Quantenberechnungen - Einführung

Gunnar Bergmann

16.6.2017

Quantenberechnungen - Einführung

- Grundlagen und mathematisches Modell
- Einschränkungen
- Simulation klassischer Computer
- Deutschs Algorithmus
- Komplexität

Quantencomputer

- nutzen quantenmechanische Eigenschaften
- können klassische Rechner effizient simulieren
- für bestimmte Aufgaben effizienter
- Quantensysteme haben unintuitives Verhalten
- bisher nur wenige Algorithmen

Quantenalgorithmen

- Simulation von Quantensystemen
- Shors Algorithmus für Primzahlzerlegung in Polynomialzeit
- Grovers Algorithmus für Suche in $\Theta(\sqrt{N})$
- Deutschs Algorithmus
  - einfaches Beispiel
  - demonstriert Vorteile von Quantenalgorithmen
  - später mehr dazu
- Simons Problem zeigt Vorteile bei randomisierten Algorithmen

Quantenbits (Qubits)

- Generalisierung von klassischen Bits
- Basiszustände $|0\rangle$ und $|1\rangle$
- Superposition:
  - Anteile von beiden Zuständen gleichzeitig
  - kann nicht genau bestimmt werden

Quantenbits (Qubits)

Notation

Qubits

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$$

$$\alpha, \beta \in \mathbb{C}$$

$$|\alpha|^2 + |\beta|^2 = 1$$

Für einfache Fälle reicht oft auch $\alpha, \beta \in \mathbb{R}$.

Quantenbits (Qubits)

- exakter Zustand nicht ermittelbar
- Messung ergibt $|0\rangle$ mit Wahrscheinlichkeit $|\alpha|^2$ und $|1\rangle$ mit $|\beta|^2$
- Messung verändert den Zustand zu $|0\rangle$ oder $|1\rangle$

Quantenregister

- mehrere Qubits
- stellt Gesamtzustand aller Qubits dar
- Operationen werden auf ganzen Registern statt einzelnen Bits definiert.

Quantenregister

Beispiel

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}$$

Quantenregister

Beispiel

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}$$

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$$

Quantenschaltkreise

- klassische Gatter können als Wahrheitstabellen dargestellt werden
- Quantengatter müssen alle Zustände behandeln
- viele weitere Einschränkungen
- nur azyklische Schaltkreise betrachtet
- Beispiel: NOT-Gatter

NOT-Gatter

- $\mathrm{NOT}(|0\rangle) = |1\rangle$
  $\mathrm{NOT}(|1\rangle) = |0\rangle$

NOT-Gatter

- $\mathrm{NOT}(|0\rangle) = |1\rangle$
  $\mathrm{NOT}(|1\rangle) = |0\rangle$
- Generalisierung: $\mathrm{NOT}(\alpha\,|0\rangle + \beta\,|1\rangle) = \beta\,|0\rangle + \alpha\,|1\rangle$

NOT-Gatter

- $\mathrm{NOT}(|0\rangle) = |1\rangle$
  $\mathrm{NOT}(|1\rangle) = |0\rangle$
- Generalisierung: $\mathrm{NOT}(\alpha\,|0\rangle + \beta\,|1\rangle) = \beta\,|0\rangle + \alpha\,|1\rangle$
- Ausgabe muss wieder Qubit sein
- Länge bleibt erhalten: $|\alpha|^2 + |\beta|^2 = 1$

NOT-Gatter

- $\mathrm{NOT}(|0\rangle) = |1\rangle$
  $\mathrm{NOT}(|1\rangle) = |0\rangle$
- Generalisierung: $\mathrm{NOT}(\alpha\,|0\rangle + \beta\,|1\rangle) = \beta\,|0\rangle + \alpha\,|1\rangle$
- Ausgabe muss wieder Qubit sein
- Länge bleibt erhalten: $|\alpha|^2 + |\beta|^2 = 1$
- Darstellung als Matrix-Vektor-Multiplikation:
  $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$

Quantenschaltkreise - weitere Eigenschaften

- alle Quantengatter sind als Matrizen darstellbar
- Matrizen sind unitär: $U^\dagger U = UU^\dagger = I$
  ($U^\dagger$ ist konjugiert transponierte Matrix)
  - Matrizen sind quadratisch: Gatter haben gleiche Eingabe- und Ausgabegröße
  - Alle Berechnungen sind linear
  - Alle Schaltkreise sind invertierbar:
    Viele Funktionen (Bits kopieren, AND, OR, XOR) nicht direkt umsetzbar

