

Grenzen der Berechenbarkeit

Der Zufall gehört unvermeidlich zur Mathematik, und es gibt stets wahre, aber unbeweisbare Sätze. Kein Anlass zu Pessimismus: Die Grenzen unserer Erkenntnis sind bestimmt durch die begrenzte Größe unserer Erkenntniswerkzeuge; und die lässt sich aufstocken.

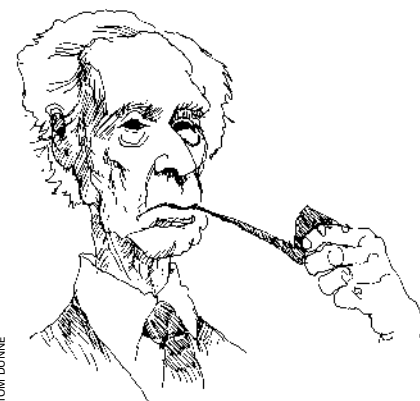
Von Gregory J. Chaitin

Computer sind ohne Zweifel überaus nützliche Geräte – so nützlich, dass eine moderne Gesellschaft kaum noch auf sie verzichten kann. Aber dafür wurden sie gar nicht erfunden. Eigentlich ging es den Vordenkern der heute allgegenwärtigen Rechner um ein philosophisches Grundlagenproblem der Mathematik (ich übertreibe nur ein bisschen). Diese erstaunliche Tatsache ist selbst den meisten Computerexperten nicht geläufig.

Die Geschichte beginnt mit David Hilbert (1862–1943), der allgemein als der führende Mathematiker seiner Zeit gilt. Zu Beginn des 20. Jahrhunderts rief Hilbert zur vollständigen Formalisierung mathematischen Beweisens auf. Dreißig Jahre später stellte sich heraus, dass genau dies aus prinzipiellen Gründen unmöglich ist. Damit war Hilberts Programm einerseits ein spektakulärer Fehlschlag, andererseits ein grandioser Erfolg, denn die Idee des Formalismus trug reiche Früchte. Nicht so sehr für die Theorie mathematischen Beweisens als vielmehr für die Entwicklung der Computer- und Informationstechnik des 20. Jahrhunderts.

Ich möchte diese fast in Vergessenheit geratene Geschichte erzählen, ohne mich in mathematischen Details zu ver-

lieren. Es würde viel zu viel Mühe kosten, die Beiträge der bedeutenden Protagonisten wie Bertrand Russell, Kurt Gödel und Alan Turing vollständig zu erklären. Aber mit etwas Geduld, so hoffe ich, können Sie das Wesentliche der verschiedenen Gedankengänge nachvollziehen – bis hin zu meinen eigenen Vorstellungen über den Zufall als natürlichen Bestandteil der Mathematik.



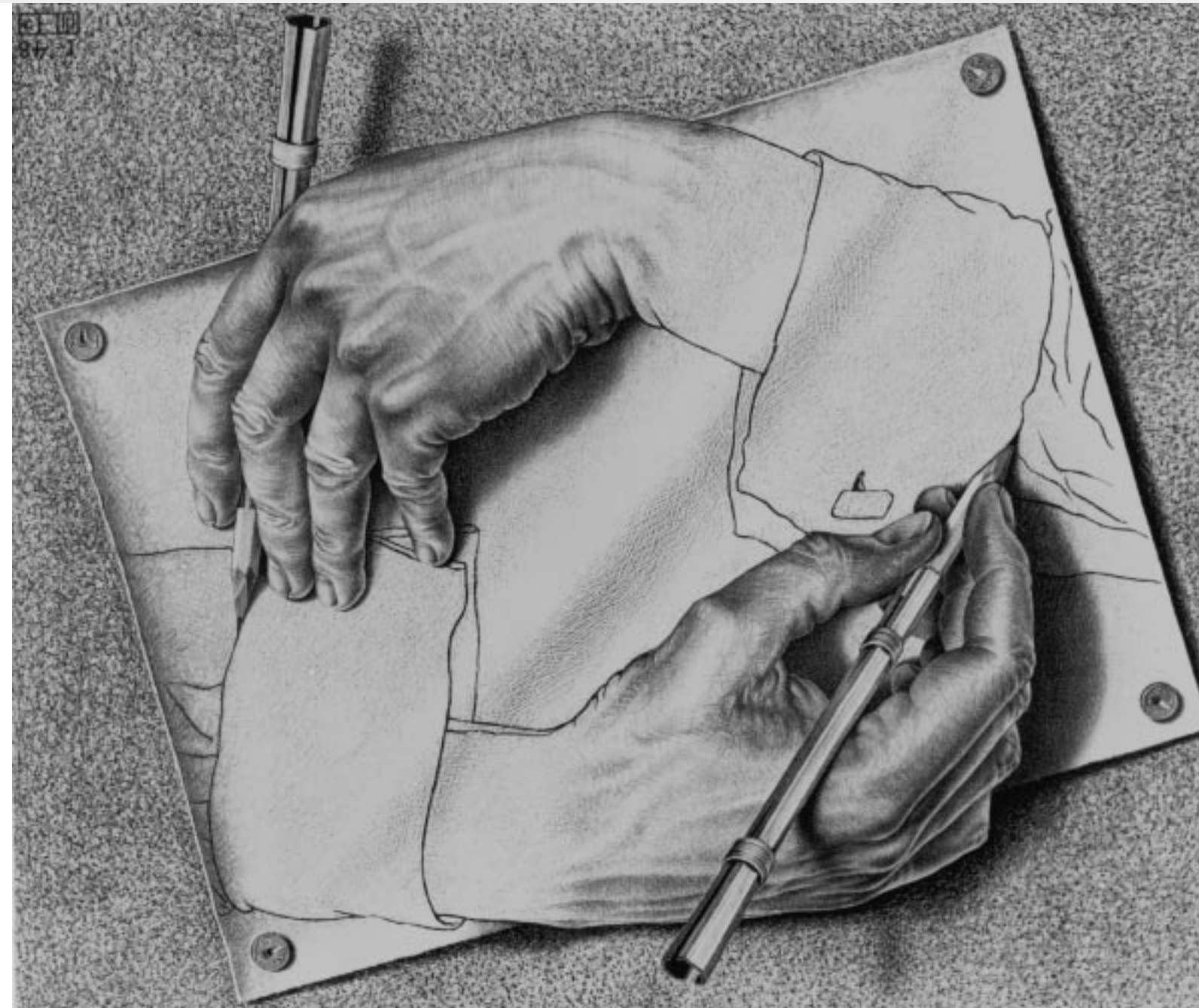
Russells Paradoxon

Beginnen wir mit Bertrand Russell (1872–1970), der seine wissenschaftliche Laufbahn als Mathematiker begann, sich dann der Philosophie zuwandte und schließlich zum Humanisten wurde. Russell ist eine Schlüsselfigur in unserer

