen Ohlbach

**Keywords:** Entscheidbarkeit, Semi-Entscheidbarkeit, charakteristische Funktion, abzählbare Mengen, rekursiv aufzählbare Mengen

**Empfohlene Vorkenntnisse:** Elementare Mengenlehre, Funktionen und Relationen

In diesem Text geht es in erster Linie um unendliche Mengen, die nicht explizit gegeben sind, sondern implizit durch Konstruktionsvorschriften spezifiziert sind. Beispiele sind die Wörter einer formalen Sprache, wie z.B. die Menge der syntaktisch korrekten Programme einer gegebenen Programmiersprache.

## 1 Entscheidbare Mengen

Für jede Menge, egal ob endlich oder unendlich, liegt fest, welche Elemente dazu gehören, und welche nicht. Es ist aber nicht immer einfach, oder gar manchmal unmöglich, die Zugehörigkeit zu der Menge algorithmisch festzustellen. Bei der Menge der syntaktisch korrekten Programme einer Programmiersprache, z.B. benötigt man dazu einen Parser, der die Programmtexte analysiert.

Man definiert zunächst:

**Definition 1 (Charakteristische Funktion)**
*Die charakteristische Funktion $\chi_A$ einer Menge $A$ ist definiert durch:*

$$\chi_A(x) = \begin{cases} 1 & \textit{falls } x \in A \\ 0 & \textit{falls } x \notin A \end{cases}$$

Diese Funktion existiert immer, und liefert immer ein Ergebnis. Sie muss aber nicht immer *berechenbar sein*. (Eine Funktion ist *berechenbar*, wenn es eine Turingmaschine gibt (oder ein Programm einer höheren Programmiersprache), die für alle Argumentwerte terminiert, und den richtigen Wert liefert).

Daher definiert man weiter:

**Definition 2 (Entscheidbare / Semi-entscheidbare Mengen)**
*Eine Menge $A$ heißt*

**entscheidbar** *falls $\chi_A$ berechenbar ist*

**semi-entscheidbar** *falls die partielle Funktion*

$$\chi'_A(x) = \begin{cases} 1 & \textit{falls } x \in A \\ \textit{undefiniert} & \textit{falls } x \notin A \end{cases}$$

*berechenbar ist. Der Fall* undefiniert *äußert sich i.A. bei der Berechnung von $\chi'_A$ dadurch, dass das Programm dafür nicht terminiert.*

**Beispiele:** Die Menge der Primzahlen ist entscheidbar. In diesem Fall wird $\chi_A$ durch einen Primzahltest implementiert.

Die Menge der syntaktisch korrekten Java Programme ist entscheidbar. In diesem Fall wird $\chi_A$ durch einen Java Parser implementiert.

Die Menge der syntaktisch korrekten *und terminierenden* Java Programme ist nur semi-entscheidbar. In diesem Fall wird $\chi'_A$ dadurch implementiert, dass man das Programm laufen lässt. Falls es terminiert, dann liefert $\chi'_A$ die 1, ansonsten terminiert es eben nicht. Dass diese Menge tatsächlich nicht entscheidbar ist, folgt aus der Unentscheidbarkeit des Halteproblems.

Die Menge der syntaktisch falschen oder *nicht terminierenden* Java Programme ist noch nicht einmal semi-entscheidbar. (Dies folgt aus einem allgemeinen Resultat, welches besagt, dass das Komplement von Typ 0 Sprachen nicht mehr Typ 0 sein muss.)

Ein Zusammenhang zwischen entscheidbar und semi-entscheidbar ergibt sich aus dem nächsten Theorem.

**Theorem 1** *Eine Menge $A$ ist genau dann entscheidbar, wenn sowohl $A$ als auch das Komplement von $A$ semi-entscheidbar sind.*

Die interessante Richtung des Beweises ist die von rechts nach links. Falls $A$ als auch das Komplement $A'$ von $A$ semi-entscheidbar sind, kann man die Berechnungsverfahren für $\chi'_A$ und $\chi'_{A'}$ parallel laufen lassen (z.B. durch zwei getrennte Threads). Eines der beiden Verfahren wird auf jeden Fall nach endlicher Zeit terminieren, und bestimmt damit das Endergebnis. Das zweite Verfahren kann man dann abbrechen.

## 2 Reihenfolgen von Mengenelementen

Viele Mengen, insbesondere auch unendliche Mengen, lassen sich in einer Reihenfolge anordnen. Die Primzahlen, z.B. lassen sich nach ihrer Größe anordnen. Die Frage ist allerdings, ob sich diese Reihenfolge immer berechnen lässt. Nicht überraschend ist: einige Reihenfolgen lassen sich berechnen, und einige Reihenfolgen lassen sich nicht berechnen.

Man bezeichnet alle Mengen, bei denen es eine Reihenfolge gibt, als *abzählbar*, und wenn sich die Reihenfolge auch noch berechnen lässt, als *(rekursiv) aufzählbar*.

