MATHEMATISCHE LOGIK
BEDROHLICHE UNENTSCHEIDBARKEIT

Aussagen, die man prinzipiell weder beweisen noch widerlegen kann, sind keineswegs so esoterisch, wie es klingt. Man findet sie, indem man das Verhalten kleiner Turing-Maschinen untersucht.

Jean-Paul Delahaye ist emeritierter Professor der Université Lille I und Forscher am Centre de recherche en informatique, signal et automatique in Lille (CRISTAL).

▸▸ spektrum.de/artikel/1485141

Wenn ein mathematisches Problem einfach aussieht, wie die Frage »Ist $4n^4+1$ für alle natürlichen Zahlen eine Primzahl?«, dann ist der durchschnittliche Mathematiker überzeugt, es lösen zu können. Im Einzelfall werden unterschiedliche Auffassungen darüber bestehen, wie schwierig ein konkretes Problem ist; aber je besser sich ein Mathematiker in einem Teilgebiet auskennt, desto sicherer ist sein Urteil über den Schwierigkeitsgrad.

In manchen Fällen versagt allerdings die Erfahrung der fähigsten Fachvertreter. Pierre de Fermat (1607–1665) hat sich sicherlich nicht vorgestellt, dass die unschuldig aussehende Aussage »Es gibt keine Lösung der Gleichung $n^q + m^q = p^q$ mit natürlichen Zahlen n, m und p und einer natürlichen Zahl $q > 2$« seine wissenschaftlichen Nachkommen 350 Jahre lang beschäftigen würde. Hinter scheinbar einfachen Aussagen verbergen sich zuweilen äußerst hartnäckige Hindernisse.

Die schwierigsten Probleme sind diejenigen, die man als »unentscheidbar« bezeichnet. Und was tun die Mathematiker? Sie suchen nach den einfachsten unter den unentscheidbaren Problemen. Dieses Spiel ist keineswegs müßig; es hilft die wesentlichen Hindernisse genauer einzugrenzen, die einer Lösung oder eben der Entscheidbarkeit entgegenstehen. Das wiederum bringt die Problemlösefähigkeit voran – und damit die Mathematik im Allgemeinen.

Was genau bedeutet Unentscheidbarkeit? Eine Aussage ist unentscheidbar in einer gegebenen Theorie, wenn sie sich in dieser Theorie weder beweisen noch widerlegen lässt. Und diese doppelte Unmöglichkeit muss ihrerseits beweisbar sein. Damit das überhaupt möglich ist, muss man die Theorie sehr präzise definiert haben und sie mindestens so weit beherrschen, dass man in ihr Aussagen beweisen kann.

Solche Theorien bereitzustellen und zu erforschen, ist die Aufgabe der mathematischen Logik, die manchmal auch Metamathematik genannt wird, weil sie mit mathematischen Mitteln ein Verständnis der Mathematik liefert. Dank ihrer Fortschritte im 19. und 20. Jahrhundert können wir heute die »deduktive Mächtigkeit« einer Theorie genau bestimmen, das heißt die Klasse der Aussagen, die in ihr beweisbar sind.

Drei klassische unentscheidbare Aussagen haben eine gewisse Berühmtheit erlangt. Die erste ist das Parallelenaxiom: Durch einen Punkt außerhalb einer gegebenen

AUF EINEN BLICK
LÖCHER IM KOSMOS DER ABSTRAKTION

1 Nach den gödelschen Unvollständigkeitssätzen hat jede formale Theorie Lücken. Sie kann insbesondere ihre eigene Widerspruchsfreiheit nicht beweisen.

2 Bisher galten diese Lücken als theoretisch bedeutsam, aber für die praktische Arbeit irrelevant: Man erwartete nicht, dass der Beweis eines mathematischen Satzes an einer Unentscheidbarkeit scheitern würde.

3 Inzwischen weiß man: Bereits der Wert einer einfach definierten Funktion an der Stelle n=1919 ist wegen einer solchen Lücke prinzipiell unberechenbar.

Geraden g gibt es genau eine Gerade g', die g nicht schneidet. Es ist nicht möglich, diese Aussage oder ihre Negation aus den übrigen Axiomen der ebenen Geometrie herzuleiten. In der Tat: Nimmt man an Stelle des Parallelenaxioms eine der möglichen Verneinungen (»es gibt keine Gerade ...« oder »es gibt mehr als eine Gerade ...«) zur Geometrie der Ebene hinzu, so erhält man an Stelle der gewöhnlichen euklidischen Geometrie eine nichteuklidische, die ebenso widerspruchsfrei ist wie die gewohnte.

Die beiden anderen Aussagen beziehen sich auf die klassische Mengenlehre. Es handelt sich um das Auswahlaxiom, das unten näher erläutert wird, und die Kontinuumshypothese: Zwischen der Unendlichkeit der natürlichen Zahlen und derjenigen der reellen Zahlen gibt es keine weitere Unendlichkeit.

Es ist unmöglich, auf der Basis der üblichen Axiome der Mengenlehre die Kontinuumshypothese oder ihre Negation zu beweisen. Dasselbe gilt für das Auswahlaxiom. Den ersten Schritt zum Beweis dieser Ergebnisse leistete 1938 der Österreicher Kurt Gödel (1906–1978), den zweiten 1963 der US-Amerikaner Paul Cohen (1934–2007).

Noch allgemeiner gilt nach Gödels erstem Unvollständigkeitssatz von 1931: Ist eine mathematische Theorie T widerspruchsfrei und ausreichend mächtig, um die üblichen Sätze der Arithmetik zu formulieren (was für alle interessanten Theorien der Fall ist), dann gibt es Aussagen A, die in dieser Theorie sinnvoll, aber unentscheidbar sind. Anders gesagt: Weder die Aussage A noch ihre Negation (non-A) ist in T beweisbar. Die Theorie weist Lücken auf, sie ist unvollständig.