Geschichte, denn er entdeckte Besorgnis erregende Paradoxa innerhalb der Logik selbst, genauer gesagt Fälle, in denen das reine, anscheinend wohl fundierte Denken auf Widersprüche führt. Das erschütterte das Fundament der Logik und damit zugleich das der Mathematik, die sich auf die Logik gründet. Mit dem Gedanken, dass die dadurch entstandene »Grundlagenkrise der Mathematik« dringend einer Lösung bedürfe, hat Russell seine Zeitgenossen stark beeinflusst.

Merkwürdigerweise trägt heute unter den Paradoxa, die Russell entdeckte, nur eines seinen Namen, und zwar dieses: Sei M die Menge aller Mengen, die sich nicht selbst als Element enthalten. Enthält die Menge M sich selbst? Wenn ja, dann gehört sie nicht zur Menge aller Mengen, die sich selbst nicht als Element enthalten. Das ist aber gerade die Menge M . Also enthält die Menge M nicht sich selbst. Geht man aber davon aus, sie enthalte nicht sich selbst, landet man ebenfalls beim Gegenteil der Aussage, mit der man begann.

Die anschaulichere Verkleidung des Russell'schen Paradoxons handelt von dem Dorfbarbier; das ist per definitionem derjenige Dorfbewohner, der alle männlichen Dorfbewohner rasiert, die sich nicht selbst rasieren. Das klingt zunächst ganz vernünftig – bis man danach fragt, ob der Barbier sich selbst rasiert.



2004 THE M.C. ESCHER COMPANY, BAARN, HOLLAND

Rasiert er sich nicht selbst, dann rasiert ihn der Barbier; das ist aber er selbst, also rasiert er sich selbst, also rasiert ihn nicht der Barbier, also ...

Nun mag man das mit einem gewissen Recht eine alberne Frage nennen. Was soll man sich über die Probleme eines hypothetischen Barbiers den Kopf zerbrechen? Soll der Mann sich doch einen Vollbart wachsen lassen. Doch im Zusammenhang mit dem mathematischen Mengenbegriff ist das Problem nicht so einfach beiseite zu schieben.

Über eine Frühform des Russell'schen Paradoxons – ohne die mengentheoretische Formalisierung – haben sich schon

die Griechen der Antike den Kopf zerbrochen. Das »Paradoxon des Epimenides« oder »Lügnerparadoxon« handelt von einem Ausspruch, den ein Mensch namens Epimenides getan haben soll: »Diese Behauptung ist falsch.« Ist sie nun falsch? Wenn ja, dann trifft die Behauptung des Satzes zu, also ist sie wahr. Einerlei ob Sie die Behauptung für wahr oder falsch halten: Sie können dem Widerspruch nicht entgehen!

Man kann das Problem auch in zwei Sätze zerlegen: »Die nachfolgende Behauptung ist wahr. Die vorhergehende Behauptung ist falsch.« Jede Aussage für sich ist unproblematisch, aber miteinander

▲ Eine Hand zeichnet die andere, und umgekehrt. Jede der beiden Handlungen ist für sich genommen unproblematisch; erst ihre Kombination schafft ein Paradox, wie in dem Satzpaar »Die nachfolgende Behauptung ist wahr / Die vorhergehende Behauptung ist falsch«. Zeichnung von Maurits C. Escher, 1948.

der verbunden ergeben sie keinen Sinn. Das alles mag wie ein bedeutungsloses Wortspiel anmuten, aber einige große Denker des 20. Jahrhunderts haben es sehr ernst genommen. ▶



TOM DUNNE

▷ **Hilberts Rettungsversuch**

Hilberts Ansatz, die Krise der Logik zu überwinden, bestand im Formalismus. Vielleicht waren ja die unauf löslich scheinenden Widersprüche durch die Vieldeutigkeit der gewöhnlichen Sprache verursacht und lösten sich auf, wenn man die umgangssprachlichen Sätze nicht wie üblich irgendwie, sondern richtig interpretierte. Also erzeuge man mit Hilfe der symbolischen Logik eine künstliche Sprache mit sehr strengen Regeln, die keine Interpretationsfreiheit und insbesondere keine Widersprüche zulassen.

Eine solche perfekte künstliche Sprache wollte Hilbert für das logische Schließen im Allgemeinen und für die Mathematik im Besonderen schaffen. Das läuft auf die »axiomatische Methode« hinaus: Man setzt sich gewisse grundlegende Behauptungen (»Postulate« oder »Axiome«) sowie wohldefinierte Deduktionsregeln und leitet aus den Axiomen unter Verwendung der Regeln gültige Sätze der Theorie her. Das Vorbild für die axiomatische Methode sind die »Elemente« des Euklid, ein Werk, in dem das gesammelte mathematische Wissen der Griechen um 300 v. Chr. in einer auch für heutige Verhältnisse bewundernswerten Klarheit aus den Axiomen entwickelt wird.

Hilbert ging es darum, vollkommene Klarheit über die Spielregeln zu schaffen: über die Definitionen, die Grundbegriffe, die Grammatik und die Sprache. Dies sollte jeden Dissens darüber, wie Mathematik zu betreiben ist, aus der Welt schaffen. Ein derartiges formales axiomatisches System wäre zwar viel zu schwer-

fällig, um damit neue mathematische Ergebnisse zu finden, aber gleichwohl philosophisch höchst bedeutsam.

Hilberts Projekt war die konsequente Fortführung einer langen Tradition, die auf Mathematiker wie Gottfried Wilhelm Leibniz (1646–1716), George Boole (1815–1864), Gottlob Frege (1848–1925) und Giuseppe Peano (1858–1939) zurückgeht. Neu war vor allem die Radikalität des Vorhabens: Die gesamte Mathematik sollte formalisiert werden.

