Shor-Algorithmus

Seminar Quanteninformationstheorie SS2008

Thomas Munker

Di. 24.Juni 2008

Inhaltsverzeichnis

1 Einführung

1.1 Problemstellung

Problemstellung

Zerlegung einer ganzen Zahl N in ihre Primfaktoren p_n

$$N = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n} \qquad \text{z.B.} \quad 15 = 3^1 \cdot 5^1$$

trivialer Lösungsansatz

- Probiere Division durch 1 bis $\sqrt{N}$
- Benötigt bis zu $\sqrt{N}$ Schritte.

  $$\sqrt{N} = 2^{\frac{1}{2}\log_2 N}$$

- benötigte Schritte sind exponentiell in der Anzahl Bits!

Lösung mit moderner Mathematik

- benötigt

  $$\exp\left((\log_2 N)^{\frac{1}{3}} (\log_2 \log_2 N)^{\frac{2}{3}}\right)$$

- ebenfalls exponentiell zur Anzahl der Bits (wenn auch schwächer)

Folgen / Nutzung

- Moderne Public Key Verfahren (asymmetrische Verschlüsselung wie z.B. RSA) verwenden dies als Falltürfunktion.

Public Key		Private Key
N	$\xleftarrow{\textit{leicht}}$	$p \cdot q (= N)$
	$\xrightarrow{\textit{schwer}}$	

Eine effiziente Primfaktorzerlegung großer Zahlen

gefährdet folglich die meisten (alle?) modernen kryptographischen Verfahren (der asymmetrischen Verschlüsselung)

Der Shor Algorithmus

Der Shor Algorithmus ist nicht deterministisch. D.h. er liefert bei gleicher Ausgangssituation nicht immer gleiche Ergebnisse. Die Ergebnisse sind demnach zwar nicht immer korrekt, dass stellt jedoch kein Problem dar, da sie leicht überprüfbar sind.

Der Shor Algorithmus faktorisiert mit etwas *weniger als* $(\log_2 N)^3$

Beispiel 1. Vergleich der benötigten Berechnungsdauern t unter Annahme eines Prozessor mit 4 GHz Rechengeschwindigkeit. n ist die Anzahl der Bits der zu faktorisierenden Zahl N

Algo.	t bei $n = 256$	t bei $n = 1024$	t bei $n = 4096$
klassisch	$\approx$ 26s	$\approx$ 1649Jahre	$\approx 2 \cdot 10^{19}$Jahre
Shor	$\approx$ 4ms	$\approx$ 0.3s	$\approx$ 17s

1.2 Nutzen eines Quantenrechners

Was können Quantencomputer?

- Sie nutzen Superposition und Verschränkung von Quantensystemen
- Quantencomputer sind daher nicht generell besser als herkömmliche Rechner.
- Ihre Überlegenheit beschränkt sich also auf zwei Aufgabengebiete:

Amplitudenverstärkung
Der Zustand des gewünschten Resultats wird mittels Superposition sehr wahrscheinlich (z.B. für Suchalgorithmen wie der Grover-Algorithmus)

Funktionseigenschaften
Das Auffinden von globalen Funktionseigenschaften (z.B. für Periodensuche, Shor-Algorithmus)

2 Klassischer Teil 1

Klassischer Teil 1
In diesem Teil führen wir die Faktorisierung zurück auf ein quantenphysikalisch lösbares Problem.

Wir gehen also davon aus dass wir die Periode einer Funktion

$$\mathrm{f}(x) = a^x \pmod N$$

Quantenmechanisch effizient finden können.