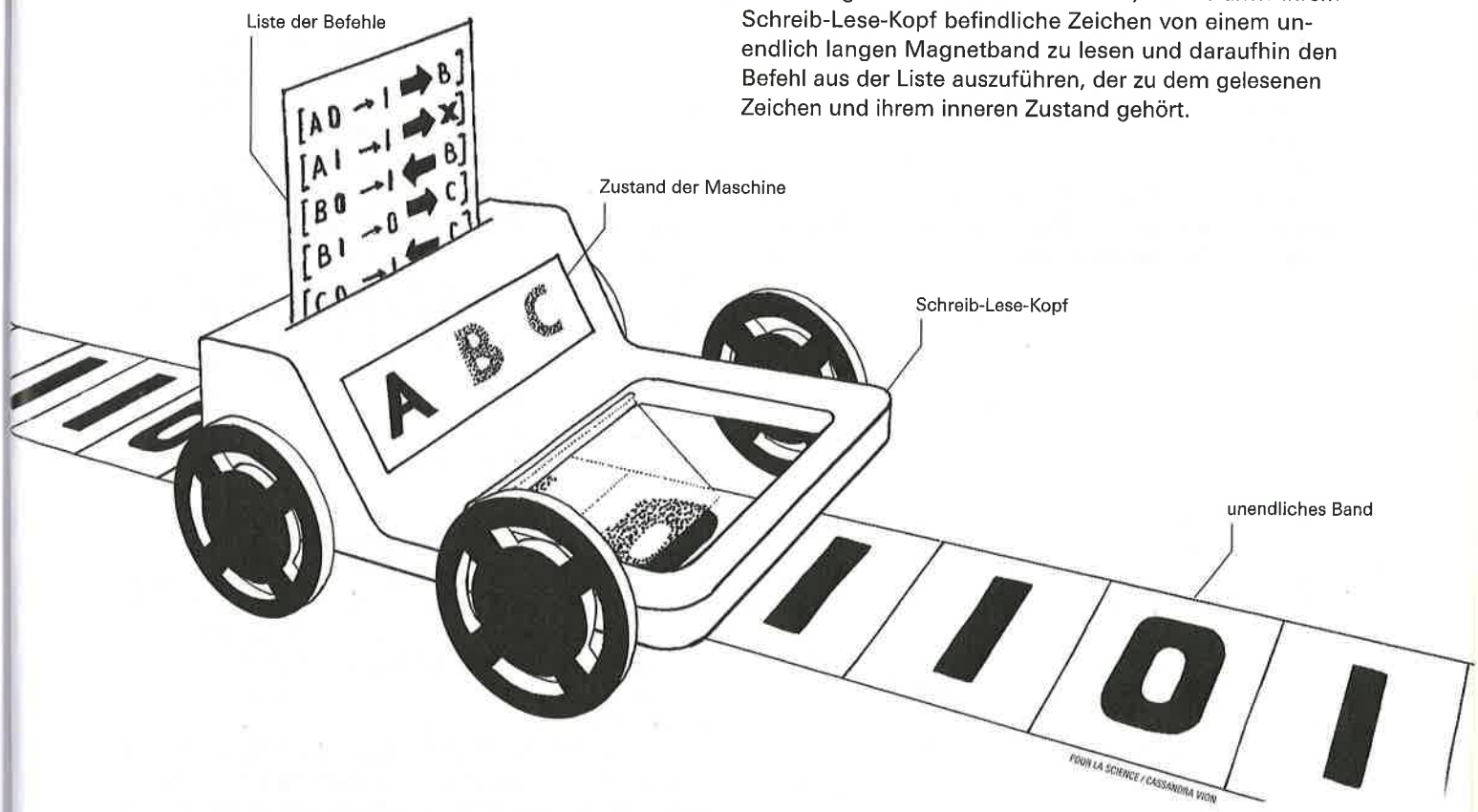
Der zweite Unvollständigkeitssatz von Gödel sagt aus, dass die Aussage »Die Theorie T ist widerspruchsfrei« in T unentscheidbar ist: Eine Theorie ist niemals stark genug, um ihre eigene Widerspruchsfreiheit zu beweisen. Wenn das Gegenteil zutrifft, die Theorie also Widersprüche enthält, ist das zwar im Allgemeinen nicht schwer zu beweisen; aber in diesem Fall wäre die Theorie vollkommen uninteressant.

Die Frage der Unentscheidbarkeit ist also eng mit derjenigen nach Widerspruchsfreiheit verknüpft. Und da eine Theorie mit ihrer Widerspruchsfreiheit steht oder fällt, ist diese Frage von zentraler Bedeutung.

Das Wort »widerspruchsfrei« ist leicht gesagt; aber im Rahmen einer formalen Theorie ist es überraschend kompliziert, den entsprechenden Sachverhalt auszudrücken. Eine häufig verwendete Formulierung lautet: »Keine Folge von Formeln, die man gemäß den Regeln der Theorie aufstellen kann, führt auf die Gleichung 0 = 1.« Um mit Aussagen dieser Art im Wortsinn rechnen zu können, überführt man sie in Aussagen über (natürliche) Zahlen

Schematische Darstellung einer Turing-Maschine

Eine Turing-Maschine tut nichts weiter, als das unter ihrem Schreib-Lese-Kopf befindliche Zeichen von einem unendlich langen Magnetband zu lesen und daraufhin den Befehl aus der Liste auszuführen, der zu dem gelesenen Zeichen und ihrem inneren Zustand gehört.



Der fleißige Biber mit drei Zuständen

Der fleißige Biber mit *n* Zuständen ist die Maschine, die unter den nicht anhaltenden Turing-Maschinen mit *n* Zuständen den Rekord hält, das heißt die maximal mögliche endliche Zahl an Schritten durchführt.

Das Programm des fleißigen Bibers mit drei Zuständen enthält sechs Befehle; es wurde von dem ungarischen Mathematiker Tibor Radó gefunden. Die Maschine macht 21 Schritte, bevor sie anhält, und schreibt dabei fünf Einsen aufs Band. Die sechs Befehle lauten:

[A 0 → 1 rechts B],
[A 1 → 1 rechts HALT],
[B 0 → 1 links B],
[B 1 → 0 rechts C],
[C 0 → 1 links C],
[C 1 → 1 links A].

