Wir vertauschen die Reihenfolge der Summation und nutzen die Linearität des Skalarproduktes: $u \cdot (v \oplus w) = u \cdot v \oplus u \cdot w$.

$$|\phi_2\rangle = \frac{1}{2^n} \sum_{w=0}^{2^n-1} \sum_{z=0}^{2^n-1} (-1)^{z \cdot (x \oplus w)} |w\rangle.$$

Für $w = x$ ist $z \cdot (x \oplus w) = 0$; wir zerlegen die Summe über w:

$$|\phi_2\rangle = \frac{1}{2^n} \left(\sum_{z=0}^{2^n-1} (-1)^0 \cdot |x\rangle + \sum_{\substack{w=0 \\ w \neq x}}^{2^n-1} \sum_{z=0}^{2^n-1} (-1)^{z \cdot (x \oplus w)} |w\rangle \right).$$

Wir untersuchen

$$\sum_{z=0}^{2^n-1} (-1)^{z \cdot (x \oplus w)}$$

für den Fall $x \neq w$ und alle möglichen Werte von z. Dann ist $x \oplus w$ nicht der Nullvektor. Wir nehmen an, in $x \oplus w$ seien gerade die Bits $b_1, \ldots, b_m$ gleich 1. Es gilt:

$$z \cdot (x \oplus w) = \begin{cases} 0 & \text{wenn in } z \text{ eine gerade Anzahl} \\ & \text{der Bits } b_1, \ldots, b_m \text{ den Wert 1 hat} \\ 1 & \text{sonst.} \end{cases}$$

Wir betrachten alle möglichen Werte von z: für die Hälfte ist die Anzahl der Bits $b_1, \ldots, b_m$ mit Wert 1 gerade, für die andere Hälfte ungerade. Damit gilt für alle w, die ungleich x sind:

$$\sum_{z=0}^{2^n-1} (-1)^{z \cdot (x \oplus w)} = 0,$$

und der Zustand nach der zweiten Hadamard-Transformation ist

$$|\phi_2\rangle = \frac{1}{2^n} \left(2^n \cdot |x\rangle + \sum_{\substack{w=0 \\ w \neq x}}^{2^n-1} 0 \cdot |w\rangle \right) = |x\rangle.$$

Wir fassen zusammen:

$$|x\rangle \xrightarrow{H_n} \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} (-1)^{x \cdot z} \cdot |z\rangle \xrightarrow{H_n} |x\rangle.$$

2.13 Der Algorithmus von Deutsch-Jozsa

Im vorherigen Abschnitt wurde die Wirkungsweise der Hadamard-Transformation auf ein Register untersucht. Das hilft uns, eine Verallgemeinerung des Problems von Deutsch zu lösen.

Aufgabenstellung

Uns liegt ein Bauteil vor, das eine mathematische Funktion berechnet. Es bildet Eingaben aus n Bits auf ein Ausgabebit ab, das heißt eine Funktion $f : \{0,1\}^n \to \{0,1\}$ wird berechnet.

Wir wissen über die berechnete Funktion f folgendes:

- Entweder ist f *konstant* und alle Eingaben werden auf die gleiche Ausgabe abgebildet,
- oder f ist *balanciert*: die Hälfte der Eingaben wird auf 1 abgebildet, die andere Hälfte auf 0. Dann gilt für die Anzahl der Urbilder $|f^{-1}(0)| = |f^{-1}(1)| = 2^{n-1}$.

Unsere Aufgabe: Wir sollen entscheiden, welche der Alternativen zutrifft.

Aufwand im klassischen Fall

Wir überlegen zunächst, für wie viele Eingaben ein klassischer Computer die Funktion auswerten muss. Auch wenn er geschickt vorgeht, kann folgende Situation eintreten: er hat bereits für $\nu = 2^{n-1}$ verschiedene Eingaben $b_1, \dots b_\nu \in \{0,1\}^n$ den Funktionswert bestimmt und herausbekommen, dass $f(b_1) = \dots = f(b_\nu)$ gilt. Er weiß dann allerdings noch immer nicht, ob die Funktion balanciert oder konstant ist. Der klassische Rechner muss also im schlechtesten Fall für $2^{n-1}+1$ Eingaben den Funktionswert bestimmen. Die Anzahl der Aufrufe ist im schlechtesten Fall exponentiell in der Eingabegröße n.