2.1 Primfaktoren aus der Periode berechnen

Gegeben
$N = pq$ wobei p und q die zu findenden Primfaktoren sind

Definition 2. $\mathrm{f}(x) = a^x \pmod N$ für eine Zufallszahl $2 \leq a \leq N - 1$

Für die Periode r von $\mathrm{f}(x)$ gilt $a^r \equiv 1 \pmod N$ Dies folgt aus der Periodizität:

$$\mathrm{f}(r + x) = \mathrm{f}(x) \quad \Rightarrow \quad \mathrm{f}(r + 0) = \mathrm{f}(0) \text{ und } \mathrm{f}(0) = a^0 (\mathrm{mod}\ N) = 1$$

Beispiel 3. wenn wir $N = 15$ gegeben haben und als Zufallszahl $a = 11$ wählen, dann bekommen wir für unsere Periode $r = 2$. D.h.

$$\mathrm{f}(x) = \mathrm{f}(x + r) = \mathrm{f}(0 + 2) = 11^2 \pmod{15} = 121 \pmod{15} = 1$$

Weitere Bedingungen

r muss gerade sein, denn dann können wir zerlegen:

$$0 \equiv \left(a^{r/2}\right)^2 - 1 = \underbrace{(a^{r/2} - 1)}_{x_-}\underbrace{(a^{r/2} + 1)}_{x_+} \pmod{N}$$

Außerdem benötigen wir:

$x_- \not\equiv 0 \pmod{N}$	gilt, da r kleinste Zahl für $a^r \equiv 1 \pmod{N}$
$x_+ \not\equiv 0 \pmod{N}$	wir brauchen Glück

Beispiel 4. Für unser Beispiel ($N = 15$, $a = 11$, $\Rightarrow\ r = 2$) bedeutet das:

$$x_- = 11^{\frac{2}{2}} - 1 \pmod{15} = 11 - 1 = 10$$
$$x_+ = 11 + 1 \pmod{15} = 12$$

Es sind also alle Bedingungen erfüllt. Wäre dies nicht der Fall müssten wir einfach ein anderes a wählen.

Satz

Aus $x_- \not\equiv 0 \pmod{N} \not\equiv x_+$ folgt das der größte gemeinsame Teiler von x_- und N die eine Primzahl $q = \mathrm{ggT}(x_-, N)$ und der größte gemeinsame Teiler von x_+ und N die andere Primzahl $p = \mathrm{ggT}(x_+, N)$ ergibt, aus welchen unsere Zahl $N = pq$ besteht

Beweis

- Weder x_+ noch x_- sind durch N teilbar (ist gegeben, s.o.)
  $\Rightarrow$ d.h. weder x_+ noch x_- enthalten alle Primfaktoren
- Das Produkt beider ist jedoch teilbar ($x_+ \cdot x_- \equiv 0 \pmod{N}$) (auch gegeben,s.o.)
  $\Rightarrow$ Das Produkt muss folglich alle Primfaktoren enthalten!
- Nun gilt ja $N = pq$ und mit $k \in \mathbb{N}$ gilt $x_+ \cdot x_- = k \cdot N = k \cdot p \cdot q$ Dieses k lässt sich nun auf die einzelnen Faktoren aufspalten: $\Rightarrow\ x_+ = k_+ \cdot p; \quad x_- = k_- \cdot q$
- Da keine der ganzen Zahlen k,k_+,k_- in N enthalten sind bleiben als gemeinsame Teiler ungleich 1 oder N selbst eben nur noch die Primfaktoren! □

Beispiel 5. Wieder in unserem Beispiel ergibt das:

$$\mathrm{ggT}(x_-, N) = \mathrm{ggT}(10, 15) = 5$$
$$\mathrm{ggT}(x_+, N) = \mathrm{ggT}(12, 15) = 3$$
$$N = pq = 5 \cdot 3 = 15 \quad \square$$

2.2 Ablauf des klassischen Teil 1

Wenn man das finden der Periode zu einem Schritt zusammenfasst findet die Faktorisierung also in 5 Einzelschritten statt, welche bei bedarf wiederholt werden:

1. Wähle eine Zufallszahl $2 \leq a \leq N-1$
2. Prüfe ob $\mathrm{ggT}(a, N) = \{1, N\}$ (Wenn nicht haben wir bereits eine Primzahl gefunden, was allerdings sehr unwahrscheinlich ist)
3. finde Periode r der Funktion $\mathrm{f}(x) = a^x \pmod N$
4. überprüfe ob
   - r gerade ist
   - $a^r \not\equiv -1 \pmod N$

   Falls nicht wählen wir einfach ein anderes a und fangen von vorne an. Die Wahrscheinlichkeiten liegen allerdings in etwa bei 50% [1]
5. berechne Primzahlen mit $p = \mathrm{ggT}(a^{r/2} - 1, N)$ und $q = \mathrm{ggT}(a^{r/2} + 1)$

3 Quantenphysikalischer Teil des Algorithmus

Quantenphysikalischer Teil

In diesem Teil wollen wir von einer Funktion $\mathrm{f}(x) = a^x \pmod N$ mit quantenphysikalischen Mitteln effizient die Periode r finden. Dies geschieht in 3 Schritten:

1. initialisiere Register
2. Rechne durch unitäre Transformation
3. Fourier-Transformation

3.1 Vorbereitungen

Benötigte Register

Wir benötigen 2 Register:

- Ein "Eingangs-" Register zum Ablegen von x
  - Die Qubitanzahl n bestimmt die Größe d des Definitionsbereich der Funktion durch $d = 2^n > N^2$ (d = Anzahl der Elemente)
    Diese wird so groß ($> N^2$) gewählt um später bestimmte Näherungen machen zu können.
  - Der Registerzustand wird beschrieben durch $|\phi\rangle = \sum_{x=0}^{d-1} \gamma(x)|x\rangle$

[1] Nachzulesen im [Mermin, Anhang M]

- Ein "Ausgangs-" Register in dem der Funktionswert f(x) abgelegt wird
  - Die Qubitanzahl m muss groß genug sein um alle N möglichen Funktionswerte aufzunehmen: $m \geq \log_2 N$
  - Zustände dort bezeichnen wir mit $|\chi\rangle$

Initialisierung der Register

Die Register müssen vor den Berechnungen natürlich einen definierten Zustand haben:

- Das "Eingangs"register soll eine gleichgewichtige Superposition der Basiszustände haben. Diese ist z.B. durch Hadamard-Gattern präparierbar: $\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}(|1\rangle + |0\rangle)$
- Das "Ausgangs"register soll den Null-Zustand haben.
- Der Zustand des Gesamtsystems ist folglich:

$$|\psi_1\rangle = (\mathbf{H}^{\otimes \mathbf{n}}|0\rangle_n)|0\rangle_m = \frac{1}{\sqrt{d}} \sum_{x=0}^{d-1} |x\rangle_n |0\rangle_m$$

Beispiel 6. Da unser Beispiel ($N = 15$, $a = 11$) ein Spezialfall ist reichen im Eingangsregister 3 Qubits aus ($\rightarrow d = 8$). Der Übersichtlichkeit wegen betrachten wir auch nicht mehr:

$$\rightarrow \qquad |\psi_1\rangle = \frac{1}{\sqrt{8}} \sum_{x=0}^{7} |x\rangle|0\rangle$$

3.2 Berechnung von f(x)

Aus der binären Darstellung von x

$$x = 2^{n-1}x_{n-1} + 2^{n-2}x_{n-2} + \cdots + 2^2 x_2 + 2x_1 + x_0; \qquad x_i \in 0, 1$$

folgt für f(x):

$$\begin{aligned} \mathrm{f}(x) = a^x &= a^{2^{n-1}x_{n-1}} \cdot a^{2^{n-2}x_{n-2}} \cdot \ldots \cdot a^{2^1 x_1} \cdot a^{x_0} \\ &= \prod_{j=0}^{n-1} a^{2^j x_j} \end{aligned}$$