Die Maschine (rechts schematisch dargestellt) wird auf ein Band gesetzt, das in beiden Richtungen (nach rechts und nach links) mit unendlich vielen Nullen beschrieben ist. Nehmen wir an, sie befinde sich im Zustand A, ihr Schreib-Lese-Kopf liest eine 0. Der anzuwendende Befehl ist folglich der erste [A 0 → 1 rechts B] in ihrem Programm. Die Maschine schreibt eine Eins, wandert ein Feld nach rechts und geht in den Zustand B über. Nehmen wir an, dass die Maschine zu Anfang auf dem vierten Feld sitzt, und kennzeichnen wir die Position des Schreib-Lese-Kopfs durch rote Farbe, so wird aus dem Anfangszustand des Systems

A 0000000000000000000...

der Zustand

B 0001000000000000000...

Die Maschine ist nun im Zustand B und liest eine Null. Sie wendet folglich Befehl Nummer 3 [B 0 → 1 links B] an, was folgende Situation ergibt:

B 0001100000000000000...

Die Berechnung verläuft nun folgendermaßen bis zum Anhalten:

A – 0000000000000000000000000...
B – 0001000000000000000000000...
B – 0001100000000000000000000...
C – 0000100000000000000000000...
A – 0000100000000000000000000...
B – 0001100000000000000000000...
C – 0001000000000000000000000...
C – 0001010000000000000000000...
C – 0001110000000000000000000...
A – 0001110000000000000000000...
B – 0011110000000000000000000...
C – 0010110000000000000000000...
A – 0010110000000000000000000...
B – 0011110000000000000000000...
C – 0011010000000000000000000...
A – 0011010000000000000000000...
B – 0011110000000000000000000...
C – 0011100000000000000000000...
C – 0011101000000000000000000...
C – 0011111000000000000000000...
A – 0011111000000000000000000...
HALT – 0011111000000000000000...

und die Regeln der Theorie in Rechenregeln. Diese »Arithmetisierung der Syntax« ist ein schwieriger und langwieriger Prozess, dessen Konzept wir ebenfalls Kurt Gödel verdanken.

Dass es überhaupt unentscheidbare Aussagen gibt, wissen wir, weil die Mathematiker eigens zu diesem Zweck und unter beträchtlichen Mühen welche konstruiert haben. Muss man aber ernsthaft damit rechnen, dass man sozusagen bei der täglichen Arbeit über eine unentscheidbare Aussage stolpert, ohne danach gesucht zu haben? Neuere Ergebnisse lassen vermuten, dass das Risiko längst nicht so klein ist, wie man gedacht hat.

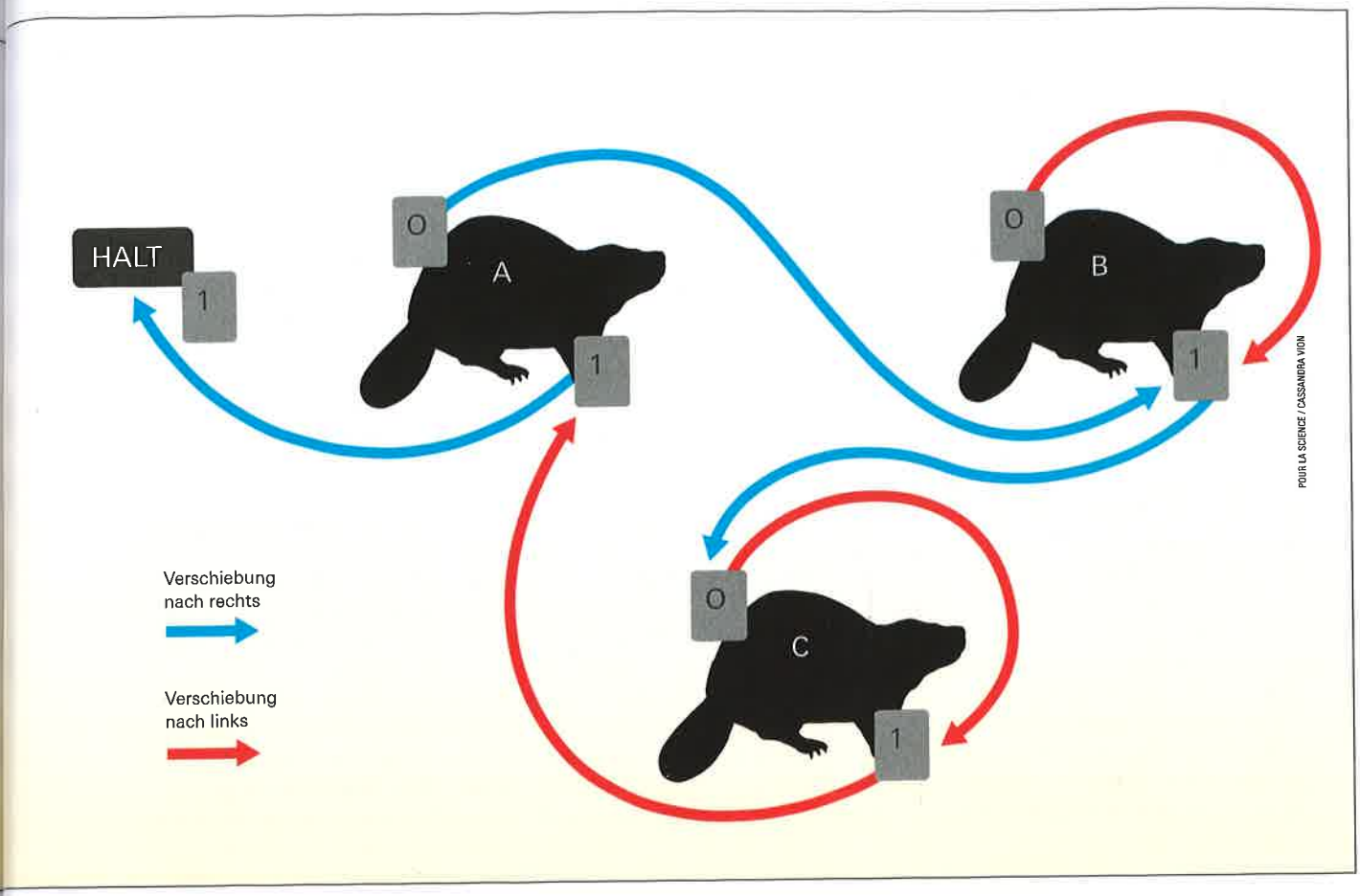
Unentscheidbarkeit bei einfachen Aussagen?