Als sich herausstellte, dass dies unmöglich ist, war das eine Jahrhundertüberraschung. Hilbert hatte sich geirrt, doch sein Irrtum erwies sich als über die Maßen fruchtbar. Der Ruhm gebührt ihm also nicht für die (falsche) Antwort, sondern für die gute Frage. Denn mit ihr hat er ein neues Forschungsgebiet begründet: die Metamathematik. Ihr Ziel ist es zu ergründen, welche Ergebnisse die Mathematik liefern kann und welche nicht.

Die Grundidee ist folgende: Sowie man im Sinne von Hilbert die Mathematik in eine künstliche Sprache gezwängt und damit ein vollständig axiomatisches System erzeugt hat, kann man die ursprüngliche Bedeutung der mathematischen Begriffe vergessen. Man spielt einfach ein Spiel mit Zeichen auf Papier, bei dem man Sätze aus Axiomen ableitet. Natürlich interessiert einen die Bedeutung der Zeichen, wenn man Mathematik treibt. Will man aber die Mathematik selbst mit mathematischen Methoden untersuchen, muss man genau davon absehen und darf lediglich eine künstliche Sprache mit ganz präzisen Regeln untersuchen.

Welche Fragen kann man in diesem Zusammenhang stellen? Zum Beispiel, ob die Aussage $1 = 0$ beweisbar ist. (Hoffen wir, dass die Antwort Nein ist!) Allgemein kann man für jede Aussage A fragen, ob A beweisbar ist – oder ob das Gegenteil von A beweisbar ist. Wir nennen ein formales Axiomensystem vollständig, wenn jede in dem Axiomensystem formulierbare Aussage A entweder beweisbar oder widerlegbar (das heißt ihr Gegenteil beweisbar) ist.

Dabei stellte Hilbert sich die Regeln so klar und unzweideutig vor, dass ihre Nachprüfung eine völlig mechanische Angelegenheit wäre. Man könnte jeden Beweis einem unbestechlichen Gutachter vorlegen, der einem »mechanischen Ver-

fahren« folgen würde. Dieser Gutachter würde den Beweis entweder akzeptieren oder mit einer Begründung verwerfen wie »Schreibfehler in Zeile 4« oder »Aussage 7 folgt nicht aus Aussage 6«; und das wäre das Ende der Diskussion.

Hilbert behauptete nicht, dass Mathematik so betrieben werden sollte. Aber wenn das im Prinzip möglich wäre, dann könnte man die Schlagkraft mathematischer Argumentation wiederum mit mathematischen Methoden untersuchen. Hilbert war der Überzeugung, dass ihm dieser Kraftakt gelingen werde. Man kann sich vorstellen, wie es die mathematische Gemeinschaft erschütterte, als 1931 ein junger Österreicher namens Kurt Gödel nachwies, dass dies unmöglich ist. Hilberts Programm kann nicht durchgeführt werden – nicht einmal im Prinzip.



TOM DUNNE

Die Gödel'sche Unvollständigkeit

Kurt Gödel (1906–1978) stammte aus Brünn (heute Brno, Tschechien); als er 1931 Hilberts Visionen den Todesstoß versetzte, arbeitete er an der Universität Wien. Von den Nationalsozialisten vertrieben, ging er an das Institute for Advanced Studies nach Princeton, wo sich damals auch Albert Einstein aufhielt (Bild Seite 92).

Gödels erstaunliche Entdeckung war, dass Hilberts Programm von Anfang an nicht durchführbar war. Es lag nicht daran, dass die kompliziertesten Aussagen über, sagen wir, unendlichdimensionale Vektorräume den Formalismus überfordern hätten. Hilberts Programm scheitert bereits in der Grundschule, genauer ge-

sagt, an der elementaren Arithmetik: den Zahlen 0, 1, 2, 3, ... , Addition, Multiplikation und den zugehörigen Rechenregeln.

Jedes formale Axiomensystem, das versucht, die Wahrheit und nichts als die Wahrheit über Addition, Multiplikation und die natürlichen Zahlen zu sagen, ist zwangsläufig unvollständig. Genauer sagt Gödels Unvollständigkeitssatz: Ein solches Axiomensystem ist entweder unvollständig oder, was noch schlimmer wäre, es enthält innere Widersprüche. Wenn man das Axiomensystem also so konstruiert, dass es nichts als die Wahrheit sagen kann, dann sagt es einem nicht die ganze Wahrheit. Wenn man es unfähig macht, falsche Aussagen zu beweisen, dann gibt es wahre Aussagen, die es nicht beweisen kann.

Gödels Beweis der Unvollständigkeit ist sehr trickreich, ziemlich paradox und fast ein bisschen verrückt. Ausgangspunkt ist das Paradoxon vom Lügner, also die Aussage »Ich sage nicht die Wahrheit«, die weder wahr noch falsch ist. Gödel konstruiert eine Aussage, die von sich selbst behauptet: »Ich bin unweisbar!«

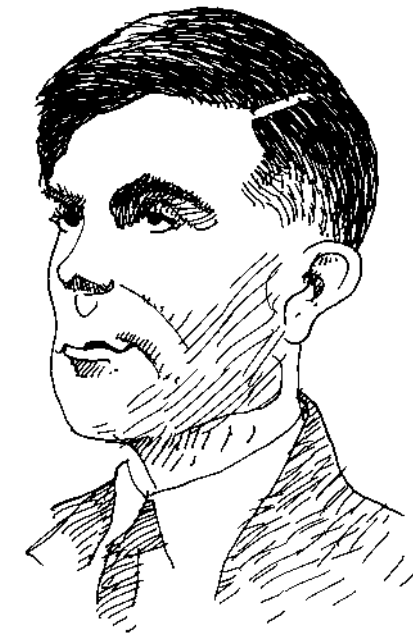
Man muss schon ziemlich genial sein, um im Rahmen der elementaren Arithmetik eine mathematische Aussage zu formulieren, die sich selbst beschreibt. Aber wenn man das geschafft hat, wird es erst richtig schwierig. Warum das? Nun, wenn die Aussage beweisbar ist, ist sie zwangsläufig falsch, und man hat einen falschen Satz bewiesen. Ist der Satz nicht beweisbar, wie er es von sich selber sagt, dann ist er wahr, also gibt es eine wahre, nicht beweisbare Aussage. Daraus folgt die Unvollständigkeit der Mathematik.

