Proseminar/Seminar **Quantum Computing**
bei Prof. Dr. Th. Beth, Dr. D. Janzing, Dipl.–Inform. M. Grassl,
Dipl.–Inform. M. Rötteler, Dipl.–Inform. P. Wocjan

Institut für Algorithmen und Kognitive Systeme
Fakultät für Informatik, Universität Karlsruhe (TH)

Sommersemester 1999

Proseminarvortrag

Faktorisierungsalgorithmus von Shor

Stefan Röhrich, stefan@roehri.ch

August 1999

Inhaltsverzeichnis

1 Einleitung

Einer der Algorithmen, der Quantum Computing auch außerhalb der direkt damit befaßten Forscher bekannt machte, war der Faktorisierungsalgorithmus von Shor, da dieser (wenn er denn einmal realisiert wird) dazu benutzt werden kann, viele heute gängige Kryptosysteme wie z. B. RSA zu brechen.

Die besten heute bekannten klassischen Algorithmen (Number field sieve) zur Zerlegung einer Zahl n in ihre Primfaktoren haben eine Laufzeit von $O(e^{(\log n)^{1/3}(\log\log n)^{2/3}})$, während Shors Faktorisierungsalgorithmus auf einem Quantencomputer nur $O((\log n)^2(\log\log n)(\log\log\log n))$ (und zusätzlich $O(\log n)$ klassische Nachbearbeitung) benötigt. Er beruht auf folgenden Grundgedanken:

- Man kann mittels eines probabilistischen Algorithmus die Faktorisierung auf die Bestimmung der Ordnung in $\mathbb{Z}_n$ (Miller 1976) reduzieren.
- Der Funktionsgraph von $a \mapsto x^a \bmod n$, den man zur Bestimmung der Ordnung benötigt, wird durch Quantenparallelismus gewonnen, so daß der aufwendigste Schritt in diesem Verfahren die Eigenschaften des Quantencomputers sehr gut ausnutzt.
- Die Periodenbestimmung dieses Graphen, d. h. das Herausrechnen des durch die Modulo-Operation entstandenen Offsets, wird mittels der auf Quantencomputern ebenfalls effizient durchführbaren diskreten Fourier-Transformation erreicht.

Diese Ausarbeitung des Promseminar-Vortrages beruht hauptsächlich auf [Sho97] (nahezu identisch mit [Sho94]), außerdem wurden einige Hinweise aus [EJ94] und [Ber97] ergänzt.

Betreut wurde ich von Pawel Wocjan, bei dem ich mich dafür herzlich bedanken möchte.

2 Reduktion der Faktorisierung auf Bestimmung der Ordnung

2.1 Vorgehensweise

Die Ordnung r eines Elementes $x \in \mathbb{Z}_n$ ist die kleinste natürliche Zahl, so daß

$$x^r \equiv 1 \bmod n. \tag{1}$$

Die Faktorisierung von n wird nach Miller wie nachfolgend aufgeführt auf die Bestimmung der Ordnung zurückgeführt. Zunächst seien $n = pq$, $p, q \neq 2$ und Primzahlen (ansonsten schrittweise Vorgehensweise, falls $n = p_1 \cdot \ldots \cdot p_k$).

Zuerst bestimmt man die Ordnung r eines zufällig gewählten x ($\in \{2, \ldots, n-1\}$). Dies ist (bis jetzt) klassisch nicht effizient möglich und hierzu wird auch bei Shors Algorithmus der Quantencomputer verwendet.

Nun erhält man die Primfaktoren von n durch $p = \mathrm{ggT}(x^{r/2} - 1, n)$ und $q = \mathrm{ggT}(x^{r/2} + 1, n)$, denn

$$(x^{r/2} - 1)(x^{r/2} + 1) = x^r - 1 \equiv 0 \bmod n \tag{2}$$

Falls r ungerade (dann ist $x^{r/2}$ undefiniert) oder $x^{r/2} \equiv -1 \bmod n$ (es ergeben sich die trivialen Faktoren als Lösung) ist, funktioniert diese Methode allerdings offensichtlich nicht, man muß es dann erneut mit einem anderen x versuchen.

