WAS KÖNNEN QUANTENCOMPUTER?

ERICH NOVAK

Mathematisches Institut, Universität Jena, Ernst-Abbe-Platz 2,
D-07740 Jena, novak@mathematik.uni-jena.de

Zusammenfassung. Quantencomputer benutzen seltsame Effekte der Quantenwelt, nämlich sogenannte „verschränkte Zustände“, mit deren Hilfe man die „Bellschen Ungleichungen“ verletzen kann. Dazu beschreiben wir zunächst ein Gedankenexperiment, das die Andersartigkeit der Quantenwelt eindrucksvoll vorführt. Dann wird die logische Struktur von Quantencomputern entwickelt. Dazu benötigt man nur Begriffe aus dem Grundstudium (lineare Algebra, unitäre Matrizen), insbesondere nur endlich dimensionale Vektorräume. Als Beispiel für die Überlegenheit von Quantencomputern diskutieren wir den Such-Algorithmus von Grover.

1. Quantencomputer benutzen Effekte der Quantenwelt

Alan Turing hatte schon im Jahr 1936 genaue Vorstellungen darüber, was Computer können, die klassisch arbeiten — also nicht die Effekte der Quantenwelt benutzen. Interessant ist, daß es im Jahr 1936 noch gar keine Computer gab. Turing hatte ein mathematisch-logisches Verständnis dafür, was Computer ganz prinzipiell leisten. Dazu mußte er wenig über die technisch-physikalische Realisierung solcher Maschinen wissen.

In gleicher Weise kann man heute ein mathematisches Modell eines Quantencomputers studieren und sich fragen, was solche Rechner können. Unabhängig davon, inwieweit sich dieses mathematische Modell technisch realisieren läßt.[1]

Quantencomputer benutzen die Effekte der Quantenwelt, und diese Quantenwelt ist sehr anders als die Welt, die wir kennen. Zunächst ist der mathematische Formalismus der Quantenmechanik anders und seltsam und man mag glauben, daß nur dieser Formalismus so fremdartig ist.

Aber es ist die Quantenwelt selbst, die seltsam ist. Dabei spielt der *Zufall* eine große Rolle. Fremdartig ist aber nicht so sehr der Zufall als solcher. Wir lernen zum Beispiel, daß der radioaktive Zerfall vom Zufall geregelt wird. Das ist ein einfacher Zufall, der sich verstehen läßt mit den klassischen Gesetzen der Wahrscheinlichkeitsrechnung.[2]

Wie der Zufall in der Quantenmechanik eine Rolle spielt, ist seltsam, und wir werden dazu ein *Gedankenexperiment* vorstellen, das auf Einstein, Podolsky und Rosen

[1] Denkbare Möglichkeiten zur Realisierung von Quantencomputern werden z.B. in Nielsen, Chuang (2000) dargestellt. Ich kann und will aber nicht über die technische Realisierbarkeit spekulieren. Mein Eindruck ist, daß zur Zeit niemand so recht weiß, ob in den nächsten 30 Jahren mit einem „brauchbaren“ Quantencomputer zu rechnen ist.

[2] Natürlich läßt sich der radioaktive Zerfall *auch* mit den Gesetzen der Quantenmechanik verstehen.

(1935) zurückgeht. Ähnliche Experimente werden seit Beginn der 1980er Jahre durchgeführt. Das Stichwort dazu lautet „Verletzung der Bell'schen Ungleichungen durch verschränkte Zustände".

2. Ein Gedankenexperiment

Charlie befindet sich zwischen Alice und Bob, die Entfernungen sind jeweils sehr groß, etwa eine Millarden Kilometer (oder eine Lichtstunde). Charlie schickt immer wieder ein Paar von Teilchen weg, jeweils ein Teilchen an Alice und an Bob. Alice bekommt ein (Q, R)-Teilchen und Bob ein (S, T)-Teilchen. Das heißt, daß Alice an ihrem Teilchen die Eigenschaft Q oder R messen kann, Bob an seinem Teilchen die Eigenschaften S oder T. Jede Messung einer der Eigenschaften Q, R, S oder T ergibt den Wert 1 oder -1.