Beispiel: CNOT

Umsetzung durch Kontrollbits

Beispiel

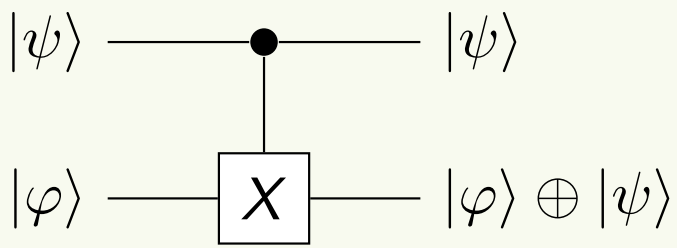
- CNOT ist verallgemeinertes XOR
- $|\psi, \varphi\rangle \rightarrow |\psi, \varphi \oplus \psi\rangle$

Beispiel: CNOT

Umsetzung durch Kontrollbits

Beispiel

- CNOT ist verallgemeinertes XOR
- $|\psi, \varphi\rangle \rightarrow |\psi, \varphi \oplus \psi\rangle$



Beispiel: CNOT

Umsetzung durch Kontrollbits

Beispiel

- CNOT ist verallgemeinertes XOR
- $|\psi, \varphi\rangle \rightarrow |\psi, \varphi \oplus \psi\rangle$
- Als Matrix: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
- CNOT ist unitär, da $\text{CNOT}^\dagger = \text{CNOT}$

No cloning theorem

Axiom

Quantenbits können im Allgemeinen nicht geklont werden.

No cloning theorem

Axiom

Quantenbits können im Allgemeinen nicht geklont werden.

- kein Gatter $|\psi, \varphi\rangle \rightarrow |\psi, \psi\rangle$
- nicht durch unitäre Matrix ausdrückbar

No cloning theorem

Axiom

Quantenbits können im Allgemeinen nicht geklont werden.

- kein Gatter $|\psi, \varphi\rangle \rightarrow |\psi, \psi\rangle$
- nicht durch unitäre Matrix ausdrückbar
- scheinbarer Widerspruch:
  CNOT mit $\varphi = |0\rangle$ ergibt $|\psi, 0\rangle \rightarrow |\psi, \psi\rangle$

No cloning theorem

- Sei $\psi = \alpha |0\rangle + \beta |1\rangle$
- Dann gilt: $|\psi, 0\rangle = \alpha |00\rangle + \beta |01\rangle$
- CNOT$(|\psi, 0\rangle) = \alpha |00\rangle + \beta |11\rangle$

No cloning theorem

- Sei $\psi = \alpha |0\rangle + \beta |1\rangle$
- Dann gilt: $|\psi, 0\rangle = \alpha |00\rangle + \beta |01\rangle$
- CNOT$(|\psi, 0\rangle) = \alpha |00\rangle + \beta |11\rangle$
- Aber:

$$\begin{aligned} |\psi, \psi\rangle &= [\alpha |0\rangle + \beta |1\rangle] [\alpha |0\rangle + \beta |1\rangle] \\ &= \alpha^2 |00\rangle + \alpha\beta |01\rangle + \alpha\beta |10\rangle + \beta^2 |11\rangle \end{aligned}$$

No cloning theorem

- Sei $\psi = \alpha |0\rangle + \beta |1\rangle$
- Dann gilt: $|\psi, 0\rangle = \alpha |00\rangle + \beta |01\rangle$
- CNOT$(|\psi, 0\rangle) = \alpha |00\rangle + \beta |11\rangle$
- Aber:

$$\begin{aligned} |\psi, \psi\rangle &= [\alpha |0\rangle + \beta |1\rangle] [\alpha |0\rangle + \beta |1\rangle] \\ &= \alpha^2 |00\rangle + \alpha\beta |01\rangle + \alpha\beta |10\rangle + \beta^2 |11\rangle \end{aligned}$$

- Gleichheit gilt nur bei $\alpha = 0$ oder $\beta = 0$

Simulation klassischer Computer

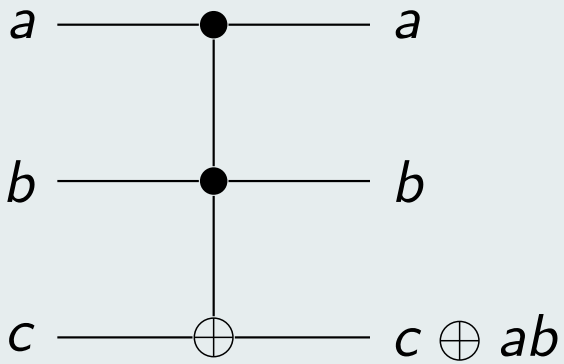
Satz

Jeder klassische Schaltkreis kann auf einem Quantencomputer effizient simuliert werden.