Es gibt eine Funktion, die nach ihrem Erfinder, dem ungarischen Mathematiker Tibor Radó (1895–1965), benannt ist. Sie ist wohldefiniert, das heißt, es gibt eine eindeutige Vorschrift, die jeder natürlichen Zahl *n* einen Funktionswert $s(n)$ zuordnet. Aber für $n = 1919$ ist dieser Wert unberechenbar. Das heißt: Jede Aussage, die diesen Wert bestimmen würde, ist in der mächtigen klassischen Mengenlehre unentscheidbar.

Die Radó-Funktion ist über ein Konzept definiert, das Alan Turing (1912–1954) 1936 eingeführt hat. Eine Turing-Maschine (Bild S. 73) ist ein Automat, der nur endlich viele Zustände annehmen kann. Mit einem Schreib-Lese-Kopf kann er Symbole von einem Magnetband, das in Felder eingeteilt ist, lesen und auf das Band schreiben, insbesondere bereits vorhandene Symbole überschreiben. Die Maschine kann zu jedem Zeitpunkt nur auf ein Feld des Magnetbands zugreifen und in jedem Arbeitsschritt auf dem Band ein Feld nach rechts oder links wandern. Ihre Arbeit besteht darin, dass sie entsprechend ihrem Programm und den auf dem Band vorgefundenen Symbolen das Band durchläuft und dabei neu beschreibt. Man nennt diese Arbeit »Rechnen«, denn eine Turing-Maschine kann mit dieser Primitivausstattung tatsächlich Rechenoperationen durchführen – im Prinzip sogar alles berechnen, was überhaupt berechenbar ist; aber das ist eine andere Geschichte.

Das Programm der Maschine besteht aus einer endlichen Liste von Befehlen der Form [A 0 → 1 rechts B]. Das bedeutet: »Wenn du im Zustand A bist und im aktuellen Feld 0 vorfindest, schreibe 1 in dieses Feld, wandere ein Feld nach rechts und gehe in den Zustand B über.«

Betrachten wir zum Beispiel eine Maschine mit zwei Zuständen A und B und den Befehlen [A 0 → 0 rechts A], [A 1 → 0 rechts A] sowie [A 2 → 0 links B]. Setzt man den Schreib-Lese-Kopf der Maschine an den Anfang eines Bandes mit den Einträgen 0101101020010..., so durchläuft sie dieses Band von links nach rechts und ersetzt die Einsen durch Nullen. Erreicht sie die 2, so läuft sie ein Feld zurück und hält an, denn es gibt keinen Befehl, der ihr sagen würde, was sie im Zustand B tun soll.



Ein unendliches Band, das nur Nullen und Einsen enthält, durchläuft die Maschine nach rechts, ohne jemals stehen zu bleiben, wobei sie jede Eins durch eine Null ersetzt. Es ist also möglich, dass eine Turing-Maschine unendlich lange arbeitet, ohne anzuhalten.

Ein Zustand wie B in unserem Beispiel, für den kein Befehl existiert, ist ein »Haltezustand«. Es ist üblich, den Haltezustand nicht mitzuzählen. (Man kann den Befehlssatz einer Turing-Maschine stets so formulieren, dass es nur einen Haltezustand gibt.) Dementsprechend ist die Maschine aus unserem Beispiel eine Ein-Zustand-Maschine. Im Folgenden beschränken wir uns, bis auf ausdrücklich erwähnte Ausnahmen, auf den Anfangszustand, in dem das (möglicherweise unendliche) Band nur mit Nullen beschrieben ist. Weiter setzen wir voraus, dass es nur die beiden Symbole 0 und 1 gibt.

Fleißige Biber

Wenn man eine kleine Maschine, also eine mit wenigen Zuständen und wenigen Befehlen, auf ein Band mit lauter Nullen setzt, wird sie entweder unendlich lange rechnen, indem sie beispielsweise das Band durchläuft und alle Nullen in Einsen verwandelt oder indem sie zwischen zwei Feldern hin- und herpendelt, ohne irgendetwas zu tun; oder sie wird eine kurze Rechnung durchführen und dann anhalten. Eine einfache Maschine produziert entweder eine unendliche Rechnung oder eine kurze. Und just diese Unterscheidung führt, wie wir sehen werden, geradewegs auf die einfachsten Formen der Unentscheidbarkeit.

Im Jahr 1962 interessierte sich Tibor Radó für die maximale Anzahl der Arbeitsschritte, die eine Turing-Maschine mit n Zuständen ausführt, bevor sie anhält – wenn überhaupt. Diese Zahl bezeichnet man mit $s(n)$ und eine Maschine, die das Maximum erreicht, als »fleißigen Biber« (»busy beaver«, siehe **Spektrum** November 1984, S. 8, und Januar 1985, S. 13). Die Zahl $s(n)$ ist eindeutig definiert, denn es gibt nur endlich viele mögliche Befehle und daher auch nur endlich viele Maschinen mit n Zuständen, die nur die Symbole 0 und 1 verwenden: Es sind $(4n + 2)^{2n}$ Stück.

Um also $s(n)$ für eine gegebene Zahl n zu berechnen, genügt es, alle Maschinen mit n Zuständen laufen zu lassen und unter denen, die anhalten, diejenige zu finden, die am längsten läuft. Natürlich kann man eine Turing-Maschine auf einem gewöhnlichen Computer simulieren, und bei den heute verfügbaren unglaublichen Rechenleistungen sollte es kein Problem sein, in kurzer Zeit alle Maschinen aus dem endlichen Sortiment durchzuspielen.

Aber das ist ein großer Irrtum! Trotz beachtlicher Anstrengungen kennt man bis heute nur die Werte $s(1) = 1$, $s(2) = 6$, $s(3) = 21$ und $s(4) = 107$. Weiter weiß man, dass $s(5) \geq 47\,176\,870$ und $s(6) \geq 7{,}4 \cdot 10^{36534}$ ist (siehe »Die Turing-Maschinen, die am längsten rechnen«, S. 76). Es gibt nämlich eine Maschine mit fünf Zuständen, die nach 47 176 870 Schritten anhält, sowie eine Sechs-Zustände-

Die Turing-Maschinen, die am längsten rechnen

Zu jeder natürlichen Zahl n gibt es eine Rekord-Turing-Maschine (den »fleißigen Biber«) mit n Zuständen. Das bedeutet: Jede andere Maschine mit n Zuständen, die nur die Symbole 0 und 1 verwendet, hält entweder früher an als der fleißige Biber oder gar nicht. Die Anzahl der Schritte des fleißigen Bibers mit n Zuständen wird mit $s(n)$ bezeichnet.