Beide entscheiden unabhängig und kurzfristig (Münzwurf), welche Eigenschaft sie messen. Die Messungen geschehen ziemlich gleichzeitig, alle drei Personen ruhen und benutzen dieselbe Uhr. Man stellt sich vor, daß sich die Messungen nicht gegenseitig stören können. Betrachte die Größe

$$QS + RS + RT - QT = (Q + R)S + (R - Q)T = \pm 2.$$

Wiederholt man das Experiment, so gilt auch für die Durchschnittswerte oder Erwartungswerte

$$E(QS) + E(RS) + E(RT) + E(-QT) \leq 2. \tag{1}$$

Dies ist eine Form der *Bell'schen Ungleichung*. Was hat dies mit der Quantenmechanik zu tun? Die Quantenmechanik behauptet, daß Charlie die Teilchen so präparieren kann, daß alle 4 Erwartungswerte gleich $\sqrt{2}/2$ sind, das heißt, auch der Fall

$$E(QS) + E(RS) + E(RT) + E(-QT) = 2\sqrt{2}$$

kann auftreten. Dies wurde auch experimentell bestätigt, klassisch läßt sich dies nicht verstehen. Also wurde beim „Beweis" von (1) ein Fehler gemacht. Wir haben zwei Annahmen gemacht, die in der klassischen Physik klar gelten. In der Quantenmechanik ist mindestens eine dieser Annahmen falsch:

- Die Größen Q, R, S, T haben „wirkliche" Werte, unabhängig von der Messung (Realismus).
- Die beiden Messungen können sich nicht stören (Lokalität).

Diese Andersartigkeit der Quantenmechanik im Vergleich zur klassischen Physik (technisch: die Existenz von verschränkten Zuständen) kann man benutzen, um völlig neuartige Quanten-Algorithmen zu konstruieren. Dazu muß man einen Formalismus kennenlernen, der mathematisch gesehen nicht allzu kompliziert ist: endlich dimensionale Vektorräume, unitäre Matrizen, etwas Wahrscheinlichkeitsrechnung. Das lernt man zu Beginn eines Studiums, etwa der Informatik, Physik, oder Mathematik.[3]

[3]Die Standardauflösung dieses EPR-Paradoxons durch die Quantenmechanik geht so: Die Größe $QS + RS + RT - QT$ läßt sich nicht messen (wenn Alice Q mißt, dann verändert sie dabei das System und kann vom ursprünglichen System nicht mehr die Eigenschaft R messen). Daher macht es keinen Sinn, dieser Größe den Wert 2 oder -2 zuzuordnen. Nicht alle Physiker waren oder sind mit dieser Erklärung zufrieden.

3. Das mathematische Modell eines Quantencomputers

Wir skizzieren zunächst die Arbeitsweise eines Quantencomputers und stellen dann den Suchalgorithmus von Grover (1996) vor, der die Fähigkeiten eines Quantencomputers eindrucksvoll demonstriert.

Wir betrachten einen 2-dimensionalen Hilbertraum H_1 über $\mathbb{C}$ sowie zwei orthonormale Vektoren e_0 und e_1 in H_1. Der Raum H_1 dient zur Darstellung der möglichen Zustände eines Quantenbits. Um die Zustände von m Quantenbits beschreiben zu können, wird das 2^m-dimensionale Tensorprodukt

$$H_m = H_1 \otimes \cdots \otimes H_1$$

mit m Faktoren verwendet. Eine Orthonormalbasis in H_m ist gegeben durch die 2^m Vektoren

$$b_\ell = e_{i_1} \otimes \cdots \otimes e_{i_m}, \qquad \ell = 0, \ldots, 2^m - 1, \tag{2}$$

mit $i_1, \ldots, i_m \in \{0, 1\}$ und

$$\ell = \sum_{j=1}^{m} i_j \, 2^{m-j}.$$

Die formal verschiedenen Objekte $(i_1, \ldots, i_m)$ und ℓ oder b_ℓ werden oft identifiziert und heißen *klassische Zustände*. Sie entsprechen den 2^m möglichen Zuständen, die m klassische Bits annehmen können.

