Quantencomputer

Der Shor-Algorithmus

Präsentation von Oleg Yuschuk

Der Shor Algorithmus

- Peter W. Shor (* 14. August 1959 in New York)
- Algorithmus zum Faktorisieren von Zahlen auf dem Quantencomputer
- Besonderheit: Der Algorithmus arbeitet in polynomialer Zeit abhängig der Qubits.

Komplexitätsklassen

- Effektivster Algorithmus zum Faktorisieren auf "normalen" PCs läuft in exponentieller Zeit. Bisher kein polynomialer Algorithmus zum Faktorisieren bekannt. Aber die nichtexistenz eines polynomialen Algorithmus ist auch nicht bewiesen.
- Anwendung: Verschlüsselung, RSA

RSA

- Asymmetrisches Verfahren zur Verschlüsselung.
- Kernstück sind zwei gigantische Primzahlen q und p die multipliziert werden q*p=N
- Aus q und p werden zwei Schluessel e und d generiert
- N und e sind zum Verschlüsseln nötig und öffentlich. N und d zum Entschlüsseln, d ist privat.

Durchführung

- Zu faktorisieren: N
- Suchen x mit:

$$x^2 = 1 \; mod \; N$$
$$\Rightarrow (x-1)(x+1) = 0 \; mod \; N$$

- ggT(x-1,N) und ggt(x+1,N) sind Faktoren

Durchführung

- Nimm zufälliges y mit ggT(y,N)=1
- Such r mit $y^r = 1 \, mod \, N$
- Wenn r gerade, Problem gelöst

Überblick zum Algorithmus

$y=7 \quad N=15$

x	1	2	3	4	5	6	7	8	9	10
$y^x \bmod N$	7	4	13	1	7	4	13	1	7	4
Messung		4				4				4
Verschränkt		2				6				10

Durchführung

- Wir nutzen 2 Register. Das erste mit n Qubits überführen wir in eine Überlagerung aller Zustände:

$$|\Psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle |0\rangle$$

- Das überführen wir dann zu:

$$|\Psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \left| x^k \, mod \; N \right\rangle$$

Durchführung

- Wir führen eine Messung am zweiten Register durch.

$$|\Psi\rangle = \frac{1}{\sqrt{2^n / r}} \sum_{k=0}^{2^n/r-1} |x_0 + kr\rangle |a\rangle$$

- Am ersten Register führen wir die Fouriertransformation durch:

$$|\Psi\rangle = \sum_{k=0}^{r-1} \alpha_k |k\, 2^n / r\rangle |a\rangle$$

Durchführung

- Jetzt messen wir das erste Register von

$$|\Psi\rangle = \sum_{k=0}^{r-1} \alpha_k \left| k\, 2^n / r \right\rangle |a\rangle$$

- Die Messung dividieren wir durch 2^n und erhalten k/r. Wenn ggT(k,r)=1 haben wir unser Ergebnis.
- Wenn $k2^n / r$ keine ganze Zahl?

Durchführung

- Wir nutzen einen Kettenbruch

$$\phi = a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + f_n}} \quad mit \quad f_n < 1$$

- Wir setzen $f_n = 0$ und schauen ob der entstehende Bruch r rausgibt.

Quanten Fouriertransformation

- Ähnelt der normalen Fouriertransformation

$$\sum_{j=0}^{N-1} \alpha_j |j\rangle \rightarrow \sum_{k=0}^{N-1} \tilde{\alpha}_k |k\rangle = \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i jk/N} \alpha_j |k\rangle$$

- Für den Fall $\alpha_j = 1/\sqrt{N}$

$$k = N, k = 0 : \sum_{j=0}^{N-1} e^{2\pi i jk/N} = N$$

$$k \neq N : \sum_{j=0}^{N-1} e^{2\pi i jk/N} = \frac{1 - e^{2\pi ik}}{1 - e^{2\pi ik/N}} = 0$$