Schaut man in Gödels Originalarbeit, stößt man auf Strukturen, die einen stark an die Programmiersprache LISP erinnern. Es werden häufig rekursive Funktionen verwendet, Funktionen, die Listen abarbeiten; und die sind charakteristisch für LISP. Es gab zwar 1931 weder Computer noch Programmiersprachen; aber mittels der Gnade der späten Einsicht können wir als das Herzstück von Gödels Originalarbeit eine Programmiersprache erkennen.

Ein anderer berühmter Mathematiker dieser Zeit, John von Neumann (1903–1957), erkannte die Bedeutung von Gödels Ergebnis sofort, obwohl ihm selbst nie in den Sinn gekommen war,

dass Hilberts Programm undurchführbar sein könnte. (Später trieb von Neumann die Entwicklung der Computertechnik in den USA maßgeblich voran.) Gödel war nicht nur genial, sondern auch mutig genug, sich vorzustellen, der große Hilbert könne im Unrecht sein.

Viele Mathematiker empfanden Gödels Schlussfolgerungen als absolut verheerend. Von der gesamten traditionellen Philosophie der Mathematik war in ihren Augen nur noch ein Scherbenhaufen geblieben. Die düstere Empfindung passte in die düstere Zeit: Die Wirtschaftskrise bestimmte den Alltag, und ein drohender Krieg warf seine Schatten voraus.



TOM DUNNE

Turings Maschine

Fünf Jahre später hatte sich der zentrale Schauplatz unserer Geschichte nach England verlagert, wo Alan Turing (1912–1954) die Unberechenbarkeit entdeckte. Erinnern wir uns an Hilberts Vorstellung von einem »mechanischen Verfahren«, mit dem man entscheiden könne, ob ein Beweis korrekt sei oder nicht. Hilbert hat diese Vorstellung nie näher ausgeführt; das tat nun Turing, indem er das mechanische Verfahren von einer gedachten Maschine ausführen ließ; heute nennt man sie Turing-Maschine (Kasten Seite 90).

Turings Originalarbeit enthält ebenso wie die Gödels eine Programmiersprache, genauer gesagt eine Struktur, die wir heute als Programmiersprache bezeich-

nen würden. Allerdings sind die beiden Sprachen sehr verschieden. Turings Erfindung von 1936 ist nicht eine höhere Sprache wie LISP, sondern eher eine Art Maschinensprache: jene für Menschen so gut wie unverständliche Folge von Nullen und Einsen, die dem zentralen Rechenwerk des Computers seine sämtlichen Einzelaktionen haarklein vorschreibt. Heute würde jeder Mensch sich mit Grausen wenden, wenn er einem Computer seine Wünsche in dieser Form mitteilen müsste.

Turings virtuelle Rechenmaschine ist sehr einfach gebaut und ihre Maschinensprache ziemlich primitiv; gleichwohl ist sie zu erstaunlichen Leistungen fähig. In seiner Arbeit von 1936 behauptete Turing, eine derartige Maschine könne jede Berechnung durchführen, zu der auch ein Mensch fähig sei.

Wirklich spannend wird der Gedankengang aber erst durch die Gegenfrage: Was kann eine solche Maschine nicht? Durch Nachdenken entlang den Argumentationslinien Gödels stieß Turing auf ein Problem, das keine Turing-Maschine lösen kann: das so genannte Halteproblem. Es besteht darin, im Voraus zu entscheiden, ob eine Turing-Maschine (oder irgendein Computerprogramm) eine gestellte Aufgabe am Ende lösen und anhalten wird.

Macht man eine Zeitvorgabe, dann ist das Halteproblem einfach zu lösen. Wenn die Frage zum Beispiel lautet, ob ein Programm innerhalb eines Jahres anhalten wird, lässt man das Programm ein Jahr lang laufen; dann merkt man ja, ob es fertig geworden ist. Wirklich schwierig wird es erst, wenn man kein Zeitlimit setzt. Es gibt Programme, die enthalten »Endlosschleifen«. Das heißt, eine gewisse Folge von Anweisungen wird ohne Ende immer wieder ausgeführt. Wollte man durch Zuschauen entscheiden, ob ein solches Programm anhält, könnte man ewig warten. Das Halteproblem ohne Zeitlimit zu lösen bedeutet also herauszufinden, ob das Programm anhält, ohne es in Gang zu setzen.

Turing argumentierte ungefähr so: Angenommen, es ist möglich, ein Programm zu schreiben, das wir einen Haltetester nennen wollen: Zu jedem beliebigen vorgelegten Computerprogramm prüft der Haltetester nach, ob es anhält oder nicht. Die Eingabedaten des Haltetesters bestehen also aus einem Programm; der Haltetester analy-

▷ siert es und gibt daraufhin eine Antwort wie »Dieses Programm wird halten« oder »Dieses Programm wird nicht halten«.

Nun schreiben wir ein zweites Programm; nennen wir es den »Haltetestverderber«. Auch dieses Programm bekommt als Eingabe irgendein Computerprogramm; dann bestimmt es mit Hilfe des Haltetesters, ob das vorgelegte Programm anhalten wird. Wenn das der Fall ist, dann – und nur dann – geht es in eine Endlosschleife.

Jetzt kommen wir zum kritischen Punkt. Wenn man in den Haltetestverderber eine Kopie seiner selbst eingibt, was passiert dann?

Zur Erinnerung: Wir haben den Haltetestverderber so geschrieben, dass er in eine Endlosschleife geht, wenn das im Test befindliche Programm anhält. Aber nun ist das im Test befindliche Programm selbst der Haltetestverderber. Also geht es, so wie es konstruiert ist, in eine Endlosschleife, wenn es anhält, das heißt aber doch, es hält gerade nicht an. Das ist ein offensichtlicher Widerspruch! Es hilft auch nichts, wenn man den gegenteiligen Ausgang annimmt. Denn hält das getestete Programm nicht, so wird das vom Haltetest gemeldet, und das Programm geht nicht in die Endlosschleife, hält also an. Das Paradoxon führte Turing zu dem Schluss, dass man

keinen universellen Haltetest konstruieren kann.