Daß man solche Zahlen findet, kann man sich auch so verdeutlichen:

Die Gleichung

$$x^2 \equiv 1 \bmod n \tag{3}$$

hat immer die trivialen Lösungen $x \equiv \pm 1 \bmod n$. Wenn n eine Primzahl ist, dann sind dies die einzigen Lösungen, anderenfalls gibt es weitere Lösungen. Betrachten wir dazu folgende Gleichungen:

$$\begin{aligned}
x_1 &\equiv 1 \bmod p && (4)\\
x_1 &\equiv 1 \bmod q \\
x_2 &\equiv -1 \bmod p && (5)\\
x_2 &\equiv -1 \bmod q \\
x_3 &\equiv 1 \bmod p && (6)\\
x_3 &\equiv -1 \bmod q \\
x_4 &\equiv -1 \bmod p && (7)\\
x_4 &\equiv 1 \bmod q.
\end{aligned}$$

In jedem Fall ist $x_i^2 \equiv 1 \bmod p$ und $\bmod q$, so daß (3) erfüllt ist. Nach dem Chinesischen Restsatz ([DW95]) hat jede dieser Gleichungen eine eindeutige Lösung $\bmod n$ und wir erhalten aus den beiden letzten Gleichungen $x_3 = a$ und $x_4 = -a \bmod n$ ein Paar nichttrivialer Lösungen. So ist $(a + 1)(a - 1) \equiv 0 \bmod n$ und $a \pm 1 \neq 0$ und somit teilt n $(a + 1)(a - 1)$ aber nicht $a \pm 1$ und der größte gemeinsame Teiler von n und $a \pm 1$ ist ein nichttrivialer Faktor von n.

2.2 Wahrscheinlichkeit für erfolgreiche Reduktion

Die Wahrscheinlichkeit, daß r ungerade oder $x^{r/2} \equiv -1 \bmod n$ ist, also daß mittels der Reduktion für ein x keine nichttrivialen Faktoren ermittelt werden können, ist kleiner $1/2$ (bzw. $1/2^{k-1}$ mit k als Anzahl der unterschiedlichen ungeraden Primfaktoren von n).

Seien p und q die Primfaktoren von n und $r_p =: 2^i u$ die Ordnung von $x \bmod p$ bzw. $r_q =: 2^j v$ von $x \bmod q$ (mit i, j maximal), damit ist $r \equiv \mathrm{kgV}(r_p, r_q) \bmod n$.

Der Algorithmus funktioniert nicht, falls $i = j$, denn dann ist entweder r ungerade ($i = j = 0$) oder $x^{r_p/2} \equiv -1 \bmod p$ und $x^{r_q/2} \equiv -1 \bmod q$ ($\mathbb{Z}_p$ bzw. $\mathbb{Z}_q$ sind zyklische Gruppen), so daß $x^r \equiv -1 \bmod n$.

Nach dem Chinesischen Restsatz können wir anstatt x zu wählen auch x_p und x_q wählen, so daß x sich aus $x \equiv x_p \bmod p$ und $x \equiv x_q \bmod q$ ergibt. Da p und q Primzahlen sind, sind die multiplikativen Gruppen $\bmod p$ bzw. $\bmod q$ zyklisch, so daß $\mathrm{P}(i = j) < 1/2$.

Diese Argumentation funktioniert analog, falls $n = p_1 \cdot \ldots \cdot p_k$.

2.3 Beispiel für $n = 3 * 5 = 15$

Dies wird nun noch einmal an einem kleinen Zahlenbeispiel verdeutlicht (um die Zahlen nicht zu groß werden zu lassen, wurde hier für den Fall, daß die Gewinnung der Primfaktorzerlegung aus der Ordnung nicht funktioniert, die etwas ungünstige Zahl $14 \equiv -1 \bmod 15$ gewählt).