Quanten Fouriertransformation

- Quantenschaltkreis $N = 2^n$

$$|j\rangle = |j_1\rangle |j_2\rangle \ldots |j_n\rangle$$

$$Bsp: |6\rangle = |1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0\rangle = |1\rangle |1\rangle |0\rangle$$

$$F|j\rangle = \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i 0.j_n}|1\rangle) \otimes (|0\rangle + e^{2\pi i 0.j_{n-1} j_n}|1\rangle) \otimes \ldots$$

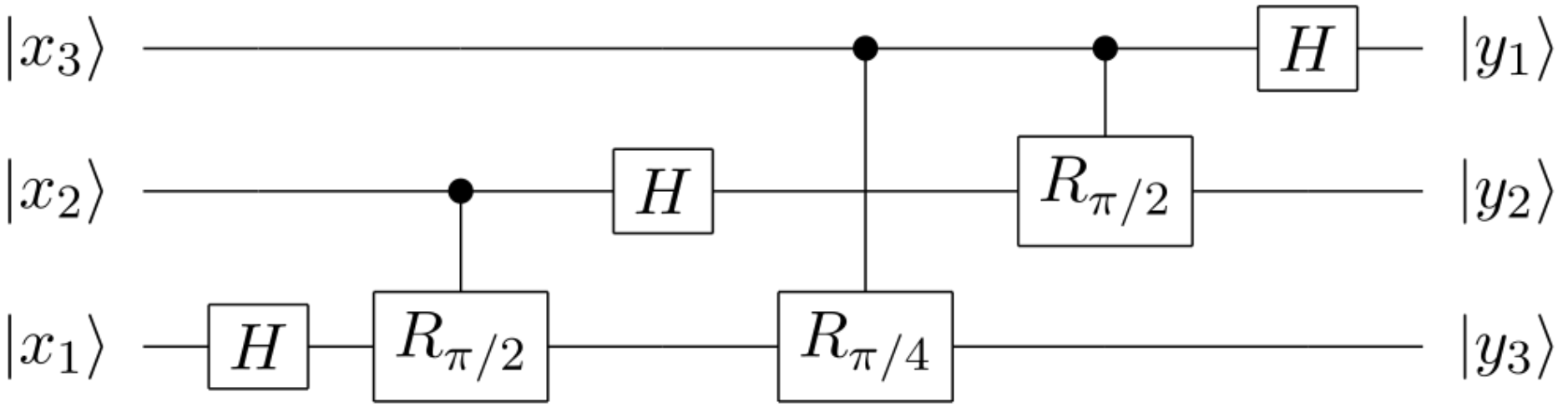
$$\ldots \otimes (|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_{n-1} j_n}|1\rangle)$$

- Diese Transformation lässt sich mit Hadamard-Gattern und Phasenrotations-Gattern erreichen.

Quanten Fouriertransformation

$$F\left|j\right\rangle = \frac{1}{\sqrt{2^n}}\left(\left|0\right\rangle + e^{2\pi i 0.j_n}\left|1\right\rangle\right) \otimes \left(\left|0\right\rangle + e^{2\pi i 0.j_{n-1}j_n}\left|1\right\rangle\right) \otimes \ldots$$

$$\ldots \otimes \left(\left|0\right\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_{n-1} j_n}\left|1\right\rangle\right)$$



Quelle: http://en.wikipedia.org/wiki/Quantum_Fourier_transform

Experimentelle Realisierung

- IBM hat 2001 die Zahl 15 faktorisiert
- 7 Qubits wurden verwendet
- Für die Qubits wurde ein einziges Molekül verwendet und die Spins wurden mithilfe von NMR gemessen.

Quantencomputer

Der Shor-Algorithmus

Präsentation von Oleg Yuschuk

Der Shor Algorithmus

- Peter W. Shor (* 14. August 1959 in New York)
- Algorithmus zum Faktorisieren von Zahlen auf dem Quantencomputer
- Besonderheit: Der Algorithmus arbeitet in polynomialer Zeit abhängig der Qubits.

Shor Professor der angewandten Mathematik am MIT. Er ist Informatiker und Mathematiker.

Neben shoralgorithmus gibt es noch Groveralgorithmus zum durchsuchen von Datenbanken. Das sind die wichtigsten Algorithmen.