Zentrale Rolle dabei spielt das Toffoli-Gatter

Toffoli-Gatter

$$|a, b, c\rangle \rightarrow |a, b, c \oplus (a \wedge b)\rangle$$



Toffoli-Gatter

$|a, b, c\rangle \rightarrow |a, b, c \oplus (a \wedge b)\rangle$

- invertierbar: zweimal Anwenden ergibt

$$|a, b, c \oplus (a \wedge b) \oplus (a \wedge b)\rangle = |a, b, c\rangle$$

- Kopieren von Bits:
  Für $a = |1\rangle, c = |0\rangle$: $|1, b, 0\rangle \rightarrow |1, b, b\rangle$
- NAND:
  Für $c = |1\rangle$: $|a, b, 1\rangle \rightarrow |a, b, \neg(a \wedge b)\rangle$
- Alle anderen Gatter können über NANDs realisiert werden.

Simulation klassischer Computer

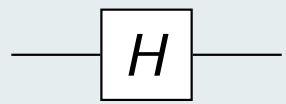
- Mehrere Kabel können an Ausgang angebracht werden
- Über verschaltete NANDs ist jede klassische Schaltung realisierbar
- Simulation ist effizient: Jedes Gatter wird durch konstant viele Toffoli-Gatter ersetzt

Deutschs Algorithmus

- einfacher Algorithmus
- zeigt Quantenparallelismus
- Aber: keine reale Anwendung

Vorbereitung: Hadamard-Gatter

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



$$|+\rangle = H \cdot |0\rangle = H \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$|-\rangle = H \cdot |1\rangle = H \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Deutschs Algorithmus

- gegeben: Funktion $f(x) : \{0, 1\} \to \{0, 1\}$
- entscheide, ob $f(0) = f(1)$
- alternativ: Berechne $f(0) \oplus f(1)$
- klassischer Algorithmus: Berechne jeweils $f(0)$ und $f(1)$.
- Deutschs Algorithmus löst das Problem mit einer Auswertung von f.

Deutschs Algorithmus

- gegeben: Funktion $f(x) : \{0,1\} \to \{0,1\}$
- Sei U_f Quantengatter und setze $|x,y\rangle \to |x, y \oplus f(x)\rangle$ um.
- Für $y = |0\rangle$ kann $f(x)$ berechnet werden.
- Stattdessen: $x = |+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$.
- Dann U_f anwenden:

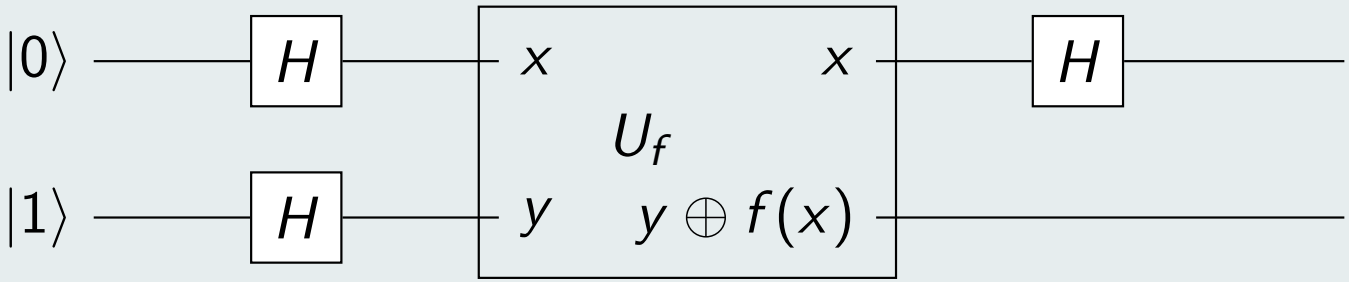
$$|x, f(x)\rangle = \frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}}$$

- Problem: Es werden zwar $f(0)$ und $f(1)$ berechnet, aber man erhält beim Messen nur jeweils eines.