	$\bmod 15$		$\bmod 3$		$\bmod 5$	
i	2^i	14^i	2^i	14^i	2^i	14^i
0	1	1	1	1	1	1
1	2	14	2	2	2	4
2	4	1	1	1	4	1
3	8	14	2	2	3	4
4	1	1	1	1	1	1
5	2	14	2	2	2	4
Ordnung	4	2	2	2	4	2

$$(2^{4/2} - 1)(2^{4/2} + 1) = 3 * 5$$

$$(14^{2/2} - 1)(14^{2/2} + 1) = 13 * 15$$

Abbildung 1: Beispiel für Reduktion Faktorisierung auf Ordnungsbestimmung

Wie man sieht, ist die Ordnung von $14 \bmod 3$ und $\bmod 5$ jeweils 2, so daß $14^{2/2} \equiv -1 \bmod 15$ und sich somit die triviale Lösung ergibt.

3 Realisierung beim Quantencomputer

3.1 Übersicht

Zur Bestimmung der Ordnung werden durch einen Quantencomputer folgende Schritte durchgeführt (q sei 2er-Potenz mit $n^2 \leq q < 2n^2$):

- Herstellung der Gleichverteilung der Funktionsargumente ($a = 0, \ldots, q - 1$ in Superposition)
- Berechnung des Funktionsgraphen $a \mapsto x^a \bmod n$ (Quantenparallelismus)

- Anwendung der diskreten Fouriertransformation auf $\mathbb{Z}_q$
- Messung

Der entsprechende Quantenschaltkreis sieht dabei schematisch wie folgt aus, wobei H die bekannte Hadamard-Matrix darstellt, E die modulare Exponentation, DFT die diskrete Fouriertransformation auf $\mathbb{Z}_q$ und B die Messung zur Standardbasis symbolisieren.

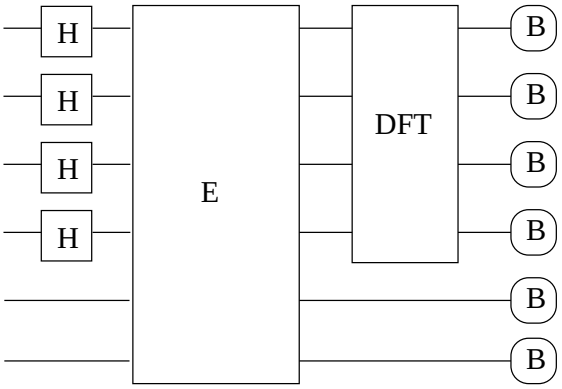

Abbildung 2: Schema des Quantenschaltkreises

3.2 Herstellung der Gleichverteilung

Jedes Bit des ersten Registers wird mittels einer Hadamard-Transformation auf $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ gesetzt, um die Zahlen $a \bmod q$ zu repräsentieren:

$$H_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{8}$$

Für mehrere Bits ergibt sich somit

$$H_n = \otimes_n H_1. \tag{9}$$

Dies läßt sich gut rekursiv implementieren:

$$H_n = \begin{pmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{pmatrix}. \tag{10}$$

Somit erhält man, wenn man diese Transformation auf ein $x \in \{0,1\}^n$ anwendet

$$H_n|x\rangle \mapsto |\varphi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{x \cdot i} |i\rangle \tag{11}$$

($\cdot$ bezeichne hier das Standardskalarprodukt über $\mathbb{F}_2^n$) und unser Quantencomputer zur Faktorisierung ist nach der Anwendung der Hadamard-Transformation nun in folgendem Zustand:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|0\rangle. \tag{12}$$

3.3 Modulare Exponentation

Unsere Transformationen bei einem Quantencomputer müssen unitär, also insbesondere reversibel sein, deshalb führen wir unseren Eingabewert a einfach in einem zusätzlichen Register weiterhin mit und legen den Funktionswert in einem weiteren, vorher nur mit $|0\rangle$ belegtem Register ab:

$$|a\rangle|0\rangle \mapsto |a\rangle|x^a \bmod n\rangle. \tag{13}$$

Die Modulare Exponentation ist effizient durchführbar, z. B. mittels Square-and-Multiply und Schönhage-Strassen für die dabei auftretenden Multiplikationen mit $O(l^2 \log l \log\log l)$ Zeit- und $O(l \log l \log\log l)$ Speicheraufwand oder mit „gewöhnlicher“ Multiplikation mit einem Aufwand von $O(l^3)$ an Zeit und $O(l)$ an Speicher (l bezeichnet hierbei eine l-Bit-Zahl). Trotzdem stellt sie die langsamste Operation des Faktorisierungsalgorithmus dar. Allerdings ist es interessant, daß der Schönhage-Strassen-Multiplikationsalgorithmus die schnelle Fourier-Transformation benutzt, die die Grundlage für viele Quantenalgorithmen darstellt. So ist es vielleicht denkbar, daß dadurch sogar die Ganzzahl-Multiplikation auf einem Quantencomputer beschleunigt werden könnte, was zu einem besseren asymptotischem Verhalten des Quantenfaktorisierungsalgorithmus führen würde und u. U. sogar das Brechen von RSA auf einem Quantencomputer asymptotisch schneller machen könnte als die RSA-Verschlüsselung auf einem klassischen Computer (näheres zur schnellen Implementation des Shor-Algorithmus wird in [Zal98] geschildert, dort findet sich auch eine Variante, die die FFT zur Multiplikation verwendet).

Diese klassischen Algorithmen lassen sich als Quantengatter realisieren, wobei darauf geachtet werde muß, daß die verwendeten Operationen reversibel sind. Insbesondere muß dabei ein Zwischenspeicher wieder auf 0 gesetzt werden, dies kann aber nicht direkt geschehen, sondern kann nur über einen Umweg erreicht werden. Um nun zu überprüfen, ob die bisherige Rechnung korrekt war, kann man nun diesen Zwischenspeicher messen, wird ein Wert ungleich 0 gemessen, wiederholt man die Rechnung, anderenfalls ist man zudem sicher, daß jede Amplitude ungleich 0 durch die Messung zerstört wurde und kann somit u. U. sogar die Genauigkeit erhöhen. Diese Methode heißt „quantum watchdog“- oder auch „quantum Zeno“-Effekt. Da die Messung aber auch einen Aufwand darstellt, müßte noch getestet werden, ob der Gewinn an Genauigkeit an dieser Stelle eine Geschwindigkeitsverbesserung des gesamten Algorithmus bewirkt.

Man erhält nun folgenden Zustand nach Anwendung der Funktion:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle |x^a \bmod n\rangle \tag{14}$$

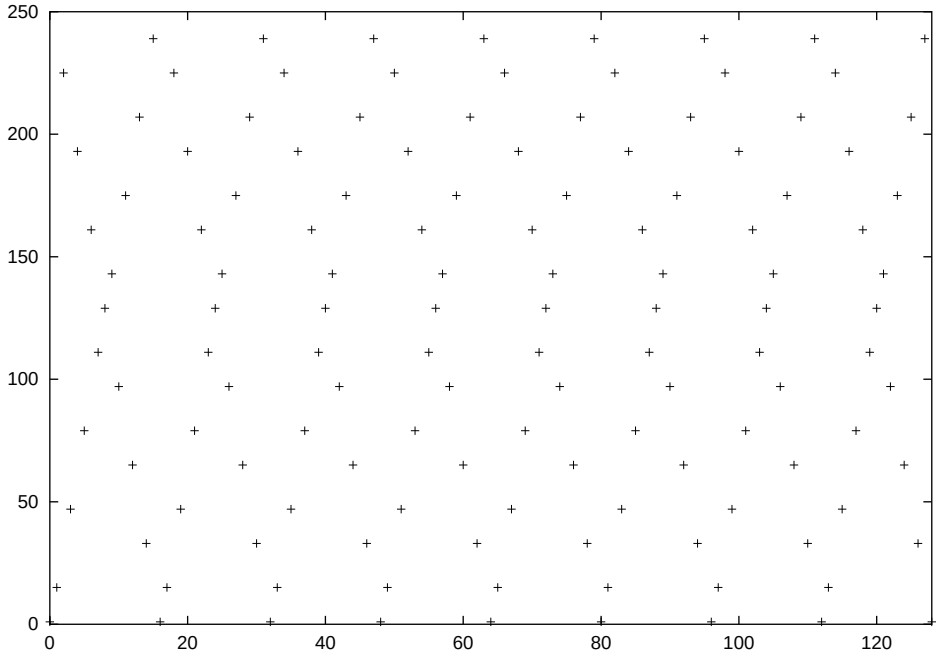

Abbildung 3: Beispiel eines Funktionsgraphen der modulo-Exponentation für $15^a \bmod 63$ ($q = 128$)