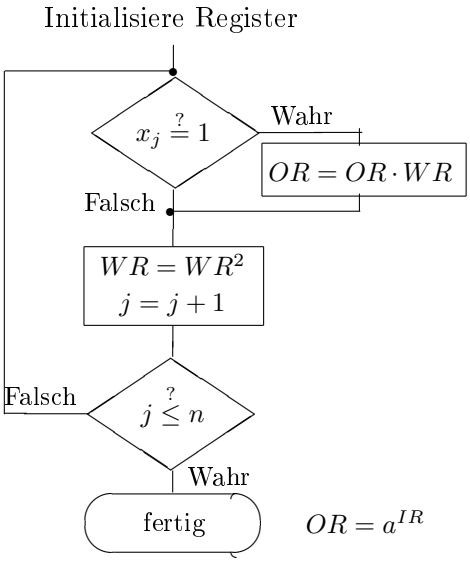
Wenn wir nun ausnutzen, dass

$$a^{2^n} = \left(a^{2^{n-1}}\right)^2$$

und dass der Modulo in eine Multiplikation (und beim Potenzieren) rein gezogen werden kann

$$a^x \pmod N = (a \pmod N)^x$$

lässt sich leicht folgender Algorithmus aufbauen:



Verwendeten Register und Initialisationswerte:

- Ausgangsregister $OR = 1$
- Eingangsregister $IR = x; x_j \in \{0, 1\}$
- Arbeitsregister $WR = a$

Beispiel 7. Für unser Beispiel ($N = 15$, $a = 11$) läuft das dann für z.B. $x = 2 = 0010$ in folgender Weise ab:

- Initialisiere mit $OR = 1$, $IR = x = 2 = 0010$, $WR = 11$
- $x_0 \stackrel{?}{=} 1$ Nein: alles bleibt
- $WR = WR^2 = 11^2 \pmod N = 121 \pmod N = 1$
- $j = j + 1 = 1$ (betrachte nächstes Bit)
- Sind wir schon über das letztes Bit hinaus? Nein, mache weiter!

- $x_1 \stackrel{?}{=} 1$, Ja: $OR = OR \cdot WR = 1 \cdot 1 = 1$
- $WR = WR^2 = 1^2 = 1$
- $j = j + 1 = 2$ (nächstes Bit)
- Sind wir schon über das letzte Bit hinaus? Nein, mache weiter!

- $x_2 \stackrel{?}{=} 1$, Nein: tue nichts
  $\vdots$

Bis alle Bits durch sind.

klassisch
Für jedes Bit x_j ist eine eigene Speicherzelle mit $\log_2 N$ Bit nötig. Speicherbedarf für $n \geq \log_2 N^2$ und $m = \log_2 N$: $n \cdot m \geq 2 \cdot m^2$ Bit

quantenmechanisch
Alle Ergebnisse superpositionieren sich im Ausgangsregister, nur $m = \log_2 N$ Qubits nötig

- Endzustand: Gleichgewichtige Superposition aller x-Werte mit entsprechenden Funktionswerten
$$|\psi_2\rangle = \frac{1}{\sqrt{d}} \sum_{x=0}^{d-1} |x\rangle |a^x (\mathrm{mod}\, N)\rangle$$
- Messung beider Register: Zufälliger Funktionswert und zugehörige x-Wert; alle anderen Information verloren

Beispiel 8. Wieder in unserem Beispiel ergibt die Berechnung der Funktion folgenden Gesamtzustand

$$|\psi_2\rangle = \frac{1}{\sqrt{8}}\{|0\rangle|1\rangle + |1\rangle|11\rangle + |2\rangle|1\rangle + |3\rangle|11\rangle + \cdots\} = \frac{1}{\sqrt{8}}\{(|0\rangle + |2\rangle + |4\rangle + |6\rangle)|1\rangle + (|1\rangle + |3\rangle + |5\rangle + |7\rangle)|11\rangle\}$$