Bislang sind folgende Resultate bekannt: $s(1) = 1$, $s(2) = 6$, $s(3) = 21$ (Lin und Radó, 1965), $s(4) = 107$ (Brady, 1975); $s(5) \geq 47\,176\,870$ (Marxen und Buntrock, 1990); $s(6) \geq 7{,}4 \cdot 10^{36534}$ (Kropitz, 2000);

$s(7) > 10^{10^{10^{10^{10^{7}}}}}$ (Wythagoras, 2014).

Der fleißige Biber mit zwei Zuständen hält nach sechs Schritten an, nachdem er vier Einsen geschrieben hat. Sein Programm:

[A 0 → 1 rechts B],
[A 1 → 1 links B],
[B 0 → 1 links B],
[B 1 → 1 rechts HALT]

Der fleißige Biber mit vier Zuständen hält nach 107 Schritten an, nachdem er 13 Einsen geschrieben hat. Sein Programm:

[A 0 → 1 rechts B],
[A 1 → 1 links B],
[B 0 → 1 rechts A],
[B 1 → 0 links C],
[C 0 → 1 rechts HALT],
[C 1 → 1 links D],
[D 0 → 1 rechts D],
[D 1 → 0 rechts A]

Der bislang beste Kandidat für den fleißigen Biber mit fünf Zuständen hält nach 47 176 870 Schritten und 4098 Einsen an. Sein Programm:

[A 0 → 1 links B],
[A 1 → 1 rechts C],
[B 0 → 1 links C],
[B 1 → 1 links B],
[C 0 → 1 links D],
[C 1 → 0 rechts E],
[D 0 → 1 rechts A]
[D 1 → 1 rechts D]
[E 0 → 1 links HALT],
[E 1 → 0 rechts A]

Maschine, die solches nach der genannten astronomischen Zahl von Schritten tut. Allerdings ist sich niemand sicher, ob diese Werte das letzte Wort sind.

Warum ist es so schwierig, $s(n)$ auszurechnen? Erstens nimmt die Anzahl der Maschinen mit n Zuständen mit wachsendem n sehr schnell zu, und es gibt keine Abkürzung: Wer wissen will, nach wie vielen Schritten eine Maschine anhält, dem bleibt bis auf wenige Ausnahmefälle nichts übrig, als sie laufen zu lassen. Zweitens und noch übler: Es gibt kein allgemeines Verfahren, um zu bestimmen, ob eine Maschine überhaupt anhält. In einigen Fällen gelingt das durch Analyse des Befehlssatzes; aber man stößt schnell auf Maschinen mit komplexem Verhalten, die sich dieser Analyse entziehen. Also weiß man nicht, ob sie vielleicht doch einen Beitrag zur Bestimmung von $s(n)$ liefern oder ob sie immer weiterlaufen und deshalb für $s(n)$ irrelevant sind.

Schlimmer noch: Es kann keinen allgemeinen Algorithmus geben, der diese Frage beantwortet. Das ist die berühmte Unentscheidbarkeit des Halteproblems, die Turing 1936 bewiesen hat. Kein Algorithmus ist fähig, für jede gegebene Maschine dieses Typs festzustellen, ob sie anhält oder nicht.

Folglich ist auch $s(n)$ nicht durch einen Algorithmus berechenbar. Aus diesem Grund hat Radó seine Funktion eingeführt. Noch interessanter ist, dass $s(n)$ schneller wächst als jede algorithmisch berechenbare Funktion.

Dafür hat Radó folgenden Widerspruchsbeweis angegeben: Nehmen wir an, es gäbe eine algorithmisch berechenbare Funktion $f(n)$, die $s(n)$ majorisiert, das heißt, für alle natürlichen Zahlen n wäre $f(n) \geq s(n)$. Dann könnte man algorithmisch herausfinden, ob eine gegebene Maschine mit n Zuständen anhält oder nicht. Und zwar berechnet man zuerst die Zahl $f(n)$ und dann $f(n)$ Rechenschritte der Maschine. Entweder hält sie während dieser Zeit an – dann ist dieser Teil der Berechnung von $s(n)$ erledigt – oder nicht; dann wird sie niemals anhalten, denn sie hat ja schon mehr als $s(n)$ Schritte zurückgelegt, was die Maximalzahl der Rechenschritte einer anhaltenden Maschine ist. Also wäre das beschriebene Verfahren ein Algorithmus zur Berechnung von $s(n)$. Den kann es nicht geben, siehe oben. Also kann es auch die Funktion $f(n)$ nicht geben, was zu beweisen war.

Anders ausgedrückt: Jede algorithmisch berechenbare Funktion $f(n)$ wird irgendwann von $s(n)$ überholt, oder formal korrekt ausgedrückt: Für jede algorithmisch berechenbare Funktion $f(n)$ gibt es eine natürliche Zahl n_0 mit der Eigenschaft, dass $f(n) \leq s(n)$ ist für alle $n > n_0$.

Unentscheidbare Aussagen dank der Radó-Funktion

Zwischen der Funktion von Radó und der Mengenlehre bestehen interessante Zusammenhänge. Diese führen uns zu neuen kurzen unentscheidbaren Aussagen, die erst vor gut einem Jahr entdeckt wurden.

Die Mengenlehre dient hier als das klassische Exemplar einer Theorie, die so mächtig ist, dass man in ihr die üblichen Sätze der Arithmetik formulieren kann. Die heute gebräuchlichste Version der Mengenlehre wird als ZFC bezeichnet. Dabei stehen Z und F für Ernst Zermelo (1871–1953)

und Abraham Fraenkel (1891–1965), die der Mengenlehre ihre heutige Form gaben, und C (»Choice«) für das Auswahlaxiom. Dieses fordert, dass es zu jeder Familie von nichtleeren Mengen, von denen keine zwei ein Element gemeinsam haben, eine Menge (die »Auswahl«) gibt, die aus jeder Menge der Familie genau ein Element enthält. (»Familie« ist hier nur, der Abwechslung zuliebe, ein anderes Wort für »Menge«.) Interessant ist das Auswahlaxiom für unendliche Familien unendlicher Mengen, die ihrerseits unstrukturiert sind. Sonst könnte man die Auswahl einfach durch das Rezept »Nimm von jeder Menge der Familie das kleinste (oder sonst wie ausgezeichnete) Element« definieren. Man verwendet nun ZFC seit einem Jahrhundert und ist noch nie auf einen Widerspruch gestoßen. Wir nehmen deshalb an, es gebe keinen.