Jeder Vektor $x \in H_m$ besitzt bezüglich der gegebenen Basis eine eindeutige Darstellung

$$x = \sum_{\ell=0}^{2^m-1} \beta_\ell \, b_\ell \tag{3}$$

mit den Fourier-Koeffizienten $\beta_0, \ldots, \beta_{2^m-1} \in \mathbb{C}$, und es gilt

$$\|x\|^2 = \sum_{\ell=0}^{2^m-1} |\beta_\ell|^2.$$

Vektoren x der Länge $\|x\| = 1$ heißen *Quantenzustände*, und jeder solche Zustand definiert in natürlicher Weise eine Wahrscheinlichkeitsverteilung auf den klassischen Zuständen: die Wahrscheinlichkeit von b_ℓ ist $|\beta_\ell|^2$.

Ein *Quantenalgorithmus* ist durch eine endliche Folge von gewissen unitären Abbildungen

$$U_i : H_m \to H_m, \qquad i = 1, \ldots, r,$$

definiert. Die Berechnung startet mit einem klassischen Zustand $b_k \in H_m$ als Input, und dann werden sukzessive die Abbildungen U_i angewandt. Das mathematische Resultat der Berechnung ist der Quantenzustand[4]

$$x = U_r \cdots U_1(b_k). \tag{4}$$

[4]Jedem Quantenalgorithmus entspricht also eine Formel der Form (4). Man kann auch sagen, daß die Programmierung eines Quantencomputers sehr einfach ist, weil es nur sehr wenige Befehle gibt. Andererseits braucht man zur Entwicklung guter Algorithmen ein großes geometrisches Geschick: Man muß sich vorstellen, was für eine Abbildung eine Formel (4) im 2^m-dimensionalen Raum bewirkt.

Erlaubt sind die folgenden beiden Arten von unitären Transformationen U, aus denen sich durch Hintereinanderausführung wie in (4) jede beliebige unitäre Transformation erzeugen läßt:

- U ändert höchstens ein einziges Quantenbit, d.h.

$$U(e_{i_1} \otimes \cdots \otimes e_{i_j} \otimes \cdots \otimes e_{i_m}) = e_{i_1} \otimes \cdots \otimes \widetilde{U}(e_{i_j}) \otimes \cdots \otimes e_{i_m}$$

  mit einer unitären Abbildung $\widetilde{U} : H_1 \to H_1$ und $j \in \{1, \ldots, m\}$. Tatsächlich kann man hier noch sehr viel restriktiver sein und z.B. nur die Matrizen W_1 (siehe unten) und $T = \begin{pmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{pmatrix}$ zulassen.[5]
- U ist eine *controlled not transformation* d.h.

$$U(e_{i_1} \otimes \cdots \otimes e_{i_j} \otimes \cdots \otimes e_{i_k} \otimes \cdots \otimes e_{i_m}) = e_{i_1} \otimes \cdots \otimes e_{i_j} \otimes \cdots \otimes e_{i_k \oplus i_j} \otimes \cdots \otimes e_{i_m}$$

  mit $j \neq k$. Das Symbol $\oplus$ bezeichnet die Addition modulo 2. Befindet sich das j-te Quantenbit im Zustand e_0, so bleibt der Zustand des k-ten Quantenbits unverändert. Ansonsten wechselt der Zustand des k-ten Quantenbits von e_0 nach e_1 oder von e_1 nach e_0, wird also negiert (controlled not).

Jede dieser erlaubten unitären Operationen kostet eine Einheit. Die *Kosten* eines durch $U_r \cdots U_1$ beschriebenen Quantenalgorithmus betragen also r. Natürlich könnte man das Ergebnis x gemäß (4) auch mit einem klassischen Computer berechnen. Allerdings wären die Kosten dann nicht r, sondern viel höher, da die zu den Abbildungen U_i gehörigen Matrizen 2^m Zeilen und Spalten komplexer Zahlen besitzen.

Wichtig ist nun folgendes: Ein Quantenalgorithmus liefert als Output *nicht* das mathematische Resultat x, sondern einen klassischen Zustand, der durch eine physikalische Messung ermittelt wird. Das Resultat dieser Messung ist zufällig. Die Wahrscheinlichkeit, den klassischen Zustand b_ℓ bzw. $\ell \in \{0, \ldots, 2^m - 1\}$ zu erhalten, beträgt $|\beta_\ell|^2$, wobei β_ℓ der entsprechende Fourier-Koeffizient des Vektors x ist.