Komplexitätsklassen

- Effektivster Algorithmus zum Faktorisieren auf "normalen" PCs läuft in exponentieller Zeit. Bisher kein polynomialer Algorithmus zum Faktorisieren bekannt. Aber die nichtexistenz eines polynomialen Algorithmus ist auch nicht bewiesen.
- Anwendung: Verschlüsselung, RSA

RSA

- Asymmetrisches Verfahren zur Verschlüsselung.
- Kernstück sind zwei gigantische Primzahlen q und p die multipliziert werden q*p=N
- Aus q und p werden zwei Schluessel e und d generiert
- N und e sind zum Verschlüsseln nötig und öffentlich. N und d zum Entschlüsseln, d ist privat.

Problematik:

Die Erstellung eines Schluessels laeuft in polynomialer Zeit. Das knacken in expontieller. Wenn wir also dieselben PCs zum knacken und generieren nutzen, dann koennen wir das knacken verdammt lang werden lassen.

Durchführung

- Zu faktorisieren: N
- Suchen x mit:

$$x^2 = 1 \; mod \; N$$
$$\Rightarrow (x-1)(x+1) = 0 \; mod \; N$$

- ggT(x-1,N) und ggt(x+1,N) sind Faktoren

x!=1 x!=N-1 triviale Faktoren N und 1
ggt(x-1,n) und ggt(x+1,n) sind Faktoren von N
Ggt leicht zu berechnen.
ggt(a,b)=ggt(b,a mod b) wenn a>b, iterativ fortfuehren.
Iterationen terminieren, Laufzeit log_2(a)

Durchführung

- Nimm zufälliges y mit ggT(y,N)=1
- Such r mit $y^r = 1 \; mod \; N$
- Wenn r gerade, Problem gelöst

Lineare Algebra sagt aus: So ein r existiert immer!
R gerade also y^r=y^(2*r/2)=(y^(r/2))^2=x^2=1

Wie garantieren wir, dass wir ein gerades r finden?
Gar nicht. Aber die Wahrscheinlichkeit ein solches r zu finden ist hoch genug, dass es unsere Asymptotische Laufzeit nicht beeinflusst.

Überblick zum Algorithmus

$y = 7 \quad N = 15$

x	1	2	3	4	5	6	7	8	9	10
$y^x \bmod N$	7	4	13	1	7	4	13	1	7	4
Messung		4				4				4
Verschränkt		2				6				10

Durchführung

- Wir nutzen 2 Register. Das erste mit n Qubits überführen wir in eine Überlagerung aller Zustände:

$$|\Psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle |0\rangle$$

- Das überführen wir dann zu:

$$|\Psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \left| x^k \bmod N \right\rangle$$

Erste Überführung passiert mit n Hadamard Gates. Trivial.

Zweite nutzt den Phase estimation Algorithm. Kompliyierter.

Die anzahl der Qubits im ersten Register erhoeht die Wahrhscheinlichkeit ein aussagekraeftiges Ergebnis zu bekommen. Die im zweiten muss log_2 N sein.

Durchführung

- Wir führen eine Messung am zweiten Register durch.

$$|\Psi\rangle = \frac{1}{\sqrt{2^n / r}} \sum_{k=0}^{2^n/r-1} |x_0 + kr\rangle |a\rangle$$

- Am ersten Register führen wir die Fouriertransformation durch:

$$|\Psi\rangle = \sum_{k=0}^{r-1} \alpha_k |k\, 2^n / r\rangle |a\rangle$$

r ist die Periodenlaenge von f(k)=x^k mod N
Es gilt f(x_0+kr)=a. Wir koennen die nicht direkt ausm ersten rausholen, da wir ja nur eine Messung durchfuehren koennen.
FourierTransformation siehe Abschnitt Fouriertransformation

Durchführung

- Jetzt messen wir das erste Register von

$$|\Psi\rangle = \sum_{k=0}^{r-1} \alpha_k |k\, 2^n / r\rangle |a\rangle$$

- Die Messung dividieren wir durch 2^n und erhalten k/r. Wenn ggT(k,r)=1 haben wir unser Ergebnis.
- Wenn $k2^n / r$ keine ganze Zahl?