Zustand im Eingangsregister
Misst man *nur* Funktionswert $|y_2\rangle$ *im Ausgangsregister*, so folgt für das Eingangsregister:
$$|\phi_2\rangle = \frac{1}{\sqrt{q}} \sum_{j=0}^{q-1} |l + jr\rangle$$

- r: Periode der Funktion f(x) $\quad \rightarrow$ leider nicht direkt messbar
- l: Kleinster Wert aus ($0 \leq l \leq r$) so dass gilt f(l) = y_2
- q: Kleinste Ganzzahl für die $(qr + l) \geq 2^n$ gilt

Beispiel 9. Wieder mit ($N = 15$, $a = 11$) geschieht durch die Messung von χ_2 im Ausgangsregister folgendes mit dem Eingangsregister ϕ_2:

$$|\chi_2\rangle = |1\rangle \quad \rightarrow \quad |\phi_2\rangle = \frac{1}{\sqrt{4}}(|0\rangle + |2\rangle + |4\rangle + |6\rangle)$$

$l = 0; \qquad q = 4; \qquad r = 2$

$$|\chi_2\rangle = |11\rangle \quad \rightarrow \quad |\phi_2\rangle = \frac{1}{\sqrt{4}}(|1\rangle + |3\rangle + |5\rangle + |7\rangle)$$

$l = 1; \qquad q = 4; \qquad r = 2$

3.3 diskrete Fouriertransformation

Quantenmechanische Fouriertransformation

Definition 10.

$$\mathbf{U_{FT}}|x\rangle = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} e^{2\pi i x y/2^n}|y\rangle$$

Wobei n wieder die Anzahl der Qubits darstellt, unser betrachteter Raum also 2^n Werte beinhaltet

- im Folgenden für 4 Bits aus einfachen Gattern aufgebaut

Darstellung durch Gatteroperationen

Wenn man sich die Definition des Hadamard- und erweiterten **Z** Operators $\mathcal{Z}|y\rangle = e^{2\pi i y/2^n}|y\rangle$ anschaut, lässt sich $\mathbf{U_{FT}}$ Umformen auf

$$\mathbf{U_{FT}}|x\rangle = \mathcal{Z}^x \mathbf{H}^{\otimes \mathbf{n}}|0\rangle$$

Wenn man nun den Zahlenoperator $\mathbf{n}$ einführt, welcher den Zahlenwert (0 oder 1) des jeweiligen Qubits zurück gibt, so befolgt der Hadamardoperator folgenden Zusammenhang:

$$e^{i\pi x \mathbf{n}}\mathbf{H}|0\rangle = \mathbf{H}|x\rangle; \quad x \in \{0,1\}$$

Nun lässt sich auch der $\mathcal{Z}^{\mathbf{x}}$ Operator mit Hilfe des Zahlenoperators zerlegen. Anwenden dieser Zerlegungen auf unsere Fouriertransformation ergibt

$$\mathbf{U_{FT}}|x_3\rangle|x_2\rangle|x_1\rangle|x_0\rangle$$
$$\exp\left[i\pi\left(\frac{1}{2}x_0\mathbf{n}_2 + (\frac{1}{2}x_1 + \frac{1}{4}x_0)\mathbf{n}_1 + (\frac{1}{2}x_2 + \frac{1}{4}x_1 + \frac{1}{8}x_0)\mathbf{n}_0\right)\right]\mathbf{H}_3\mathbf{H}_2\mathbf{H}_1\mathbf{H}_0|x_0\rangle|x_1\rangle|x_2\rangle|x_3\rangle$$

Die Indizes deuten jeweils das Qubit an, auf welches der Operator wirkt. Es fällt auf dass die Qubits ihre Wertigkeit für die Darstellung von x vertauscht haben.