Mit einem besonders einfachen Quantenalgorithmus läßt sich der Befehl „Wähle ein Bit $x \in \{0, 1\}$ zufällig“ realisieren. Hierzu sei $m = 1$ und $W_1 : H_1 \to H_1$ die durch

$$W_1(e_i) = \frac{1}{\sqrt{2}}(e_0 + (-1)^i\, e_1), \qquad i = 0, 1,$$

definierte *Walsh-Hadamard-Transformation*. Insbesondere gilt $W_1(e_0) = 1/\sqrt{2} \cdot (e_0 + e_1)$, so daß der Algorithmus W_1 mit Input e_0 als Output 0 oder 1 liefert, und zwar jeweils mit Wahrscheinlichkeit 1/2. Dieser Quantenalgorithmus erzeugt also ein zufälliges Bit und kann als *idealer Zufallszahlengenerator* angesehen werden.

Der bereitgestellte Formalismus ist noch nicht ausreichend, um das Gedankenexperiment von Abschnitt 2 völlig zu verstehen. Dazu müßte man noch genauer diskutieren, was nach einer Messung passiert.[6] Daher wird hier nur folgendes gesagt: Charlie erzeugt den Quantenzustand

$$C = \frac{1}{\sqrt{2}}(e_0 \otimes e_1 - e_1 \otimes e_0).$$

[5]Legt man dieses restriktivere Modell zugrunde, so lassen sich beliebige unitäre Transformationen nicht mehr exakt, aber immer noch beliebig genau, realisieren.

[6]Lesern die tiefer eindringen wollen, empfehle ich besonders Albert (1992) und Audretsch (2005).

Dieser Zustand ist verschränkt, d.h. C ist nicht von der Form $C = v_1 \otimes v_2$. Charlie schickt das erste Bit an Alice, das zweite an Bob. Alice und Bob wenden (auf „ihr“ Teilchen) gewisse Messungen an und es ergeben sich die Effekte, die wir im Abschnitt 2 beschrieben haben. Mit welchen Teilchen diese „spukhaften Fernwirkungen“ erzeugt werden, ist im Prinzip nicht wichtig. Tatsächlich benutzt man meist polarisationsverschränkte Photonen. Der Abstand zwischen Alice und Bob beträgt beispielsweise 500 Meter, siehe Audretsch (2002).

4. Der Suchalgorithmus von Grover

Wir betrachten jetzt folgendes *Suchproblem*. Sei $f : \{0, \dots, N-1\} \to \{0, 1\}$ eine Funktion, die genau einmal den Wert 1 annimmt. Gesucht ist die Stelle ℓ mit $f(\ell) = 1$.

Jeder deterministische oder randomisierte Algorithmus, der das Problem für festes N mit hoher Wahrscheinlichkeit (etwa 3/4) löst, muß eine zu N proportionale Anzahl von Funktionsauswertungen vornehmen. Der Quantenalgorithmus von Grover löst das Problem hingegen für festes N zu Kosten, die nur proportional zu $N^{1/2} \cdot \ln N$ sind. Durch die Verwendung des Quantenalgorithmus werden die Kosten also wesentlich reduziert.

Der Einfachheit halber nehmen wir im folgenden $N = 2^m$ an. Der Algorithmus von Grover arbeitet dann mit m Quantenbits und benutzt ein *Quanten-Orakel* $Q_f : H_m \to H_m$ zur Auswertung der Funktion f. Die unitäre Abbildung Q_f ist dabei durch

$$Q_f(b_\ell) = (-1)^{f(\ell)} \cdot b_\ell$$

für die Basis gemäß (2) definiert. Man stellt sich vor, daß der Operator Q_f als „black box“ zur Verfügung steht, und legt die Kosten für den Aufruf des Quanten-Orakels als $\mathbf{c} > 0$ fest.