3.4 DFT$_q$

Als nächsten wichtigen Schritt führen wir nun die diskrete Fouriertransformation $\bmod q$ auf unserem ersten Register aus, die a ($0 \leq a < q$) wie folgt abbildet:

$$|a\rangle \mapsto \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} |c\rangle e^{2\pi i ac/q}. \tag{15}$$

Das heißt, wir wenden die unitäre Abbildung auf a an, die der Matrix mit den (a, c)-Einträgen $\frac{1}{\sqrt{q}} e^{2\pi i ac/q}$ entspricht. Diese Transformation ist auf einem Quantencomputer effizient durchführbar (DFT_{2^n} in $O(n^2)$ möglich, im Gegensatz dazu benötigt man dabei auf einem

klassischen Computer mittels der Fast Fourier Transformation $O(2^n n)$ Schritte) und Thema eines eigenen Vortrages.

Wir erhalten nun den folgenden Zustand:

$$\frac{1}{q}\sum_{a=0}^{q-1}\sum_{c=0}^{q-1} e^{2\pi iac/q}|c\rangle|x^a \bmod n\rangle. \tag{16}$$

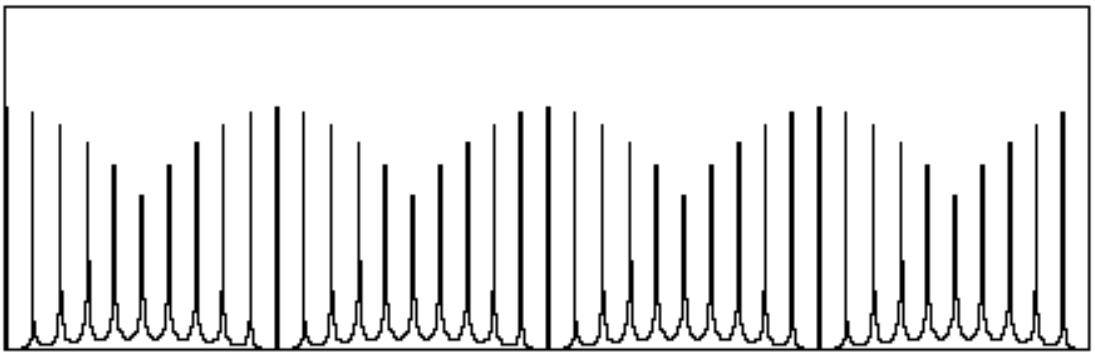

Abbildung 4: Ergebnis der Anwendung der DFT$_q$ für $n = 187, q = 804$

4 Auswertung des Ergebnisses der DFT

Als letzten Schritt in unserem Quantencomputer messen wir nun den gewonnen Zustand unseres Systems (eigentlich würde es genügen, $|c\rangle$ zu messen, wir messen aber jetzt $|c\rangle$ und $|x^a \bmod q\rangle$), und rechnen danach klassisch weiter.

Die diskrete Fouriertransformation wird bei diesem Algorithmus dazu benötigt, die Periode im Funktionsgraphen der modularen Exponentation zu ermitteln, die unserer gesuchten Ordnung r entspricht. Dabei wird praktisch der „Offset“ k eines erhaltenen Wertes $f(k + br)$ ausgelöscht und die Periode finden wir in der gemessenen Amplitude wieder (allerdings als Kehrwert und mit einem Faktor versehen), zusätzlich tritt eine Phasenverschiebung auf, da wir aber sowieso nur die Amplitude messen, können wir diese ignorieren.