Wenn r ungerade ist wiederholen wir alles nochmal. Ausserdem muessen wir testen ob r/2 alle Bdeingungen erfuellt, da die Fouriertransformation auch vernachlaessigte Zusatzterme (klein) erstellt.
Falls ggT(k,r)!=1 dann nochmal durchfuehren und kgV bilden bis es passt.

Durchführung

- Wir nutzen einen Kettenbruch

$$\phi = a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + f_n}} \quad mit \quad f_n < 1$$

- Wir setzen $f_n = 0$ und schauen ob der entstehende Bruch r rausgibt.

Quanten Fouriertransformation

- Ähnelt der normalen Fouriertransformation

$$\sum_{j=0}^{N-1} \alpha_j |j\rangle \rightarrow \sum_{k=0}^{N-1} \tilde{\alpha}_k |k\rangle = \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i jk/N} \alpha_j |k\rangle$$

- Für den Fall $\alpha_j = 1/\sqrt{N}$

$$k = N, k = 0: \sum_{j=0}^{N-1} e^{2\pi i jk/N} = N$$

$$k \neq N: \sum_{j=0}^{N-1} e^{2\pi i jk/N} = \frac{1 - e^{2\pi ik}}{1 - e^{2\pi ik/N}} = 0$$

Normale FT transformiert die Funktionswerte. Hier nehmen wir die Amplituden der Quantenzustände und interpretieren sie als Funktioneswerte.

Quanten Fouriertransformation

- Quantenschaltkreis $N = 2^n$

$$|j\rangle = |j_1\rangle |j_2\rangle \ldots |j_n\rangle$$

$$Bsp: |6\rangle = |1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0\rangle = |1\rangle |1\rangle |0\rangle$$

$$F|j\rangle = \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i 0.j_n}|1\rangle) \otimes (|0\rangle + e^{2\pi i 0.j_{n-1} j_n}|1\rangle) \otimes \ldots$$

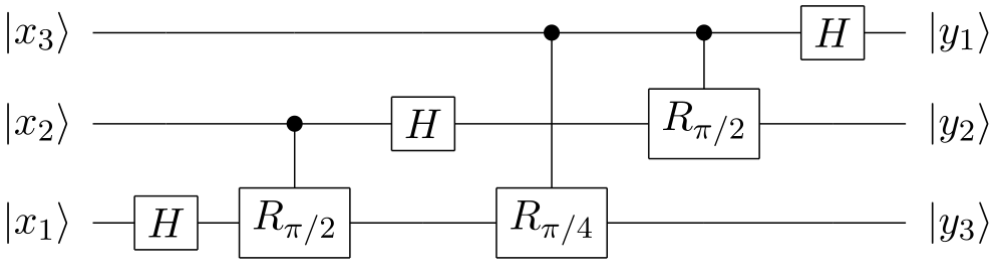
$$\ldots \otimes (|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_{n-1} j_n}|1\rangle)$$

- Diese Transformation lässt sich mit Hadamard-Gattern und Phasenrotations-Gattern erreichen.

Quanten Fouriertransformation

$$F\left|j\right\rangle = \frac{1}{\sqrt{2^n}}\left(\left|0\right\rangle + e^{2\pi i 0.j_n}\left|1\right\rangle\right)\otimes\left(\left|0\right\rangle + e^{2\pi i 0.j_{n-1}j_n}\left|1\right\rangle\right)\otimes\ldots$$

$$\ldots\otimes\left(\left|0\right\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_{n-1}j_n}\left|1\right\rangle\right)$$



Quelle: http://en.wikipedia.org/wiki/Quantum_Fourier_transform

Experimentelle Realisierung

- IBM hat 2001 die Zahl 15 faktorisiert
- 7 Qubits wurden verwendet
- Für die Qubits wurde ein einziges Molekül verwendet und die Spins wurden mithilfe von NMR gemessen.