Man beweist nun: Ist eine Funktion $u(n)$ nicht algorithmisch berechenbar, so kann die Mengenlehre ZFC nicht alle ihre Werte liefern. Formal korrekt ausgedrückt: Es existiert mindestens eine natürliche Zahl n, so dass die Aussage $u(n) = k$, die den Wert der Funktion u an der Stelle n angibt, in ZFC unentscheidbar ist. Das trifft auch in einer beliebigen anderen widerspruchsfreien Theorie zu, die man an Stelle von ZFC verwendet, allerdings nicht unbedingt mit demselben n.

Was die Funktion $s(n)$ von Radó betrifft, so weiß man seit 1962, dass es bei Verwendung von ZFC mindestens eine natürliche Zahl n_0 gibt, für die man den Wert $s(n_0)$ nicht bestimmen kann: Weder die Aussage $s(n_0) = k$ noch ihre Negation sind für das richtige k in ZFC beweisbar. Die Berechnung von $s(n_0)$ übersteigt die deduktive Mächtigkeit der Theorie.

Wie groß ist n_0? Ungefähr 10? 1000? Oder eher in der Nähe von 100 Milliarden? Diese Frage stellten sich Scott Aaronson, Professor am MIT, und sein Student Adam

Fleißige Biber berechnen die Entropie

Die Kenntnis der Werte von Radós Funktion $s(n)$ verhilft zu unerwarteten Ergebnissen. Unter der Annahme, dass der vermutete Wert $s(5) = 47\,176\,870$ der richtige sei, hat eine Arbeitsgruppe um Fernando Soler-Toscano von der Universidad de Sevilla (Spanien) die Kolmogorow-Komplexität kurzer Folgen aus Nullen und Einsen berechnet. Diese Größe wird auch »algorithmische Entropie« genannt, weil sie eine Verallgemeinerung der statistischen Entropie aus Claude Shannons Informationstheorie ist.

Die Kolmogorow-Komplexität einer (endlichen) Folge ist definiert als die Länge des kürzesten Programms, das als Output die Folge erzeugt. Wenn die Folge überhaupt keine Regelmäßigkeit hat, ist das kürzeste erzeugende Programm dasjenige, das die Folgenglieder der Reihe nach aufzählt, und damit im Wesentlichen so lang wie die Folge selbst.

Diese klassische Definition funktioniert nicht für sehr kurze Folgen, da Programme eine gewisse Mindestlänge zu haben pflegen, die auch noch von der verwendeten Programmiersprache abhängt (diese Unterschiede werden bedeutungslos für lange Folgen). Aber es gibt eine Alternative: Wenn eine Folge eine große Regelhaftigkeit (und damit eine geringe Komplexität) hat, dann wird es unter allen Turing-Maschinen mit – zum Beispiel – fünf Zuständen sehr viele geben, die diese Folge produzieren.

Man zählt also unter allen anhaltenden Turing-Maschinen mit fünf Zuständen aus, wie viele von ihnen eine bestimmte Folge aufs Band schreiben, bevor sie anhalten. Je größer diese Anzahl, desto geringer ist die Komplexität der Folge. Genauer: Nach einer Formel von Leonid Levin ist die Komplexität $K(t)$ der Folge t näherungsweise gleich $-\log_2(k/N(5))$, also dem negativen Zweierlogarithmus des Bruchteils aller Turing-Maschinen, die t produzieren. k ist die Anzahl dieser Maschinen und $N(5) = 22^{10} = 26\,559\,922\,791\,424 \approx 2{,}6 \cdot 10^{13}$ die Gesamtzahl aller Turing-Maschinen mit fünf Zuständen.

Wie bestimmt man k? Man lässt jede der $N(5)$ Maschinen laufen, bis sie anhält oder bis sie $s(5)$ Schritte vollführt hat. Um diese gigantische Rechnung überhaupt durchführen zu können, muss man also $s(5)$ kennen, ersatzweise den oben genannten Schätzwert. Wenn $s(5)$ doch noch größer sein sollte als 47 176 870, bleiben die Maschinen, die jenseits dieser Schrittzahl doch noch anhalten, unberücksichtigt. Man hofft, dass diese am Ergebnis nicht mehr viel ändern.

Die Formel von Leonid Levin ist bemerkenswert und sehr tief liegend. Sie bedeutet, dass die komplexen Folgen (das sind solche, für die das sie erzeugende Programm lang ist) auch diejenigen sind, die eine zufällig ausgewählte Maschine mit der kleinsten Wahrscheinlichkeit erzeugt.

Die Tabelle zeigt einige der so berechneten Werte.

FOLGE	HÄUFIGKEIT	ENTROPIE
1	0,175036	2,51428
0	0,175036	2,51428
11	0,0996187	3,32744
10	0,0996187	3,32744
01	0,0996187	3,32744
00	0,0996187	3,32744
111	0,0237546	5,3962
000	0,0237546	5,3962
110	0,0229434	5,44578
100	0,0229434	5,44578
011	0,0229434	5,44578
001	0,0229434	5,44578
101	0,0220148	5,5038
010	0,0220148	5,5038
1111	0,0040981	7,03083
0000	0,0040981	7,03083
1110	0,00343136	8,187
1000	0,00343136	8,187
0111	0,00343136	8,187
0001	0,00343136	8,187

Goldbach-Maschinen und Riemann-Maschinen

Eine Turing-Maschine, die in der Mengenlehre ZFC einen Widerspruch sucht (und vermutlich nie anhält, weil es keinen gibt), ist groß und unhandlich – wenig verwunderlich bei der Größe der Aufgabe. Da ist es durchaus interessant, das Problem und die zugehörige Maschine einige Nummern kleiner zu wählen.