Interessant ist die Folgerung, die Turing daraus ableitete. Falls es unmöglich ist, im Voraus durch Berechnungen festzustellen, ob ein Programm anhalten wird oder nicht, dann gibt es auch keine Möglichkeit, dies im Voraus durch logisches Schließen zu erkennen. Es gibt kein formales Axiomensystem, mit dessen Hilfe sich entscheiden lässt, ob ein Programm schließlich anhält. Warum? Wenn man ein formales Axiomensystem in dieser Weise anwenden könnte, dann hätte man ja ein »mechanisches Verfahren« oder, was dasselbe ist, ein Computerprogramm, das entscheidet, ob ein vorgelegtes Programm anhält oder nicht, mit einem Wort: einen Haltetest. Aber den gibt es nicht, wie wir oben gesehen haben.

Man kann ein Programm schreiben, das genau dann anhält, wenn es nicht anhält. Dieses Paradoxon ist im Prinzip dasselbe, auf welches Gödel im Rahmen seiner zahlentheoretischen Untersuchungen gestoßen war. Allerdings hat Turing Gödels Aussage noch erheblich verallgemeinert.

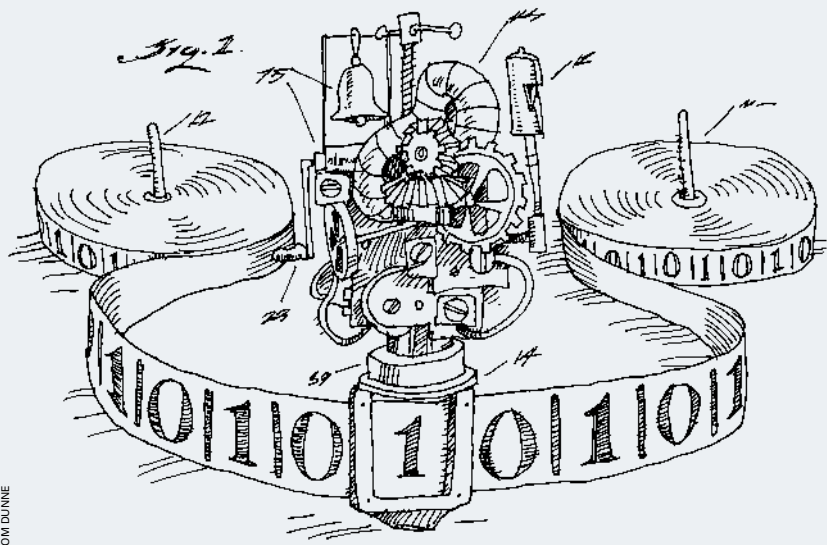
Der Zufall in der Mathematik

Nach dem Ausbruch des Zweiten Weltkriegs wandten sich die großen Geister sehr irdischen Anwendungen der Computer zu: Turing der Entschlüsselung geheimer Funksprüche des deutschen Kriegsgegners und von Neumann der Entwicklung der amerikanischen Atombombe. Darüber geriet die Unvollständigkeit formaler Axiomensysteme für eine Weile in Vergessenheit.

Mit dem Ende des Zweiten Weltkriegs verschwand die Generation der Mathematiker, die sich mit diesen tiefen philosophischen Grundlagenfragen der Mathematik beschäftigten hatten, im Wesentlichen von der Bildfläche. Und nun betrat der kleine Gregory Chaitin die Bühne des Geschehens.

In den späten 1950er Jahren las ich als Jugendlicher im Scientific American vom Juni 1956 einen Artikel über Gödel und die Unvollständigkeit. Gödels Resultat faszinierte mich, aber ich verstand es nicht wirklich und kam zu der Überzeugung, dass an der Sache etwas faul sei. Irgendetwas musste noch dahinter stecken. Dann las ich über Turings Resultat, das zweifellos tiefer ging als das

Die Turing-Maschine

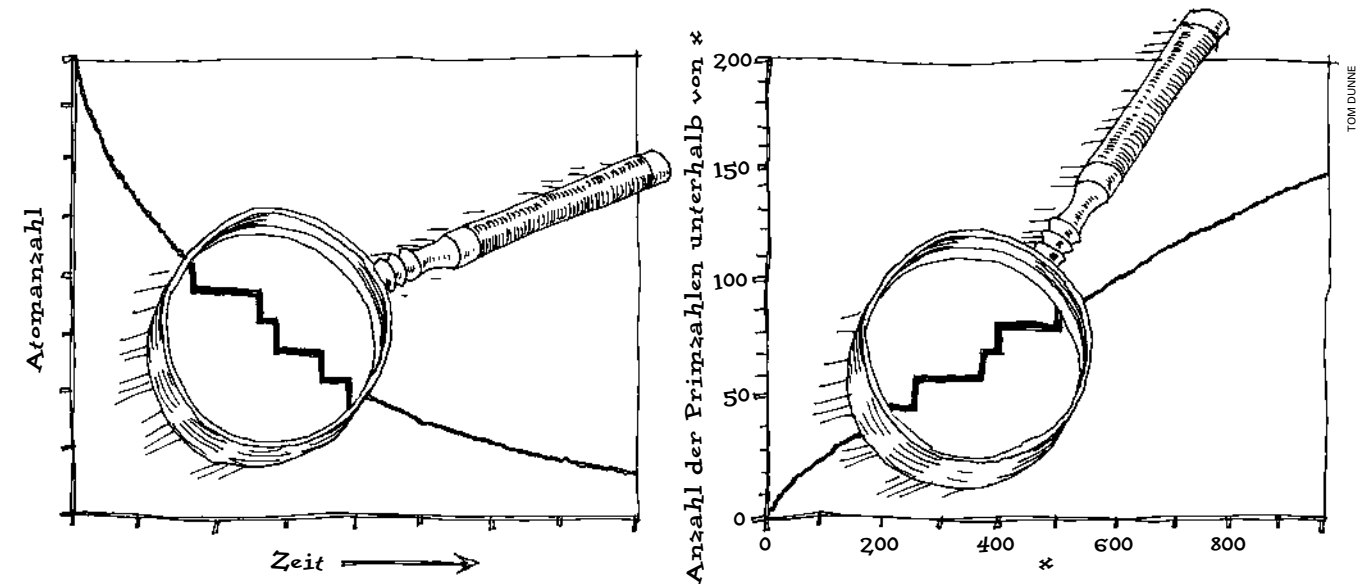


Die hypothetische Maschine, die Alan Turing in seiner wegweisenden Arbeit von 1936 einführte, arbeitet auf einem sehr langen Magnetband, dessen Felder je ein Binärzeichen (0 oder 1) tragen. In einem Arbeitsschritt liest die Maschine das Zeichen, das sich unter ihrem Schreib-/Lesekopf befindet. In Abhängigkeit von diesem Zeichen sowie von ihrem inneren Zustand schreibt sie ein neues Zeichen an derselben Stelle aufs Band – oder auch nicht –, rückt ein Feld nach rechts oder links auf dem Band und nimmt einen neuen inneren Zustand an. Dann beginnt der nächste Arbeitsschritt.