**Definition 3 (Abzählbare Mengen)** *Eine Menge $M$ ist* abzählbar *wenn man Sie mit natürlichen Zahlen indiziert aneinanderreihen kann: $M = \{x_0, x_1, x_2, \ldots\}$.*

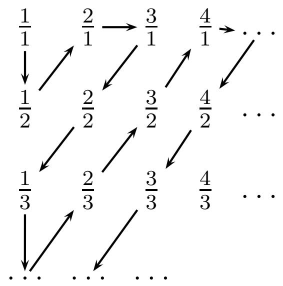
**Beispiele:**

- die Menge der Primzahlen ist abzählbar.
- die Menge der Zeichenketten (Strings) über einem Alphabet ist abzählbar. Man verwendet die lexikographische Ordnung über Zeichenketten.
- die Menge der syntaktisch korrekten Java-Programme ist abzählbar. Man verwendet die lexikographische Ordnung auf dem Programmtext.
- die Menge der *terminierenden* Java-Programme ist abzählbar. Man nimmt die Reihenfolge der syntaktisch korrekten Programme wie oben, und lässt die nicht-terminierenden Programme weg.
- die Menge der *nicht terminierenden* Java-Programme ist abzählbar. Man nimmt die Reihenfolge der syntaktisch korrekten Programme wie oben, und lässt die terminierenden Programme weg. (Diese Reihenfolge existiert zwar, lässt sich aber nicht berechnen.)

**Definition 4 ((Rekursiv) aufzählbare Mengen)** *Eine Menge $M$ ist* (rekursiv) aufzählbar *wenn es eine berechenbare Funktion $f$ gibt, mit $M = \{f(0), f(1), f(2), \ldots\}$.*

Jede aufzählbare Menge ist natürlich auch abzählbar, aber nicht umgekehrt. Aus der Unentscheidbarkeit des Halteproblems folgt, dass die Menge der nicht-terminierenden Programme zwar abzählbar ist (mit der lexikographischen Reihenfolge der Programmtexte), aber nicht rekursiv aufzählbar (man kann eben nicht feststellen ob ein Programm nicht terminiert). Die anderen, in der obigen Liste aufgeführten abzählbaren Mengen sind dagegen auch rekursiv aufzählbar.

Ein zur damaligen Zeit etwas überraschendes Resultat war, dass auch die rationalen Zahlen aufzählbar sind. Für die Aufzählung braucht man die Diagonalkonstruktion:



Doppelt vorkommende Zahlen können bei der Konstruktion übersprungen werden. Damit sind die rationalen Zahlen gleichmächtig zu den natürlichen Zahlen.

Das nächste Theorem stellt einen Zusammenhang zwischen rekursiver Aufzählbarkeit und Semi-Entscheidbarkeit her:

**Theorem 2** *Eine Teilmenge einer rekursiv aufzählbaren Menge ist selbst rekursiv aufzählbar genau dann wenn sie semi-entscheidbar ist.*

Im interessanten Teil des Beweises geht es um die Konstruktion der Aufzählungsfunktion $f$. Sei also eine Menge $A$ rekursiv aufzählbar, d.h. wir haben eine berechenbare Funktion $g$, so dass $A = \{g(0), g(1), \ldots\}$. Sei $M \subseteq A$ die Menge, die wir aufzählen wollen. Dafür haben wir ein Semi-Entscheidungsverfahren für $M$.

Der Trick ist, das Semi-Entscheidungsverfahren *parallel* für $g(0), g(1), \ldots$ zu starten (z.B. in eigenen Threads). Die Reihenfolge, in der diese Threads terminieren, bestimmt die Aufzählungsfunktion $f$. Es könnte z.B. $f(0) = Thread(g(5))$, $f(1) = Thread(g(3))$ usw. sein. Die Threads, wo das Semi-Entscheidungsverfahren nicht terminiert, werden einfach ignoriert.

Mit diesem Verfahren kann man z.B. die terminierenden Programme einer Programmiersprache aufzählen: Die rekursiv aufzählbare Basismenge wäre die lexikographische Reihenfolge der Unicode-Strings. Für jeden solchen String startet man einen Thread, der den String compiliert, und falls das gelingt, das compilierte Programm startet. Die Reihenfolge, in denen diese compilierten Programme terminieren, bestimmt die Aufzählung. Diese Reihenfolge richtet sich dann nach der Laufzeit, und ist insbesondere dann nicht mehr lexikographisch geordnet. Eine solche Ordnung kann auch nicht mehr hergestellt werden, da man zu keinem Zeitpunkt weiß, ob ein lexikographisch kleineres Programm nicht doch noch irgendwann später terminiert.

Für die nicht-terminierenden Programme funktioniert diese Technik leider nicht.

Ein Aufzählungsverfahren für eine Menge $M$ kann auf einfache Weise auch als Semi-Entscheidungsverfahren für die Zugehörigkeit zur Menge genutzt werden: Um $x \in M$ zu testen, zählt man die Elemente der Menge auf. Falls $x \in M$ gilt, taucht $x$ nach endlicher Zeit in der Aufzählung auf, und man beendet die Suche mit Erfolg. Falls $x \notin M$, terminiert das Verfahren leider nicht.

Folgende Aussagen für eine Menge $A$ sind dann äquivalent:

- $A$ ist rekursiv aufzählbar,
- $A$ ist Teilmenge einer rekursiv aufzählbaren Menge und semi-entscheidbar,
- die charakteristische Funktion $\chi'_A$ ist berechenbar,
- A ist der Definitionsbereich einer berechenbaren Funktion, nämlich der Funktion, die jedem Element seine Nummer zuordnet,
- A ist der Wertebereich einer berechenbaren Funktion, nämlich der Funktion, die die Elemente von $A$ aufzählt.

# Stichwortverzeichnis