Liegt uns die Funktion hingegen als Quantenbauteil

$$U_f : |x_{n-1} \dots x_0, y\rangle \mapsto |x_{n-1} \dots x_0, y \oplus f(x)\rangle$$

vor, kommt ein Quantencomputer mit nur einem Aufruf aus. Der folgende Algorithmus stammt von David Deutsch und Richard Jozsa von der Universität Bristol (siehe [34]).

Der Algorithmus von Deutsch-Jozsa

Wir verwenden ein Register mit $n+1$ Quantenbits $|x_{n-1} \dots x_0\rangle|y\rangle$, siehe auch Abbildung 2.20.

1. $|x_{n-1} \dots x_0\rangle|y\rangle \leftarrow |0 \dots 0\rangle|1\rangle$
2. Wende die Hadamard-Transformation H_{n+1} an:
$|x\rangle|y\rangle \leftarrow H_{n+1}|x\rangle|y\rangle$
3. Werte f aus:
$|x\rangle|y\rangle \leftarrow U_f|x\rangle|y\rangle$
4. Wende die Hadamard-Transformation auf $|x\rangle$ an:
$|x\rangle|y\rangle \leftarrow (H_n|x\rangle)|y\rangle$
5. Miss das Register:
Ist $|x\rangle = |0 \dots 0\rangle$: Ausgabe *konstant*,
Sonst: Ausgabe *balanciert*.

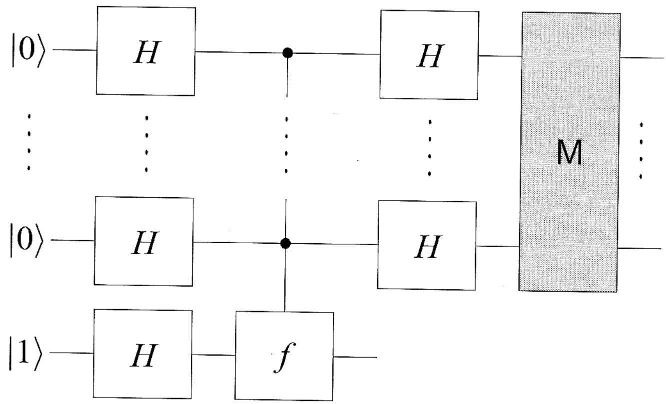

Abbildung 2.20: Schaltkreis für den Algorithmus von Deutsch-Jozsa

Analyse

Schritt 2

Wir beginnen mit Schritt 2, der Anwendung von H_{n+1} auf $|0\dots0\rangle|1\rangle$. Dies entspricht der Anwendung von H_n auf $|0\dots0\rangle$ zusammen mit H auf $|1\rangle$:

$$|0\dots0\rangle|1\rangle \xrightarrow{H_{n+1}} \left(\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle\right)\cdot\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)=|\phi_2\rangle.$$

Schritt 3

Zunächst betrachten wir, wie die Anwendung von U_f im Schritt 3 für ein fixiertes x wirkt:

$$\begin{aligned}|x\rangle\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle) \quad &\xrightarrow{U_f} \quad |x\rangle\frac{1}{\sqrt{2}}(|f(x)\rangle-|1\oplus f(x)\rangle)\\ &= \quad (-1)^{f(x)}\cdot|x\rangle\cdot\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle).\end{aligned}$$

Also hat U_f auf die Superposition $|\phi_2\rangle$ die Wirkung:

$$|\phi_2\rangle \quad \xrightarrow{U_f} \quad \left(\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle\right)\cdot\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)=|\phi_3\rangle.$$