Die Wahrscheinlichkeit, einen bestimmten Zustand $|c\rangle$ und $|x^k \bmod n\rangle$ $(0 \leq k < r)$ zu beobachten, beträgt:

$$\left|\frac{1}{q}\sum_{a:x^a\equiv x^k} e^{2\pi iac/q}\right|^2. \tag{17}$$

Da x nun aber die Ordnung r besitzt, können wir diese Summe umschreiben, indem wir $a = br + k$ setzen und erhalten:

$$\left| \frac{1}{q} \sum_{b=0}^{\lfloor (q-k-1)/r \rfloor} e^{2\pi i (br+k)c/q} \right|^2 . \tag{18}$$

Wir können nun $|e^{2\pi ikc/q}| = 1$ ausklammern und rc durch $rc \bmod q$ ersetzen, wobei dieser Rest noch so „verschoben“ wird, daß er im Intervall $(-q/2, q/2]$ liegt, wir schreiben dafür $\{rc\}_q$ und erhalten

$$\left| \frac{1}{q} \sum_{b=0}^{\lfloor (q-k-1)/r \rfloor} e^{2\pi i b \{rc\}_q/q} \right|^2 . \tag{19}$$

Die Summe können wir nun als Integral schreiben ([Heu98] Eulersche Summenformel) und gewinnen daraus

$$\frac{1}{q} \int_{b=0}^{\lfloor (q-k-1)/r \rfloor} e^{2\pi i b \{rc\}_q/q} db + O\left(\frac{\lfloor (q-k-1)/r \rfloor}{q} (e^{2\pi i \{rc\}_q/q} - 1) \right). \tag{20}$$

Wenn $|\{rc\}_q| \leq r/2$ (dies ist unabhängig von k!) ist obiger Fehlerterm durch $O(1/q)$ beschränkt und das Integral, also die Wahrscheinlichkeit den Zustand $|c, x^k \bmod n\rangle$ zu messen, ist groß. Dies kann man sich z. B. am komplexen Einheitskreis veranschaulichen, falls die einzelnen Amplituden ungefähr in die gleiche Richtung „zeigen“, ergibt sich eine Verstärkung.

Wenn wir nun in dem Integral $u = rb/q$ substituieren, erhalten wir

$$\frac{1}{r} \int_{0}^{\lfloor (q-k-1)/r \rfloor r/q} e^{2\pi i \frac{\{rc\}_q}{r} u} du. \tag{21}$$

Nun können wir, da $k < r$, mit einem Fehler von nur $O(1/q)$ die obere Grenze des Integrals durch 1 ersetzen und bekommen

$$\frac{1}{r} \int_{0}^{1} e^{2\pi i \frac{\{rc\}_q}{r} u} du. \tag{22}$$

$\{rc\}_q/r$ ist im Intervall $[-\frac{1}{2}, \frac{1}{2}]$, der Betrag des Integral wird also für $\{rc\}_q/r = \pm\frac{1}{2}$ minimal und beträgt $2/(\pi r)$. Der Betrag dieser Größe ist eine untere Schranke für die Wahrscheinlichkeit, einen bestimmten Zustand mit $\{rc\}_q \leq r/2$ zu beobachten, deswegen ist diese asymptotisch durch $4/(\pi^2 r^2)$ beschränkt und für genügend große n mindestens $1/3r^2$.

Die Wahrscheinlichkeit $|c, x^k \bmod n\rangle$ zu messen, beträgt also mindestens $1/3r^2$, falls