Weiter bezeichne W_m das m-fache Tensorprodukt der Walsh-Hadamard-Transformation W_1, d.h. $W_m(e_{i_1} \otimes \cdots \otimes e_{i_m}) = W_1(e_{i_1}) \otimes \cdots \otimes W_1(e_{i_m})$, und Q_0 sei der unitäre Operator mit $Q_0(b_0) = -b_0$ sowie $Q_0(b_\ell) = b_\ell$ für $\ell = 1, \dots, 2^m - 1$. Der Suchalgorithmus von Grover ist dann gegeben durch

$$x = \big(-W_m Q_0 W_m Q_f\big)^k \big(W_m(b_0)\big). \tag{5}$$

Boyer, Brassard, Høyer, Tapp (1998) zeigen, daß der Algorithmus das Suchproblem für $N = 2^m$ mit Wahrscheinlichkeit $1 - 2^{-m}$ löst, falls

$$k = \lfloor \pi/4\theta \rfloor$$

mit

$$\sin\theta = 2^{-m/2}$$

gewählt wird, siehe auch Nielsen, Chuang (2000, Sec. 6.1). Die Anzahl k der Aufrufe des Orakels Q_f ist proportional zu $N^{1/2}$. Die Rechenkosten jeder Iteration in (5) betragen ungefähr $\ln N$, so daß sich die Kosten des Algorithmus bis auf eine Konstante zu $N^{1/2} \cdot \ln N$ ergeben.

Wir wollen das soeben genannte Ergebnis über den Algorithmus (5) hier nicht vollständig beweisen, obwohl dazu nur Hilfsmittel der Linearen Algebra nötig sind.

Wir wollen aber festhalten, was zu zeigen ist: Sei $f = f_\ell : \{0, \dots, N-1\} \to \{0,1\}$ durch $f_\ell(\ell) = 1$ und $f_\ell(j) = 0$ für $j \neq \ell$ gegeben. Dann gilt für x gemäß (5)

$$|\langle x, b_\ell \rangle|^2 \geq 1 - 2^{-m},$$

d.h. der klassische Anfangszustand b_0 wird „fast“ auf den Vektor b_ℓ gedreht. Dies zeigt man so: Sei

$$z = \frac{1}{\sqrt{N-1}} \sum_{k \neq \ell} b_k$$

die gleichmäßige Überlagerung aller Nicht-Lösungen. Dann sind b_ℓ und z zueinander orthogonal. Bereits im ersten Schritt des Grover-Algorithmus wird b_0 auf eine Linearkombination von b_ℓ und z abgebildet, nämlich auf

$$W_m(b_0) = s = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} b_k,$$

und man verbleibt dann stets in diesem 2-dimensionalen Raum. Eine Iteration der Abbildungen $-W_m Q_0 W_m Q_f$ bewirkt eine Drehung um den Winkel 2α von z in Richtung b_ℓ. Dabei ist α der Winkel zwischen z und s. Man kann auch schreiben

$$s = b_\ell \cdot \sin\alpha + z \cdot \cos\alpha$$

und nach k Iterationen ergibt sich der Vektor

$$x_k = b_\ell \cdot \sin((2k+1)\alpha) + z \cdot \cos((2k+1)\alpha).$$

Setzt man nun k wie oben angegeben, so ergibt sich fast $x_k = b_\ell$.

Es ist bekannt, daß der Grover-Algorithmus in folgendem Sinne optimal ist. Will man das Suchproblem mit Wahrscheinlichkeit mindestens $1/2$ lösen, so benötigt man eine Anzahl von Orakelaufrufen, die proportional zu $N^{1/2}$ ist. Siehe Nielsen, Chuang (2000, Sec. 6.6).

5. Das Ergebnis von Shor

Shor hat im Jahr 1994 ein Ergebnis gezeigt, das sehr großes Aufsehen erregt hat. Er hat dafür im Jahr 1998 den Nevanlinna-Preis erhalten, den man als „Nobelpreis für die theoretische Informatik“ ansehen kann. Er wird nur alle 4 Jahre vergeben.

Shor hat gezeigt, daß man eine n-stellige Zahl mit einem Quantencomputer in einer Zeit, die proportional zu n^3 ist, in Primfaktoren zerlegen kann. Für dieses Ergebnis interessieren sich auch die Geheimdienste dieser Welt, weil die Verschlüsselung von Daten in der Regel darauf beruht, daß man keinen Weg (für klassische Computer, andere gibt es derzeit nicht) kennt, große Zahlen schnell (polynomial in n) in Primfaktoren zu zerlegen.

6. Berechnung von hochdimensionalen Integralen

Bisher wurden Quantencomputer meist für Probleme der diskreten Mathematik betrachtet, mein eigenes Gebiet ist die numerische Mathematik. Es ist bekannt, daß man hochdimensionale Integrale mit klassischen Computern und deterministischen Algorithmen nur sehr langsam berechnen kann.