Schritt 4

Im vierten Schritt wird die Hadamard-Transformation auf den $|x\rangle$-Teil angewendet, für Zwischenschritte verweisen wir auf den vorhergehenden Abschnitt.

$$|\phi_3\rangle \xrightarrow{H_n\otimes I_2} \left(\frac{1}{2^n}\sum_{z=0}^{2^n-1}\sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot z}|z\rangle\right)\cdot\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)=|\phi_4\rangle.$$

Schritt 5

Um die Auswirkung der Messung zu untersuchen, betrachten wir $|x_{n-1}\dots x_0\rangle$ für ein festes $|z\rangle$, also die (nicht normierte) Summe

$$|\phi_4'\rangle=\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot z}|z\rangle.$$

Nehmen wir zunächst an, f sei *konstant*. Dann ist $f(x)$ für alle x gleich. Im Fall $z=0$ gilt außerdem $x \cdot z = 0$ und wir erhalten in diesem Fall:

Fall: konstant

$$|\phi_4'\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} \pm|0\rangle = \pm|0\rangle.$$

Messen von $|x_{n-1} \dots x_0\rangle$ liefert also das Ergebnis $|0\rangle$. Da $|\phi_4'\rangle$ ein zulässiger Zustand von $|x_{n-1}, \dots, x_0\rangle$ ist, verschwindet die Amplitude von jedem $|z\rangle$ mit $z \neq 0$. Es ist instruktiv, das nachzurechnen:

Aufgabe 2.22: Zeigen Sie, dass $|z\rangle$ für $z \neq 0$ die Amplitude 0 hat.

Wenn f balanciert ist, mit welcher Wahrscheinlichkeit beobachten wir dann $|0\rangle$? Für $z=0$ ist

Fall: balanciert

$$|\phi_4'\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)}|0\rangle.$$

Für die eine Hälfte der x ist $f(x)=0$, für die andere ist $f(x)=1$. Also ist die Amplitude von $|0\rangle$ gleich 0. ◇

Das folgende Problem lässt sich auf ganz ähnliche Weise lösen: wieder müssen wir etwas über eine Funktion $f: \{0,1\}^n \to \{0,1\}$ herausfinden, die uns als Black Box gegeben ist. Wir wissen, dass es sich um eine der 2^n Funktionen

Bernstein-Vazirani

$$f_a(x) = a \cdot x = a_0x_0 \oplus \dots \oplus a_{n-1}x_{n-1} \text{ für } a \in \{0,1\}^n$$

handelt. Die Funktion ist also über einen Vektor $a \in \{0,1\}^n$ definiert und bildet jede Eingabe auf das Skalarprodukt mit a ab.

Der Algorithmus von Bernstein-Vazirani

Wir verwenden ein Register mit $n+1$ Quantenbits $|x_{n-1} \dots x_0\rangle|y\rangle$

1. $|x_{n-1} \dots x_0\rangle|y\rangle \leftarrow |0 \dots 0\rangle|1\rangle$
2. Wende die Hadamard-Transformation H_{n+1} an:
   $|x\rangle|y\rangle \leftarrow H_{n+1}|x\rangle|y\rangle$
3. Werte f aus:
   $|x\rangle|y\rangle \leftarrow U_f|x\rangle|y\rangle$
4. Wende die Hadamard-Transformation auf $|x\rangle$ an:
   $|x\rangle|y\rangle \leftarrow (H_n|x\rangle)|y\rangle$
5. Miss das Register:
   Ist $|x\rangle = |a\rangle$:
   Ausgabe *Das Bauteil berechnet die Funktion* $f_a(x) = a \cdot x$.

Die Ähnlichkeit mit dem Algorithmus von Deutsch-Jozsa ist so groß, dass die Analyse dem Leser überlassen wird.

Aufgabe 2.23: Analysieren Sie den Algorithmus von Bernstein-Vazirani und zeigen Sie, dass er korrekt arbeitet.

Die Idee stammt von Ethan Bernstein und Umesh Vazirani (1993).