$$-\frac{r}{2} \leq rc \bmod q \leq \frac{r}{2}, \tag{23}$$

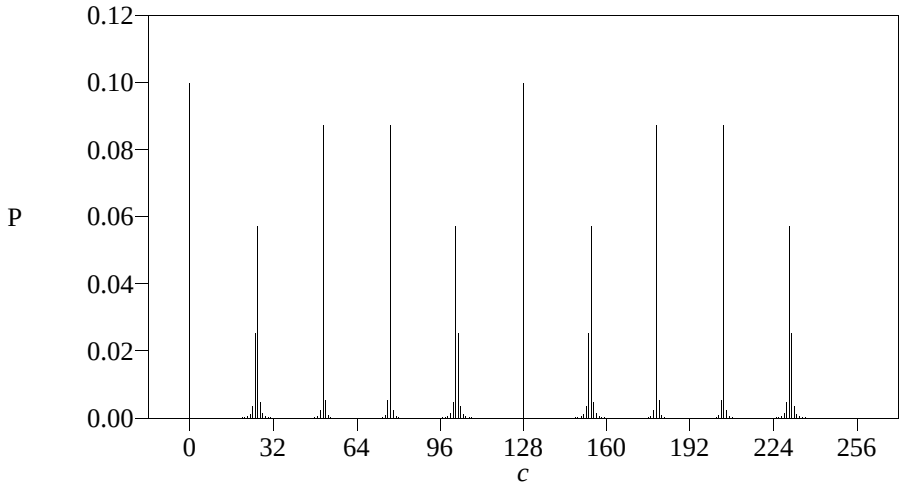

Abbildung 5: Wahrscheinlichkeit c zu beobachten, $q = 256$, $r = 10$ ([Sho97])

d. h. wenn gilt

$$\exists d : -\frac{r}{2} \leq rc - dq \leq \frac{r}{2}. \tag{24}$$

Daraus erhalten wir:

$$\left|\frac{c}{q} - \frac{d}{r}\right| \leq \frac{1}{2q}, \tag{25}$$

wobei wir c und q kennen.

Würde r nun q teilen, erhielten wir somit direkt unsere gesuchte Ordnung r. Dies ist allerdings im allgemeinen nicht der Fall, wir können aber über einen kleinen Umweg trotzdem r ermitteln, indem wir die sogenannte Kettenbruchentwicklung anwenden.

Da zu Beginn $q > n^2$ gewählt wurde, gibt es höchstens einen Bruch d/r mit $r < n$, der obige Bedingung erfüllt. Wir erhalten diesen Bruch vollständig gekürzt, indem wir c/q auf den nächsten Bruch runden, der einen Nenner kleiner als n hat. Dieser Bruch kann mittels Kettenbruchentwicklung von c/q effizient gefunden werden.

Ein (endlicher, einfacher) Kettenbruch ist ein Ausdruck der Form

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\cdots + \cfrac{1}{a_N}}}}} \tag{26}$$

mit $a_0, \ldots, a_N \in \mathbb{N} \cup \{0\}$. Jede positive rationale Zahl kann durch einen Kettenbruch dargestellt werden, der wie folgt berechnet wird: Sei $a_0 = \lfloor x \rfloor$ und $x = a_0 + \xi_0$ für ein $\xi_0 \in$

$[0,1)$. Wenn $\xi_0 \neq 0$ dann sei $a_1 = \lfloor 1/\xi_0 \rfloor$ und $1/\xi_0 = a_1 + \xi_1$ für ein $\xi_1 \in [0,1)$ usw. Bei rationalem x terminiert dieser Prozeß und wir erhalten die a_i der Kettenbruchentwicklung (vgl. [EJ94], [Knu97] Analyse des Euklidischen Algorithmus).

Ist d nun teilerfremd zu r, erhalten wir damit r (wir bekommen durch die Kettenbruchentwicklung einen vollständig gekürzten Bruch). Es gibt $\phi(r)$ mögliche Werte von d teilerfremd zu r ($\phi(m) := |\{k \in \{1,\ldots,m\} : \mathrm{ggT}(k,m) = 1\}|$ ist die Eulersche ϕ-Funktion, die die Anzahl der zu m teilerfremden Zahlen angibt). Jeder dieser Brüche d/r ist nahe an einem Bruch c/q mit $|c/q - d/r| \leq 1/2q$. Es gibt zudem r mögliche Werte für x^k, da r die Ordnung von x ist.

Da jeder dieser Zustände mit einer Wahrscheinlichkeit von mindestens $1/3r^2$ auftritt, erhalten wir r mit einer Wahrscheinlichkeit größer oder gleich $\phi(r)/3r$. Wenn wir nun ausnutzen, daß $\phi(r)/r > \delta/\log\log r$ für eine Konstante δ ist (siehe [HW79], zitiert in [Sho97] und [EJ94]), müssen wir das Experiment also nur $O(\log\log r)$ mal wiederholen, um r mit einer hohen Wahrscheinlichkeit zu erhalten.