Man konstruiere eine Turing-Maschine, die ein Gegenbeispiel zur Goldbach-Vermutung sucht (**Spektrum** Dezember 2008, S. 94): Das wäre eine gerade natürliche Zahl größer als 2, die sich nicht als Summe zweier Primzahlen schreiben ließe. Eine derartige »Goldbach-Maschine« hält nur an, wenn die Goldbach-Vermutung falsch ist. Die Anzahl der Zustände dieser Turing-Maschine, die man versucht, möglichst klein zu halten, ist ein Maß für die Schwierigkeit der Vermutung. Der kürzlich erzielte Rekord ist 47. Ein Mensch, der nur unter seinem Internet-Spitznamen »code golf addict« bekannt ist, hat eine Maschine mit 27 Zuständen ins Netz gestellt, die noch der Nachprüfung harrt. Wenn sich herausstellen sollte, dass die Goldbach-Vermutung unentscheidbar in ZFC ist, dann wäre bereits $s(47)$ in ZFC unberechenbar.

Dieselbe Argumentation gilt für die berühmte riemannsche Vermutung aus der Zahlentheorie (**Spektrum** September 2008, S. 86). Sie besagt, dass die nichttrivialen Nullstellen der riemannschen Zetafunktion alle den Realteil ½ besitzen. Zudem liefert sie genauere Angaben über die Verteilung der Primzahlen.

Die kleinste derzeit bekannte Turing-Maschine, die genau dann anhält, wenn die riemannsche Vermutung falsch ist, hat 744 Zustände.

Yedidia. Interessant ist es, ein möglichst kleines n_0 zu finden; denn wenn die Theorie für ein gewisses n_0 versagt, dann auch für alle folgenden. Versagen bedeutet ja, dass das Verhalten einer Turing-Maschine mit n_0 Zuständen nicht entscheidbar ist. Es ist aber kein Problem, eine Maschine zu definieren, die mehr Zustände hat als die problematische, sich jedoch genau so verhält (weil sie die zusätzlichen Zustände nie annimmt). Außerdem bietet ein kleines n_0 den morbiden Reiz, dass die gefürchtete Unentscheidbarkeit sozusagen um die Ecke lauert.

Aaronson und Yedidia konnten beweisen, dass ZFC für $n = 7918$ versagt. Kein Mathematiker, der mit der klassischen Mengenlehre arbeitet, wird jemals $s(7918)$ kennen können. Anders gesagt: Unter den Turing-Maschinen mit 7918 Zuständen, die nicht anhalten, tun dies einige aus Gründen, welche die Theorie nicht analysieren kann. Dies ist das erste Mal, dass man eine in der Mengenlehre unentscheidbare Aussage derart konkret benennen kann.

Kaum hatten Yedidia und Aaronson im Mai 2016 ihre Arbeit publiziert, folgten andere, die mit verbesserten Beweistechniken den Wert von n_0 drücken konnten. Stefan O'Rear bewies, dass ZFC für $s(5349)$ versagt, und besserte diese Zahl wenige Tage später nochmals nach: Auch $s(1919)$ liegt außerhalb der Reichweite von ZFC.

Seitdem kann niemand mehr behaupten, dass die in der Mengenlehre unentscheidbaren Aussagen esoterisch und – über ihre Unentscheidbarkeit hinaus – völlig uninteressant wären. Die Frage nach dem Wert von $s(1919)$ ist vollkommen natürlich, denn sie betrifft den elementarsten Teil der mathematischen Theorie der Berechnung.

Die Methode von Yedidia und Aaronson ist interessant, weil sie an entscheidender Stelle neue Wege geht. Üblicherweise würde man alle Beweise, die man in ZFC überhaupt formulieren kann, der Reihe nach aufzählen und daraufhin überprüfen, ob einer von ihnen die Aussage »0=1« beweist. Mit einem Verfahren (der »Arithmetisierung«), das ebenfalls von Gödel stammt, kann man alle diese – in einer formalen Sprache ausgedrückten – Beweise in Zahlen verwandeln und mit diesen Zahlen rechnen; also kann man diese Aufgabe auch einer Turing-Maschine anvertrauen.

Eine solche Maschine wird niemals anhalten, denn ZFC ist widerspruchsfrei; aber das kann man nicht beweisen, weil sonst ZFC ihre eigene Widerspruchsfreiheit beweisen würde, was nach dem zweiten gödelschen Unvollständigkeitssatz unmöglich ist. Die so konstruierte Turing-Maschine hätte jedoch mehrere 100 000 oder gar Millionen Zustände. Folglich würde man die Unentscheidbarkeit von $s(n) = k$ nur für ein n in dieser Größenordnung beweisen.

Die Maschine Z hält niemals an

Aaronson und Yedidia vermeiden diese riesigen Zustandszahlen mit Hilfe neuerer Arbeiten von Harvey Friedman, Emeritus der Ohio State University. Friedman arbeitet seit Jahren arithmetische Aussagen aus, die in ZFC oder anderen Theorien unentscheidbar sind. Eine dieser Aussagen bezieht sich auf eine Eigenschaft von Graphen; sie ist äquivalent zu der Aussage, dass ZFC widerspruchsfrei ist. Diese Aussage ist zwar zu abstrakt, um hier erklärt werden zu können, lässt sich jedoch vergleichsweise leicht in arithmetischen Ausdrücken formulieren. Eine Maschine, die ein Gegenbeispiel zu ihr sucht, ist folglich relativ einfach im Vergleich zu derjenigen, die man durch direkte Arithmetisierung von ZFC erhalten würde.

Der Rest des Arguments verläuft wie oben. Die Maschine – man nennt sie Z – prüft der Reihe nach einen Graphen nach dem anderen, ob er die genannte Eigenschaft besitzt. Sie hält an, wenn das nicht der Fall ist. Das wird zwar nicht passieren, denn ZFC ist widerspruchsfrei. Es ist jedoch unmöglich, zu beweisen, dass sie nicht anhalten

wird, denn sonst würde ZFC wieder ihre eigene Widerspruchsfreiheit beweisen.

Bezeichnen wir die Anzahl der Zustände von Z mit n_0. Um $s(n_0)$ zu kennen, müsste man in ZFC für jede Maschine mit n_0 Zuständen, die nicht anhält, beweisen können, dass sie nicht anhält. Es gibt aber eine solche Maschine, nämlich Z, für die das unmöglich ist. Also kann ZFC die Aussage $s(n_0) = k$ nicht für den richtigen Wert von k beweisen.