Zu Beginn des Arbeitsprozesses stehen auf dem Band die Eingabedaten.

Wenn die Maschine ihre Arbeit beendet hat, steht das Ergebnis der Berechnungen auf dem Band, die Maschine geht in den speziellen Zustand »fertig« über und hält an. Das Programm der Maschine besteht aus einer Tabelle, die zu jeder denkbaren Kombination aus gelesenen Zeichen und innerem Maschinenzustand die auszuführenden Aktionen ansagt: neues Zeichen auf dem Band, nach rechts oder nach links rücken, neuer innerer Zustand.

Turing zeigte, dass diese Maschine trotz ihrer Einfachheit und mit einer bescheidenen Anzahl innerer Zustände alles berechnen kann, was überhaupt berechenbar ist – wenn ausreichend Zeit und Speicherplatz zur Verfügung stehen.



von Gödel, aber ich war immer noch nicht zufrieden. So kam ich auf die verrückte Idee mit dem Zufall.

Ich hatte als Kind auch viel über andere wissenschaftliche Dinge gelesen, vor allem die Grundlagen der Physik: Relativitätstheorie, Kosmologie und ganz besonders Quantenmechanik. Dabei lernte ich, dass die physikalische Welt sich ziemlich verrückt verhält, wenn es um die ganz kleinen Objekte geht. In diesem mikroskopischen Bereich sind die Ereignisse grundsätzlich nicht vorhersagbar, das heißt, es regiert der Zufall (der echte und nicht nur unsere Unwissenheit). Daraufhin kam mir die Frage in den Sinn, wie es denn um den Zufall in der reinen Mathematik bestellt sei (Bild oben). Vielleicht lag ja gerade darin der wahre Grund für die Unvollständigkeit.

Vor allem die elementare Zahlentheorie konnte einen auf entsprechende Gedanken bringen. Denken wir nur an die Primzahlen. Ob eine bestimmte Zahl eine Primzahl ist oder nicht, erscheint ziemlich unvorhersehbar. Es gibt statistische Aussagen wie den so genannten Primzahlsatz, der die relative Häufigkeit von Primzahlen innerhalb eines großen Bereichs ziemlich genau beschreibt. Aber die Verteilung der Primzahlen im Einzelnen sieht schon sehr zufällig aus (siehe meinen Artikel in Spektrum der Wissenschaft 9/1988, S. 62).

Ich kam also auf die Idee, dass diese eng mit der Mathematik verwobene Zufälligkeit der tiefere Grund für die Unvollständigkeit sei. In den 1960er Jahren präsentierte ich ein paar neue Ideen, die ich als »algorithmische Informationstheorie« bezeichnen möchte. Unabhängig

von mir entwickelte Andrej N. Kolmogorow (1903–1987) in Moskau eine ähnliche Theorie. Hinter dem bombastischen Namen steckt ein sehr einfacher Grundgedanke. Es geht darum, die Komplexität einer Berechnung zu messen, das heißt den Aufwand, der für die Lösung eines Problems mindestens getrieben werden muss.

Das erste Mal hörte ich von der Idee der Komplexität bei von Neumann. Für Turing war der Computer ein abstraktes mathematisches Konzept gewesen, eine gedachte perfekte Maschine, die niemals Fehler macht und beliebig viel Zeit und Kapazität zur Verfügung hat. Der nächste logische Schritt für die Mathematiker war, zu untersuchen, wie viel Zeit eine Berechnung in Anspruch nimmt – eine Form von Komplexität. Um 1950 rückte von Neumann diese Laufzeitkomplexität ins Zentrum des Interesses und begründete damit ein heute weit entwickeltes Forschungsfeld.

Information und Entropie

Mich interessierte nicht in erster Linie die Rechenzeit, obwohl sie in der Praxis meistens die entscheidende Größe ist, sondern eher die Länge von Computerprogrammen, also die Menge von Information, die ein Computer braucht, um eine bestimmte Aufgabe zu erfüllen. Warum ist das interessant? Weil der Komplexitätsbegriff, der sich auf den Programmumfang bezieht, die »Informationskomplexität«, eng mit dem Entropiebegriff aus der Physik zusammenhängt.

Die Entropie spielt eine herausragende Rolle in den Arbeiten des berühmten

▷ Der Zufall herrscht in der Quantenmechanik. Der scheinbar gleichmäßig und stetig verlaufende Zerfall einer radioaktiven Substanz entpuppt sich bei genauerem Hinsehen als eine Folge diskreter Sprünge (links). Dabei ist nicht vorhersagbar, zu welchem Zeitpunkt das jeweils nächste Atom zerfällt. Die Anzahl der Primzahlen unterhalb einer vorgegebenen Grenze x ist grob betrachtet ebenfalls eine glatte, wohldefinierte Kurve. Bei genauerem Hinsehen entpuppt sich die Kurve als aus Sprüngen in unregelmäßigen Abständen zusammengesetzt (rechts). Der genaue Wert der jeweils nächstgrößeren Primzahl kann nicht aus einer allgemeinen Theorie hergeleitet werden.

Physikers Ludwig Boltzmann (1844–1906) über statistische Mechanik und Thermodynamik. Sie ist ein Maß für die Unordnung, das Chaos, den Zufall in einem physikalischen System. Die Entropie eines Kristalls ist gering, während ein Gas, etwa bei Zimmertemperatur, eine hohe Entropie aufweist.