Eine gewisse Abhilfe bieten randomisierte Algorithmen (Monte-Carlo-Verfahren): Mit Hilfe von Zufallszahlen lassen sich Integrale viel schneller berechnen. Man kann nun zeigen, daß man mit Hilfe von Quantencomputern hochdimensionale Integrale noch schneller berechnen kann oder könnte, siehe Novak (2001).

In letzter Zeit hat insbesondere Stefan Heinrich (Kaiserslautern) die Quanten-Komplexität von numerischen Problemen (Berechnung von Integralen, Lösung von Differentialgleichungen) untersucht. In der Arbeit Heinrich (2002) wird auch das entsprechende Berechenbarkeitsmodell genau beschrieben.

Ausgangspunkt für all diese Überlegungen ist wieder der Grover-Algorithmus, der znächst zu einem Zählalgorithmus erweitert wird: Für eine Boolesche Funktion $f : \{0, \dots, N-1\} \to \{0, 1\}$ ist

$$S(f) = \sum_{i=0}^{N-1} f(i)$$

(näherungsweise) gesucht, wobei N sehr groß ist. Dieses Problem wurde von Brassard et al. (2002) und Kwas, Woźniakowski (2004) studiert. Der Zählalgorithmus kann dann wiederum für die Numerische Integration (und für andere Probleme der Numerischen Mathematik) angewendet und modifiziert werden. Eine sehr gute Einführung in diesen Themenkreis bietet Heinrich (2004).

7. Epilog

Oft hat die Mathematik die Welt theoretisch vorweggenommen: Ich habe gesagt, daß Alan Turing schon 1936 wußte, was klassische Computer leisten können. Solche Beispiele gibt es viele. Dennoch kann letztlich erst die technische Entwicklung zeigen, inwieweit sich die theoretische Idee eines Quantencomputers realisieren läßt.

Vorerst haben Quantencomputer eher einen theoretischen Nutzen. Sie helfen uns einerseits, die Quantenmechanik besser zu verstehen. Andererseits verhelfen sie uns zu einem besseren Verständnis der randomisierten Algorithmen: Konstruktionsprinzipien für Quantenalgorithmen wurden in jüngster Zeit auch dafür benutzt, um neue Monte-Carlo-Methoden zu finden. So hat die durchaus spekulative Beschäftigung mit dem Quantencomputer eine Bedeutung selbst dann, wenn wir z.Z. nicht wissen, ob Quantencomputer in nächster Zeit technisch realisierbar sind.

Literatur

1. D. Z. Albert (1992): Quantum Mechanics and Experience. Harvard University Press.

2. J. Audretsch (2002): Verschränkte Welt. Wiley-VCH, Weinheim.

3. J. Audretsch (2005): Verschränkte Systeme. Die Quantenphysik auf neuen Wegen. Wiley-VCH, Weinheim.

4. M. Boyer, P. Brassard, P. Høyer, A. Tapp (1998): Tight bounds on quantum searching. Fortschr. Phys. **46**, 493–505.

5. G. Brassard, P. Høyer, M. Mosca, A. Tapp (2002): Quantum amplitude amplification and estimation. In: Quantum computation and information, 53–74, Contemp. Math. 305, Amer. Math. Soc., Providence, RI, 2002.

6. S. Heinrich (2002): Quantum summation with an application to integration. J. Complexity **18**, 1–50.

7. S. Heinrich (2004): Quantum complexity of numerical problems. In: Foundations of computational mathematics: Minneapolis, 2002, 76–95, London Math. Soc. Lecture Notes Ser., 312, Cambridge Univ. Press, Cambridge, 2004.

8. M. Kwas, H. Woźniakowski (2004): Sharp error bounds on quantum Boolean summation in various settings. *J. Complexity* **20**, 669–698.

9. M. A. Nielsen, I. L. Chuang (2000): Quantum Computation and Quantum Information. Cambridge University Press.

10. E. Novak (2001): Quantum complexity of integration. J. Complexity **17**, 2–16.

11. P. W. Shor (1997): Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comp. **26**, 1484–1509.

12. A. M. Turing (1936): On computable numbers, with an application to the Entscheidungsproblem. Proc. Lond. Math. Soc., II. Ser. **42**, 230–265.