Übrigens ist die fragliche Eigenschaft von Graphen in einer Theorie beweisbar, die nur wenig stärker ist als ZFC. Folglich hat man allen Grund zu der Annahme, sie sei wahr. Zu allem Überfluss ließen Aaronson und Yedidia ihre – im Computer simulierte – Maschine Z tatsächlich laufen, um zu sehen, ob sie anhält. Nach einigen Stunden hatte sie nicht angehalten.

Die Arbeiten Friedmans lieferten zwar eine große Abkürzung; aber damit allein wären die heute bekannten kleinen Werte von n_0 nicht zu finden gewesen. Das gelang erst mit einer zweiten originellen Idee. Aaronson und Yedidia schufen eine formale Sprache namens Laconic, mit der Friedmans Eigenschaft sehr knapp ausgedrückt werden kann – lakonisch eben. Zusätzlich schrieben sie ein Übersetzungsprogramm, das aus einem Text in Laconic eine Turing-Maschine nach den Vorgaben von Radó macht. So kamen sie zu ihrer Maschine mit 7918 Zuständen. Verbesserungen an der Sprache und dem Übersetzungsprogramm führten zu der Reduktion auf 1919.

Es ist anzunehmen, dass man auch den Rekord n_0 = 1919 noch unterbieten kann. Immerhin ist der Wert $s(5)$ bereits so schwierig zu berechnen, dass er bis heute unbekannt ist. Irgendwo zwischen 5 und 1919 stößt ZFC an ihre prinzipiellen Grenzen, und zwar, so wie es heute aussieht, eher früher als später. Vielleicht ist in der Stunde, in der Sie das lesen, der Rekord bereits gebrochen. Ein Blick in Scott Aaronsons Blog www.scottaaronson.com/blog/ dürfte Klarheit schaffen. ◂

QUELLEN

Aaronson, S.: The 8000th Busy Beaver Number Eludes ZF Set Theory: New paper by Adam Yedidia and me. www.scottaaronson.com/blog/?m=201605, Mai 2016

Calude, C. S., Calude, E.: Evaluating the Complexity of Mathematical Problems. Part 1: Complex Systems 18, S. 267–285, 2009; Part 2: Complex Systems 18, S. 387–401, 2010

Marxen, H.: List of the Known Busy Beaver values. www.drb.insel.de/~heiner/BB/index.html, 2016

Michel, P.: The Busy Beaver Competition: A Historical Survey. https://arxiv.org/abs/0906.3749, 2009

Soler-Toscano, F. et al.: Calculating Kolmogorov Complexity from the Output Frequency Distributions of Small Turing Machines. In: PLoS One 8, e96223, 2014

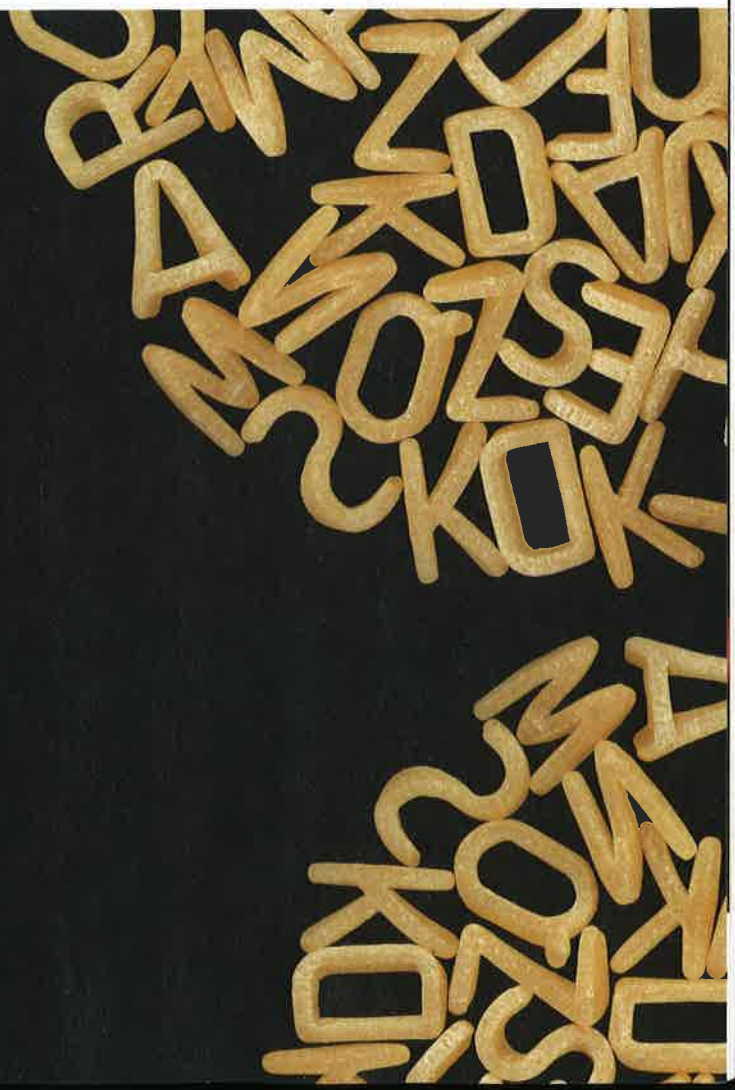
Yedidia, A., Aaronson, S.: A Relatively Small Turing Machine Whose Behavior Is Independent of Set Theory. https://arxiv.org/abs/1605.04343, 2016

Spektrum der Wissenschaft

SCHREIB-WERKSTATT

Möchten Sie mehr darüber erfahren, wie ein wissenschaftlicher Verlag arbeitet, und die Grundregeln fachjournalistischen Schreibens erlernen?

Dann profitieren Sie als Teilnehmer des Spektrum-Workshops »Wissenschaftsjournalismus« vom Praxiswissen unserer Redakteure.

Ort: Heidelberg
Spektrum-Workshop »Wissenschaftsjournalismus«; Preis: € 139,– pro Person; Sonderpreis für Abonnenten: € 129,–

Telefon: 06221 9126-743 | service@spektrum.de
spektrum.de/schreibwerkstatt