Daher führt man einen Vertauschungsoperator $\mathbf{P}$ ein, welcher die Reihenfolge der Qubits vertauscht. Wenn man nun nach ein wenig Umstellung den Operator $\mathbf{V}_{ij} = \exp\left(i\pi \mathbf{n}_i \mathbf{n}_j / 2^{|i-j|}\right)$ einführt folgt daraus:

$$\begin{aligned}&\mathbf{U_{FT}}|x_3\rangle|x_2\rangle|x_1\rangle|x_0\rangle \\ &\quad = \mathbf{H_3}(\mathbf{V_{32}}\mathbf{H_2})(\mathbf{V_{31}}\mathbf{V_{21}}\mathbf{H_1})(\mathbf{V_{30}}\mathbf{V_{20}}\mathbf{V_{10}}\mathbf{H_0})\mathbf{P}|x_3\rangle|x_2\rangle|x_1\rangle|x_0\rangle\end{aligned}$$

Zur Erinnerung:

$\mathbf{H_j}$ das Hadamard-Gatter auf das j-te Qubit

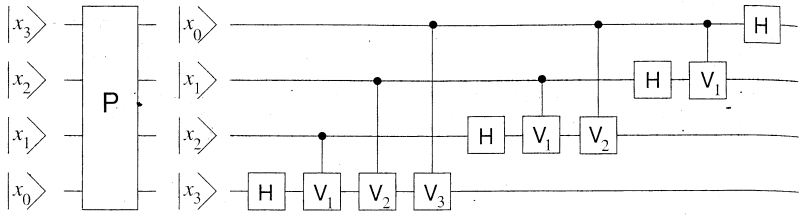
$\mathbf{V_{ij}}$ ein Phasengatter $\mathbf{V_{ij}} = \exp(i\pi \mathbf{n_i} \mathbf{n_j} / 2^{|i-j|})$ welches je nach Abstand der Qubits unterschiedlich stark auf diese wirkt

$\mathbf{n_j}$ gibt den Zustand des j-ten Qubits wieder

$\mathbf{P}$ vertauscht die Zustände der Qubits untereinander

Gatterschaltplan der Fouriertransformation

Diese Schaltung entspricht jetzt folgendem Schaltplan der verschiedenen Gatter $\mathbf{V}_{ij}; \mathbf{H}_n$. Die Vertauschung $\mathbf{P}$ erfolgt praktisch natürlich einfach durch ein umgekehrtes auslesen des Ergebnisses.



Dies wird besonders deutlich wenn man die Schaltung, von links nach rechts gelesen, mit folgender Gleichung, von rechts nach links gelesen, vergleicht und beachtet das die jeweiligen Operatoren ja nur auf die Qubits wirken, welche in dessen Index stehen:

$$\mathbf{U_{FT}}|x\rangle = \mathbf{H_3}(\mathbf{V_{32}H_2})(\mathbf{V_{31}V_{21}H_1})(\mathbf{V_{30}V_{20}V_{10}H_0})\mathbf{P}|x_3\rangle|x_2\rangle|x_1\rangle|x_0\rangle$$

Quanten Fouriertransformation

Diese Fouriertransformation

$$\mathbf{U_{FT}}\left(\sum_{x=0}^{2^n-1}\gamma(x)|x\rangle\right) = \sum_{x=0}^{2^n-1}\tilde{\gamma}(x)|x\rangle$$

mit

$$\tilde{\gamma}(x) = \frac{1}{2^{n/2}}\sum_{y=0}^{2^n-1} e^{2\pi i x y/2^n}\gamma(y)$$

wandelt uns nun also einen Zustand $|\phi_2\rangle$ (Eingangsreg.) um in folgenden Zustand $|\phi_3\rangle$:

$$|\phi_3\rangle = \mathbf{U_{FT}}|\phi_2\rangle = \frac{1}{\sqrt{2^n q}}\sum_{y=0}^{2^n-1} e^{2\pi i l y/2^n}\left(\sum_{j=0}^{q-1} e^{2\pi i r j y/2^n}\right)|y\rangle$$

Messung nach Fouriertransformation

Für die Wahrscheinlichkeit einen bestimmten Zustand $|\phi_3\rangle = |y\rangle$ zu messen bedeutet das:

$$p(y) = |\langle y|\phi_3\rangle|^2 = \frac{1}{2^n q}\left|\sum_{j=0}^{q-1} e^{2\pi i r j y/2^n}\right|^2$$

Hier fällt auf das diese Wahrscheinlichkeit zu dem bestimmten Messwert y nur noch von unserem gesuchten Wert r abhängt.