Der Entropiebegriff steht im Zusammenhang mit der fundamentalen philosophischen Frage, warum die Zeit nur in eine Richtung läuft. Im täglichen Leben ist der Unterschied zwischen Vergangenheit und Zukunft unüberschaubar. Ein Glas zerbricht, aber ein Scherbenhaufen setzt sich nicht von selbst zum Glas zusammen. In der Boltzmann'schen Theorie wird das dadurch ausgedrückt, dass die Entropie immer nur zunehmen kann, das heißt, die Unordnung im System wird immer größer. Das ist der wohl

▷ bekannte Zweite Hauptsatz der Thermodynamik.

Boltzmanns Zeitgenossen sahen keine Möglichkeit, dieses Ergebnis aus der Newtonschen Physik herzuleiten. Schließlich ist in einem Gas, wo die Atome durcheinander wirbeln wie Billardkugeln, jedes Ereignis in der Zeit umkehrbar. Wenn es möglich wäre, für eine kurze Zeit zu filmen, was sich in einem kleinen Gasvolumen abspielt, könnte man hinterher nicht entscheiden, ob der Film vorwärts oder rückwärts läuft. Das steht im Widerspruch zu Boltzmanns Theorie und der täglichen Erfahrung, die besagen, dass die Zeit sehr wohl eine festgelegte Richtung hat: Ein System, das sich anfangs in einem sehr geordneten Zustand befindet, entwickelt sich im Verlauf der Zeit in einen sehr ungeordneten, chaotischen Zustand. In dem Fall, dass das System das ganze Universum ist, hat dieser Endzustand einen schrecklich dramatischen Namen: »Wärmetod«.

Interessanterweise gibt es nun einen Zusammenhang zwischen dem Grad von Unordnung in einem physikalischen System und der Größe eines Computer-

programms. Um zu beschreiben, wo sich alle Atome eines Gases befinden, braucht es ein sehr großes Programm, während für einen Kristall auf Grund seiner regulären Struktur ein sehr viel kleineres Programm ausreicht. Somit sind die Entropie und die Informationskomplexität eines Computerprogramms – die man heute Kolmogorow-Chaitin-Komplexität oder auch Kolmogorow-Komplexität nennt – eng miteinander verknüpft: Die Entropie eines Systems ist im Wesentlichen die Länge des kürzesten Programms, welches den Zustand des Systems beschreibt.

Dieser Begriff von Komplexität ist wiederum eng verknüpft mit der Philosophie der wissenschaftlichen Methode. Der Informatiker Ray Solomonoff hat das auf einer Tagung im Jahr 1960 zur Sprache gebracht; ich erfuhr davon erst einige Jahre später, als ich selbst schon zu ähnlichen Schlüssen gekommen war. Unter mehreren Theorien, die dasselbe leisten, ist stets die einfachste vorzuziehen. Dieses Prinzip ist als »Ockhams Rasiermesser« bekannt, denn schon der mittelalterliche Gelehrte Wilhelm von

Ockham (um 1285–1349) unternahm es, den Wildwuchs unter wissenschaftlichen Theorien auszulichten. Was ist nun in unserem Kontext eine Theorie? Ein Computerprogramm, das Beobachtungen voraussagt. Und die Aussage, dass die einfachste Theorie die beste ist, übersetzt sich in die Feststellung, dass unter mehreren Computerprogrammen gleicher Funktion das kürzeste die beste Theorie liefert.

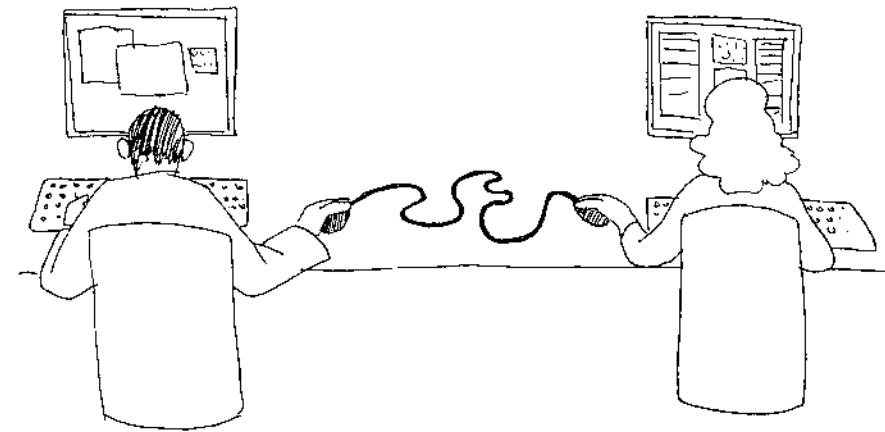
Zufall ist, wenn man es nicht kürzer sagen kann

Was ist, wenn es keine wirklich gute Theorie gibt? Wenn das kürzeste Programm, das zur Reproduktion einer bestimmten Menge von Beobachtungsdaten fähig ist, genauso groß ist wie die Datenmenge? Dann hilft keine Theorie; die Daten sind nicht komprimierbar, sondern zufällig. Die Qualität einer Theorie misst sich daran, inwieweit sie in der Lage ist, die Beobachtungsdaten zu komprimieren, das heißt durch eine wesentlich kleinere Menge von theoretischen Voraussetzungen und Deduktionsregeln zu ersetzen, aus denen die Daten rekonstruierbar sind.

Auf diese Weise kann man Zufall als etwas definieren, das nicht weiter komprimierbar ist. Die einzige Möglichkeit, ein vollkommen zufälliges Objekt zu beschreiben, besteht in der vollständigen Aufzählung aller seiner Daten. Da es keine Struktur gibt, ist eine kürzere Darstellung nicht möglich. Das andere Extrem ist ein Gegenstand mit einer sehr regulären Struktur, zum Beispiel die millionenfache Wiederholung der Ziffernfolge 01. Das ist ein sehr großes Objekt mit einer sehr kurzen Beschreibung.

Wenn man sich nun näher mit der Länge von Computerprogrammen beschäftigt – Informationskomplexität statt Laufzeitkomplexität –, geschieht etwas sehr Merkwürdiges: Wo man auch hinschaut, stößt man auf das Phänomen der Unvollständigkeit. Das geschieht schon mit der ersten Frage, die sich im Rahmen meiner Theorie stellt. Man

◀ Kurt Gödel und Einstein wurden in Princeton gute Freunde, und das, obgleich Einstein die entscheidende Rolle des Zufalls in der Physik und in der Mathematik, die Gödels Arbeiten zu erkennen halfen, nie akzeptiert hat.



misst ja die Komplexität eines Gegenstands durch die Länge des kürzesten Programms, das noch zu seiner Beschreibung fähig ist. Aber wie kann man sicher sein, dass man wirklich das kürzeste Programm gefunden hat? Gar nicht! Erstaunlicherweise übersteigt die Aufgabe, das kürzeste Programm zu finden, die Möglichkeiten mathematischen Schließens.