Deutschs Algorithmus: Vorüberlegungen

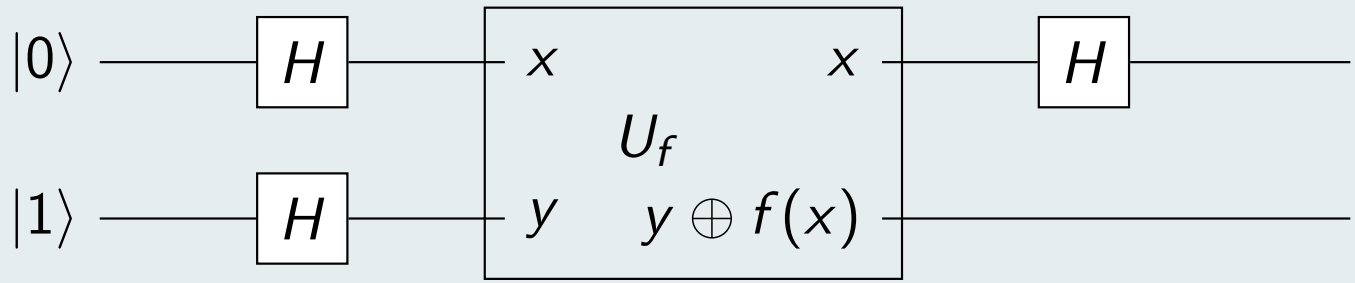
- Quanteninterferenz
- Sei nun wieder x beliebig.
  Setze $y = |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$ und wende U_f an.
- $$U_f \cdot \left(|x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \right) = (-1)^{f(x)} |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]$$

Deutschs Algorithmus



$$|\psi_1\rangle = \left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

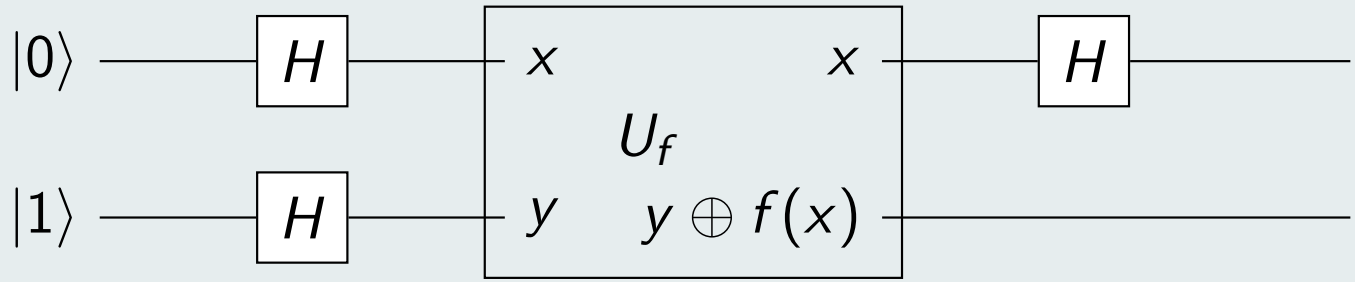
Deutschs Algorithmus



Anwendung von U_f ergibt

$$|\psi_2\rangle = \begin{cases} \pm \left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right] \left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & \text{für } f(0) = f(1) \\ \pm \left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & \text{für } f(0) \neq f(1) \end{cases}$$

Deutschs Algorithmus



$$|\psi_3\rangle = \left\{ \begin{array}{ll} \pm|0\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] & \text{für } f(0) = f(1) \\ \\ \pm|1\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] & \text{für } f(0) \neq f(1) \end{array} \right\}$$

$$= \pm|f(0) \oplus f(1)\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

Deutsch-Jozsa Algorithmus

- Generalisierung von Deutschs Algorithmus auf n bits.
- gegeben: Funktion $f(x) : \{0,1\}^n \to \{0,1\}$
- entscheide, ob f konstant oder balanciert (Hälfte 0, Hälfte 1) andere Werte treten nicht auf
- auf klassischem Rechner: $\Theta(2^n)$
- auf Quantenrechner in Linearzeit mit $\Theta(n)$ Qubits

Komplexität

- Simulation von klassischen Schaltkreisen ohne Zeitverlust
- Deutsch-Jozsa-Algorithmus exponentiell schneller
- Polynomialzeit auf Quantenalgorithmen BQP:
  bounded error quantum polynomial time
- $P \subseteq BPP \subseteq BQP \subseteq PSPACE$
- NP ? BQP

Zusammenfassung

- Quantenrechner sind (vermutlich) mächtiger als klassische
- Bei Komplexität ist noch vieles unbekannt
- noch keine nutzbaren Quantenrechner
- experimentelle Systeme mit wenigen Qubits konnten Quantenalgorithmen nutzen
- Möglichkeit zur technischen Realisierung
- bisher nur wenige Algorithmen