Die Überlegenheit von Quantenalgorithmen Lektion 2– Der Deutsch-Algorithmus

Der von David Deutsch im Jahr 1985 entwickelte **Deutsch-Algorithmus** stellt einen wichtigen Meilenstein in der Geschichte des Rechnens dar. Er war das erste Beispiel dafür, dass ein Quantenalgorithmus einen potenziellen Vorteil gegenüber klassischen Verfahren bieten kann. Diese Entdeckung eröffnete die Perspektive, dass Computer auf Basis quantenmechanischer Prinzipien bestimmte Problemstellungen effizienter lösen könnten als klassische Rechner.

In dieser Lektion lernen Sie, wie der Deutsch-Algorithmus funktioniert. Um seine Bedeutung vollständig zu verstehen, müssen Sie zunächst nachvollziehen, wie sich das Problem, das dieser Algorithmus löst, mit klassischen Verfahren angehen lässt.

Die Aufgabe

Stellen Sie sich vor, wir haben eine Blackbox-Funktion f, die als Eingabe ein einzelnes klassisches Bit $(0 \text{ oder } 1)$ erhält und als Ausgabe wiederum ein einzelnes klassisches Bit $(0 \text{ oder } 1)$ liefert. Insgesamt gibt es vier mögliche Funktionen, die sich in der Blackbox befinden könnten:

$f_1(0) = 0$ $f_1(1) = 0$	$f_2(0) = 1$ $f_2(1) = 1$	$f_3(0) = 0$ $f_3(1) = 1$	$f_4(0) = 1$ $f_4(1) = 0$

Das Problem besteht darin, dass wir nicht wissen, welche dieser Funktionen sich in der Blackbox befindet. Unsere Aufgabe ist es, dies allein dadurch herauszufinden, dass wir die Eingaben der Blackbox verändern und die Ausgaben beobachten.

Aufgabe 1

Die Tabellenkalkulationsdatei „Deutsch Functions“ zeigt, wie diese vier Funktionen arbeiten, und enthält außerdem einen Generator für eine unbekannte Funktion (*mystery function)*. Klicken Sie auf die Schaltfläche „Generate mystery function“, um zufällig eine der vier Funktionen auszuwählen. Wenn Sie die Eingabe in Zelle B12 verändern, wird die Ausgabe der unbekannten Funktion in Zelle E12 angezeigt. Wie können Sie durch Variation der Eingaben herausfinden, welche unbekannte Funktion vorliegt, und wie viele Abfragen sind dafür nötig? Begründen Sie Ihre Antwort.

ENTNOMMEN AUS QUANTEN-COMPUTING IM MINT-UNTERRICHT
Ein Projekt von Science on Stage Deutschland e.V.

Konstante Funktionen und balancierte Funktionen

Nehmen wir nun an, wir ordnen die vier möglichen Funktionen in zwei Klassen ein:

Konstante Funktionen		**Balancierte Funktionen**	
$f_1(0) = 0$ $f_1(1) = 0$	$f_2(0) = 1$ $f_2(1) = 1$	$f_3(0) = 0$ $f_3(1) = 1$	$f_4(0) = 1$ $f_4(1) = 0$

Aufgabe 2

Verwenden Sie erneut die Tabellenkalkulation: Wie viele Abfragen sind nötig, um festzustellen, ob die unbekannte Funktion konstant oder balanciert ist? Wie verhält sich dies im Vergleich dazu, die konkrete Funktion selbst zu bestimmen? Begründen Sie Ihre Antwort.

Zusammenfassung der klassischen Ergebnisse

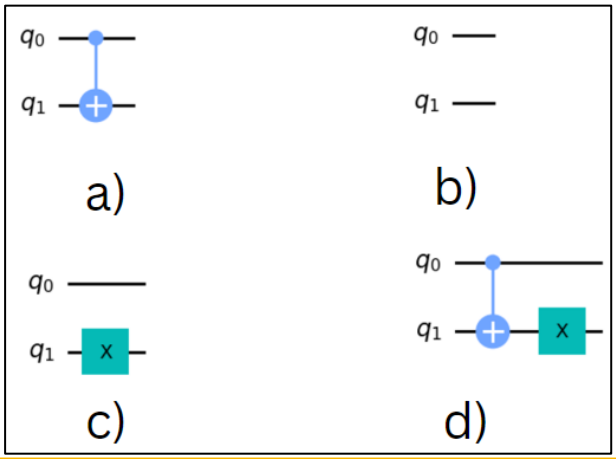
Wir haben gesehen, dass in beiden Fällen zwei Abfragen erforderlich sind, um mit klassischen Suchverfahren das gewünschte Ergebnis zu erhalten. Im ersten Fall – also bei der Bestimmung der konkreten Funktion – würde auch ein Quantencomputer zwei Abfragen benötigen. Im zweiten Fall hingegen – also bei der Entscheidung, ob die Funktion konstant oder balanciert ist – kann Quantencomputing eine deutliche Verbesserung ermöglichen.

Konstruktion der Funktion auf einem Quantencomputer

Der erste Schritt bei der Untersuchung dieses Problems auf einem Quantencomputer besteht darin, die Funktionen mithilfe von Quantengattern zu konstruieren. Es ist möglich, alle vier Funktionen durch Kombinationen von NOT-Gattern, CNOT-Gattern oder auch ganz ohne Gatter darzustellen.

Dazu benötigen wir zwei Qubits: q_0, das die Eingabe der Funktion aufnimmt, und q_1, das die Ausgabe liefert (nachdem es zunächst in den Zustand $|0\rangle$ gesetzt wurde).

