Faktorisierung auf dem Quantencomputer

Von Joël Huber, Kantonsschule Freudenberg

In dieser Arbeit habe ich versucht zu verstehen, wie Quantencomputer funktionieren und wie man mit ihnen Zahlen faktorisieren kann, und habe alles, was ich gelernt habe, in Code umgesetzt, den man auf einem Quantencomputer ausführen könnte. Dieser Code ist online verfügbar unter:

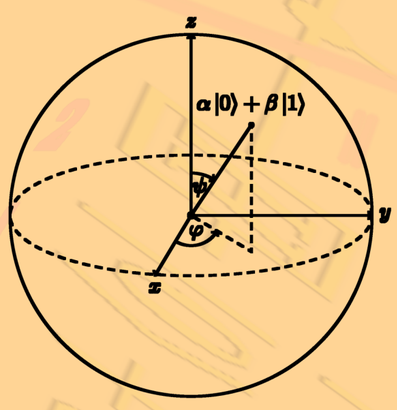
https://github.com/johutha/QInteger-QAlgorithms

Qubits

Ein normales Bit hat zwei Zustände - 0 und 1 - und befindet sich immer in genau einem dieser beiden Zustände. Kann man zwei solche Zustände in einem quantenmechanischen System gezielt herbeiführen, treten auch die Gesetze der Quantenphysik in Kraft. Das System kann sich dann auch in einer Superposition der beiden Zustände befinden, also 0 und 1 gleichzeitig sein. Dies erlaubt es uns, Überlagerungen von vielen verschiedenen Zuständen auf Quantencomputern herbeizuführen und mit diesen parallel zu rechnen.

Bit 0 oder 1

Qubit



Das Faktorisierungsproblem

$$10157579498$$

$$= 2 \times 11^2 \times 29^3 \times 1721$$

Das Faktorisierungsproblem besteht darin, eine Zahl in ihre Primfaktoren zu zerlegen. Auf klassischen Computern sind keine Verfahren bekannt, die dieses Problem effizient lösen. Auf den Begriff der Effizienz wird in meiner Arbeit weiter eingegangen. Die Laufzeiten der meisten klassischen Algorithmen wachsen exponentiell mit der Anzahl Bits der zu faktorisierenden Zahl.

Shors Algorithmus

Im Gegensatz zu klassischen Computern können Quantencomputer das Faktorisierungsproblem effizient lösen. Der Algorithmus dazu wurde von Peter Shor im Jahr 1994 entdeckt. Dieser Algorithmus nutzt geschickt die Eigenschaften der Quantencomputer aus, um die Ordnung einer Zahl in der primen Restklassengruppe der gesuchten Zahl zu bestimmen. Daraus kann man mit einigen zahlentheoretischen Überlegungen und klassischen Berechnungen einen nichttrivialen Faktor der Zahl finden und somit die Zahl nach und nach faktorisieren.

$$r = \mathrm{ord}_n(a) \Rightarrow a^{\frac{r}{2}} \not\equiv 1 \pmod{n}$$

$$\text{falls } a^{\frac{r}{2}} \not\equiv -1 \pmod{n}$$

$$n \nmid (a^{\frac{r}{2}} + 1) \qquad n \nmid (a^{\frac{r}{2}} - 1)$$

$$n \mid (a^{\frac{r}{2}} + 1)(a^{\frac{r}{2}} - 1)$$

$$1 < \mathrm{ggT}((a^{\frac{r}{2}} + 1),\ n) < n$$