Die Begründung ist etwas verwickelt, deshalb möchte ich an dieser Stelle nur das Endergebnis vorstellen. Es ist einer meiner Lieblingssätze über Unvollständigkeit: Hat man Axiome im Umfang von n Bits gegeben, so ist es nicht möglich zu beweisen, dass ein Programm das kürzeste ist, wenn es länger ist als n Bits. Genauer: Zu den Axiomen und Schlussregeln einer Theorie gehört ein Programm – Hilberts »mechanisches Verfahren« –, das die Korrektheit eines vorgelegten Beweises, sprich der Herleitung eines Satzes aus den Axiomen mittels der Schlussregeln, überprüft. Die Länge n (in Bits) dieses Programms ist die kritische Länge. Für ein Programm, das länger ist als n Bits, ist es nicht möglich zu beweisen, dass es das kürzestmögliche Programm zur Beschreibung eines bestimmten Gegenstands ist.

Es stellt sich also heraus, dass man im Allgemeinen die Informationskomplexität eines Gegenstands gar nicht berechnen kann, denn dazu müsste man die Länge des kürzesten Programms kennen, das den Gegenstand berechnet. Das ist unmöglich, wenn das Programm länger ist als die Informationsmenge n der Axiome. Und das ist praktisch immer der Fall, denn die Anzahl an Axiomen, die Mathematiker in der Regel benutzen, ist ziemlich übersichtlich, da sie sonst keiner glauben würde; n ist also eine relativ kleine Zahl.

Es gibt einen unendlich großen Kosmos mathematischer Wahrheiten – eine unendliche Menge an Information –; aber jedes gegebene Axiomensystem kann nur einen winzigen, endlichen Teil davon erfassen, einfach weil es zu klein ist. So gesehen ist Gödels Unvollständigkeitssatz keineswegs kompliziert und mysteriös, sondern im Gegenteil völlig natürlich und unvermeidlich.

Und wie geht's weiter?

In nur drei gedanklichen Schritten sind wir einen weiten Weg gegangen: von Gödel, der uns mit der schockierenden Tatsache konfrontiert, dass logisches Schließen seine Grenzen hat, über Turing, der uns diese Grenzen deutlich plausibler macht, bis zur Informations-

komplexität, die uns wiederum unausweichlich mit der Unvollständigkeit, der Begrenztheit jeglichen mathematischen Denkens, konfrontiert.

An dieser Stelle kommt meistens die Gegenfrage: »Das ist ja alles sehr schön, und die algorithmische Informationstheorie ist sicher eine gute Theorie; aber wo ist ein Beispiel für eine Aussage, die tatsächlich die Möglichkeiten mathematischen Schließens überfordert?« Viele Jahre lang habe ich auf diese Frage geantwortet: »Vielleicht Fermats letzter Satz.« Doch dann passierte es. Im Jahr 1993 präsentierte Andrew Wiles einen Beweis. Zunächst enthielt er einen Fehler, aber mittlerweile ist alle Welt davon überzeugt, dass er in seiner letzten Fassung korrekt ist. Zu dumm. Mit der algorithmischen Informationstheorie kann man zwar nachweisen, dass es eine Menge unbeweisbarer Sätze gibt, aber man bekommt keine Aussagen über konkrete mathematische Fragestellungen.

Der Unvollständigkeitssatz hat ja schon etwas Pessimistisches an sich. Wenn man ihn wörtlich nimmt, könnte man meinen, es sei in der Mathematik keinerlei Fortschritt möglich. Wie kommt es, dass die Mathematiker trotzdem so erfolgreich arbeiten? Vielleicht wird es der nächsten Generation junger Mathematiker gelingen, den Grund dafür zu finden. ◀



ARCHIV DES INSTITUTE OF ADVANCED STUDY



Gregory J. Chaitin ist Mathematiker am Watson-Forschungszentrum der IBM in Yorktown Heights (US-Bundesstaat New York) und Gastprofessor an den Universitäten Buenos Aires und Auckland. Seit 35 Jahren ist er der prominenteste Vertreter der algorithmischen Informationstheorie, die er als Jugendlicher erfand. Seine jüngste Weiterentwicklung der Theorie betrifft Prognosen über die Länge echter Computerprogramme.

Dieser Artikel ist ein Auszug aus seinem Buch »Conversations with a mathematician«, der seinerseits einen 1999 an der Universität von Massachusetts in Lowell gehaltenen Vortrag wiedergibt.

© American Scientist Magazine (www.americanscientist.org)

Kurt Gödel. Von Gianbruno Guerriero. Spektrum der Wissenschaft Biografie 1/2002

Der Gödelsche Beweis. Von E. Nagel und J. R. Newman. Oldenbourg, München 2001

Die Entdeckung des Unmöglichen. Von John D. Barrow. Spektrum Akademischer Verlag, Heidelberg 2001

Gödel, Escher, Bach. Von Douglas R. Hofstadter. 8. Auflage, dtv 2001

Das logische Dilemma. Leben und Werk von Kurt Gödel. Von W. Dawson jr. Springer, Heidelberg 1999

Zufall und Chaos. Von David Ruelle. Springer, Heidelberg 1992

Conversations with a mathematician. Von Gregory J. Chaitin. Springer, Heidelberg 2002

Exploring randomness. Von Gregory J. Chaitin. Springer, Heidelberg 2001

Gödel: a life of logic. Von John L. Casti und Werner DePauli. Perseus, Cambridge (Massachusetts) 2000

The unknowable. Von Gregory J. Chaitin. Springer, Heidelberg 1999

The limits of mathematics. Von Gregory J. Chaitin. Springer, Heidelberg 1998

Information, randomness and incompleteness. Von Gregory J. Chaitin. World Scientific, 1990

Algorithmic information theory. Von Gregory J. Chaitin. Cambridge University Press, 1987

Weblinks zu diesem Thema finden Sie bei www.spektrum.de unter »Inhaltsverzeichnis«.

AUTOR UND LITERATURHINWEISE