Verwenden Sie den IBM Quantum Composer, um die folgenden vier Schaltkreise zu erstellen:



Beispiel

Für den Schaltkreis (a) gilt: Wenn wir das Eingangsqubit q_0 auf den Zustand $|0\rangle$ setzen und das Ausgangsqubit q_1 zunächst ebenfalls auf den Zustand $|0\rangle$, dann sehen wir, dass der Endzustand von q_1 ebenfalls $|0\rangle$ ist, da das CNOT-Gatter nicht ausgelöst wird.

Das bedeutet: Eine Eingabe von 0 wird von der Funktion auf eine Ausgabe von 0 abgebildet.

Wenn wir nun das Eingangsqubit q_0 auf den Zustand $|1\rangle$ setzen, sehen wir, dass der Endzustand von $q_1$nun $|1\rangle$ ist, da das CNOT-Gatter jetzt ausgelöst wird und den Zustand von q_1 invertiert.

Das bedeutet: Eine Eingabe von 1 wird von der Funktion auf eine Ausgabe von 1 abgebildet.

Diese Zuordnung entspricht der Funktion f_3 (der ersten der balancierten Funktionen).

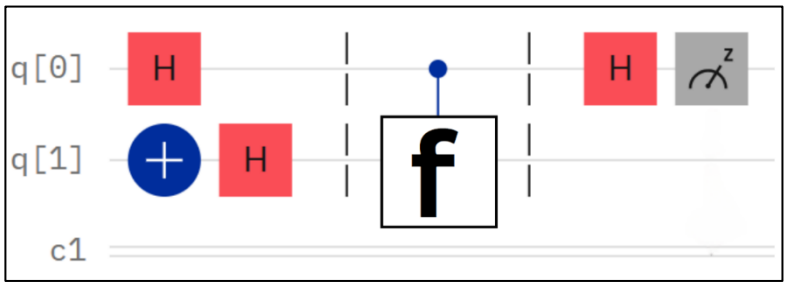
Aufgabe 3

Die Tabellenkalkulationsdatei „Quantum Circuits“ enthält die vier dargestellten Schaltkreise. Der Eingabezustand von q_0 ist in den fett gedruckten blauen Zellen dargestellt, der Endzustand des Ausgangsqubits q_1 in den fett gedruckten roten Zellen. Bestimmen Sie durch Variation der Anfangszustände von q_0 und durch Beobachtung der Endzustände von q_1, welcher Funktion (f_1, f_2, f_3 oder f_4) die Schaltkreise (b), (c) und (d) jeweils entsprechen. Tragen Sie Ihre Antworten unten ein:

Bestimmung der unbekannten Funktion mit einem Quantencomputer

Die Technik, die wir hier verwenden, nutzt Superposition durch die Anwendung von Hadamard-Gattern. Dies ist einer der wichtigsten Tricks des Quantencomputings, und wir werden ihn in jedem Quanten-Suchalgorithmus dieses Kurses wiedersehen.

Erstellen Sie den folgenden Schaltkreis im IBM Quantum Composer:



Beachten Sie, dass das **NOT-Gatter** verwendet wird, um den Zustand von q_1 auf $|1\rangle$ zu initialisieren, während das Fehlen eines NOT-Gatters auf q_0 bedeutet, dass dieses Qubit auf den Zustand $|0\rangle$ initialisiert wird.

Aufgabe 4

Diese Aufgabe kann entweder mit dem Quantum Composer oder mit der Tabellenkalkulationsdatei „Quantum Circuits with Hadamards“ bearbeitet werden. Stellen Sie q_0 auf den Zustand $|0\rangle$ ein und q_1 auf den Zustand $|1\rangle$. Fügen Sie anschließend nacheinander die Schaltkreise für f_1, f_2, f_3 und f_4 in den Abschnitt des Schaltkreises ein, der mit der Blackbox f gekennzeichnet ist. Betrachten Sie die Werte, die q_0 nach der Messung annimmt. Wie kann diese Information dazu genutzt werden, zu bestimmen, ob eine Funktion balanciert oder konstant ist? Wie viele Abfragen sind nun erforderlich, um dies zu zeigen? Tragen Sie Ihre Antworten unten ein:

Zusammenfassung

Durch die Nutzung von Superposition konnten wir bestimmen, ob die Funktion konstant oder balanciert ist, und dafür nur eine einzige Abfrage statt zwei benötigen. Damit kann ein Quantencomputer dieses Problem mit weniger Abfragen lösen als ein klassischer Computer.

Der Deutsch-Algorithmus macht außerdem zwei wichtige Punkte deutlich:

(1) Quantencomputer sind nicht in jeder Situation klassischen Computern überlegen. Wie wir hier gesehen haben, hilft der Quantenalgorithmus nur bei der Unterscheidung zwischen konstanten und balancierten Funktionen, nicht jedoch bei der Bestimmung, welche konkrete Funktion sich in der Blackbox befindet. Ob ein Problem durch einen Quantencomputer effizienter gelöst werden kann, muss daher immer sorgfältig geprüft werden.

(2) Die Lösung dieses Problems wurde durch die Betrachtung des Eingangsqubits gefunden, nicht durch die Betrachtung des Ausgangsqubits. Die Information wurde im Eingangsqubit gewonnen, während im Ausgangsqubit Information verloren ging. Quantencomputing ist häufig mit solchen Trade-offs verbunden, und Lösungen müssen oft an unerwarteten Stellen gesucht werden.

Obwohl das Deutsch-Problem ein etwas künstliches Beispiel ist, zeigte es das Potenzial von Quantencomputern und ebnete den Weg für weitere, inhaltlich bedeutendere Quantenalgorithmen, die wir in den nächsten Lektionen kennenlernen werden.