Wir betrachten nun Werte $y = y_k = k\frac{2^n}{r} + \delta_k$ mit $|\delta_k| \leq 1/2$ und $k < r$ Die Wahrscheinlichkeit einen der Werte y_k zu messen ist nun (nach Rechnung & Näherung):

$$\sum_{k=0}^{r-1} p(y_k) = p(y_0)r \geq (4/\pi^2) = 0.4053$$

4 Klassischer Teil 2

4.1 Quantenmechanisches Messergebnis

Quantenmechanisches Ergebnis
Hohe Wahrscheinlichkeit $y = 2^n k/r + \delta_k$ zu messen, wobei r die Periode der Funktion $\mathrm{f}(x) = a^x (\mathrm{mod} N)$

Variablen
n: Anzahl Qubits
N: zu zerlegende Zahl
y: Messergebnis
k: zufällige Ganzzahl
r: Periode der Funktion
$\delta_k \leq \frac{1}{2}$

4.2 Auswertung des Messergebnisses

Auswertung des Messergebnisses

- Nur $\frac{k}{r} = \frac{k_0}{r_0}$ bestimmbar, nicht die einzelnen k und r

r_0 und k_0
sind die Faktoren von r und k die diese beiden nicht gemeinsam haben. D.h.:

$$\frac{k}{r} = \frac{f_k \cdot k_0}{f_r \cdot r_0} \quad \text{mit} \quad f_k \stackrel{!}{=} f_r = f$$

Bedingung
$2^n > N^2$ damit $\frac{y}{2^n} = \frac{k}{r} + \frac{\delta_k}{2^n} \approx \frac{k}{r}$

- Dann ist r_0 (und k_0) durch eine Partialbruchzerlegung bestimmbar
- Wenn k_0 kein Teiler von r ist ($\approx 60\%$ wahrscheinlich), dann ist $r_0 = r$
- sonnst probieren wir einfach niedrige vielfache von r_0 aus, da $r = r_0 \cdot$ gemeinsame_Faktoren(k,r) und die gemeinsamen Faktoren wahrscheinlich niedrig sind.
- falls auch dies erfolglos ist, wiederholt man die Periodenbestimmung einfach

4.3 Gesamtablauf der Periodenbestimmung

Die quantenmechanische Periodenbestimmung läuft dann nach folgendem Schema ab:

1. initialisiere die Register
2. berechne das Ausgangsregister
3. führe die Fouriertransformation des Eingangsregister durch
4. messe ein y im Eingangsregister
5. bestimme r_0 durch eine Partialbruchzerlegung
6. probiere niedrige Vielfache $(1, 2, 3, \ldots)$ von r_0 als Periode. Falls dies erfolglos ist beginnt man von vorne (bei 1.) und findet so ein anderes r_0

Das Wichtigste des Shor-Algorithmus

- Der Shor-Algorithmus führt die Primfaktorzerlegung auf eine Periodensuche zurück
- Die Periodensuche ist dank quantenmechanischer Fouriertransformation auch für große Zahlen effizient lösbar

Literatur

[Audretsch] Jürgen Audretsch, *Verschränkte Systeme*, WILEY–VCH, 2005,lbs 780 /a92b:i

[Mermin] N. D. Mermin, *Quantum Computer Science*, Cambridge University Press, 2007

[Bowmeester] D. Bouwmeester, A. Ekert, A. Zeilinger, *The Physics of Quantum Information*, Springer-Verlag, 2000,phy 203k /b69(1a)

[wiki] http://de.wikipedia.org/wiki/Shor-Algorithmus