Dies kann in der Praxis noch weiter verbessert werden, indem bei gemessenem $|c\rangle$ auch noch Zahlen nahe an c wie $c \pm 1, c \pm 2, \ldots$ ausprobiert werden, da auch diese noch eine einigermaßen hohe Wahrscheinlichkeit haben, nahe an einem Bruch qd/r zu sein.

Außerdem kann man die Rechnung nicht nur mit einem Kandidaten r' für r, sondern, da der Bruch c/q ja vollständig gekürzt zu d'/r' vorliegt, auch für $2r', 3r', \ldots$ durchführen, ob diese Werte vielleicht die Ordnung von x sind. Weiterhin ist es auch noch möglich, daß man, wenn man zwei Kandidaten für r gefunden hat, das kleinste gemeinsame Vielfache dieser beiden Werte ausprobiert, auch dadurch lassen sich die Versuche auf dem Quantencomputer reduzieren, so daß man zwar mehr klassische Nachbearbeitung braucht, aber der schwierig zu realisierende Quantencomputer nur wirklich da eingesetzt wird, wo er unbedingt benötigt wird.

5 Ausblick

Shors Algorithmus bot Anlaß für weitere Forschungen in diesem Gebiet, er machte Quantum Computing richtig bekannt und verhalf ihn auch zu einer gesteigerten Aufmerksamkeit außerhalb der Forschung. Obwohl die praktische Realisierung in nächster Zukunft allerdings sehr fraglich ist, machte man doch in den letzten Jahren erhebliche Fortschritte beim Bau von kleinen Quantencomputern, so daß es vielleicht gar nicht mehr allzu lange brauchen wird, bis dieser Algorithmus einmal implementiert wird.

Allerdings sind in der Praxis wohl doch zunächst eher noch Fortschritte auf klassischem Gebiet zu erwarten, oder man kann z. B. einfach dadurch schneller faktorisieren, daß man für einzelne Schritte der klassischen Algorithmen spezielle Hardware entwickelt (vgl. z. B. den Entwurf des TWINKLE-Gerätes von Adi Shamir).

Literatur

[Ber97] BERTHIAUME, ANDRÉ: *Quantum Computation*. In: HEMASPAANDRA, LANE A. und

ALAN L. SELMAN (Herausgeber): *Complexity Theory Retrospective II*, Seiten 23–51. Springer, New York, 1997.

[DW95] DRUMM, V. und W. WEIL: *Lineare Algebra und Analytische Geometrie*. 1995.

[EJ94] EKERT, ARTUR und RICHARD JOZSA: *Notes on Shor's Efficient Algorithm for Factoring on a Quantum Computer*. NIST Workshop on Quantum Computing and Communication, Gaithersburg, MD, 18. – 19. August, 1994.

[Heu98] HEUSER, HARRO: *Lehrbuch der Analysis – Teil 1*. Teubner; Stuttgart, Leipzig, 12. Auflage, 1998.

[HW79] HARDY, G. H. und E. M. WRIGHT: *An Introduction to the Theory of Numbers*. Oxford University Press, New York, 5. Auflage, 1979.

[Knu97] KNUTH, DONALD E.: *The Art of Computer Programming – Volume 2 Seminumerical Algorithms*. Addison-Wesley Longman, 3. Auflage, 1997.

[Sho94] SHOR, PETER W.: *Polynomial–Time Algorithms for Prime Factorization and Discrete Logarithms*. In: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, Seiten 124–134. IEEE Computer Society Press, November 1994. Siehe auch LANL preprint quant–ph/9508027.

[Sho97] SHOR, PETER W.: *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*. SIAM Journal on Computing, 26(5):1484–1509, 1997.

[Zal98] ZALKA, CHRISTOF: *Fast versions of Shor's quantum factoring algorithm*. LANL preprint quant–ph/9806084, 1998.