Das Deutsch-Jozsa-Problem - Lösung mit Quantencomputern

Florian Klotz

Das Deutsch-Jozsa-Problem ist eines der ältesten Probleme, mit dem bewiesen werden konnte, das Quantencomputer einige Probleme schneller als klassische Computer lösen können. Im Folgen soll das Deutsch-Jozsa-Problem erläutert, und als quantenmechanischer Algorithmus gelöst werden.

I. DAS DEUTSCH-JOZSA-PROBLEM

A. Allgemeines

Das Deutsch-Joza-Problem beschreibt die Frage, ob eine Funktion balanciert oder konstant ist. Gegeben ist dafür eine unbekannte Funktion

$$f : \{0,1\}^n \to \{0,1\} .$$

Sie gibt für jedes Eingabebit einen Wert aus. Dabei steht n für die Anzahl der Eingangsbits und $\{0,1\}$ für die Werte, die Ein- und Ausgangsbits annehmen können, also 0 und 1. Zudem ist zugesichert, dass die Funktion entweder konstant oder balanciert ist. f ist unbekannt, also als "Black Box" gegeben. Somit kann die Definition nicht zum erschließen des Problems verwendet werden. Zudem wird sie meißt als "Oracle" bezeichnet.

Zudem gibt es eine Anzahl $N = 2^n$ möglicher binärer Eingaben, die auf die Eingabebits geschrieben werden und jeden Zustand beschreiben, die die Eingabebits annehmen können. Die Funktion ist konstant, wenn sie für alle Eingangswerte den selben Ausganswert zurückgibt. Balanciert ist sie, wenn sie für exakt die Hälfte der Eingabewerte den einen und für die andere Hälfte den anderen Wert zurückgibt.

B. Geschichte

Das Deutsch-Jozsa-Problem ist eine Erweiterung des Deutsch-Problems[1]. Es wurde 1992 durch Richard Jozsa verallgemeinert, nachdem David Deutsch 1985 das nach ihm benannte Problem aufgestellt hatte. Dieses beschäftigt sich mit dem selben Ansatz wie das Deutsch-Jozsa-Problem, jedoch mit nur einem Bit, also $n = 1$.

Somit ist die Funktion konstant, wenn

$$f(0) = f(1) .$$

bzw. balanciert wenn

$$f(0) \neq f(1) .$$

II. LÖSUNG DES DEUTSCH-JOZSA-PROBLEMS

Das Deutsch-Jozsa-Problem hat zwar grundsätzlich keine Realen Anwendungen, außer Geschwindigkeitssteigerungen durch Quantencomputer zu zeigen, kann aber etwas leichter vorstellbar gemacht werden. So kann sich die Funktion, die betrachtet wird, als teurer z.B. elektrischer Komponent oder Chip vorgestellt werden. Ziel ist es den Chip so wenig wie möglich zu benutzen, um die Abnutzung möglichst gering zu halten.

Für die folgenden Lösungswege wird die balancierte Funktion

$$f(x) = x_0 \oplus x_1 x_2$$

verwendet. Die Funktion hat also drei Eingabebits. Zuerst wird x_1, also Bit 1 mit x_2, also Bit 2 multipliziert. Dieses Ergebniss wird dann binär auf x_0, das 0-te Bit addiert. So ergibt sich folgende Ergebnisstabelle:

Binäre Darstellung $x_2x_1x_0$	Funktion $f(x) = x_0 \oplus x_1x_2$
000	0
001	1
010	0
011	1
100	0
101	1
110	1
111	0

A. Klassische Lösung des Deutsch-Jozsa-Problems:

Klassisch wird das Problem durch einfaches Ausprobieren gelöst. Da zugesichert wird, dass die Funktion nur balanciert oder konstant sein kann, kann die Antwort durch Ausschlussverfahren ermittelt werden. Falls mehr als die Hälfte der Eingabewerte den selben Wert haben, muss die Funktion konstant sein, da für eine balancierte Funktion nur die Hälfte der Ausgaben gleich sein dürfen. Unterscheiden sich jedoch mindestens zwei Ausgabewerte voneinander, so muss die Funktion balanciert sein, da bei einer konstanten Funktion alle Ausgaben gleich sind.

So ergibt sich eine Maximalanzahl an Eingaben, die jedoch regulär ausprobiert werden müssen

$$N_{max} = \frac{1}{2}N + 1 = 2^{n-1} + 1 .$$

Die Maximalanzahl beschreibt zudem auch die Maximale Laufzeit, da für jede Eingabe die selben Operatoren angewandt werden.

Bei der gegebenen Funktion $f(x) = x_0 \oplus x_1x_2$ können z.B. die Kombinationen 000 bis 011 oder 100 bis 111 ausprobiert werden.

B. Lösung mit Quantencomputern

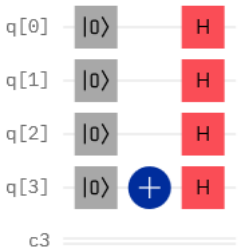
Mithilfe von Quantencomputern kann das Problem wesentlich schneller gelöst werden, der benötigte Algorithmus ist jedoch wesentlich komplizierter.

Zuerst werden die $n+1$ Qubits auf den Basisvektor $|0\rangle$ initialisiert. Das $n+1$te, sog. Ancilla Bit wird danach invertiert, um den Basisvektor $|1\rangle$ darzustellen. Danach

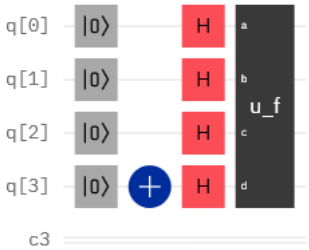
wird auf alle Bits n das sog. Hadamard-Gate angewandt. Dadurch werden die Qubits in Superpositionen zwischen $|0\rangle$ und $|1\rangle$ gebracht. Die Qubits haben also gleichzeitig den Wert $|0\rangle$ und $|1\rangle$ und nehmen alle möglichen Eingaben gleichzeitig ein. Dies wird wie folgt dargestellt:

$$\begin{aligned}\hat{\mathcal{H}}|0\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = |+\rangle\,, \\ \hat{\mathcal{H}}|1\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} = |-\rangle\,.\end{aligned}$$

Als Circuit:



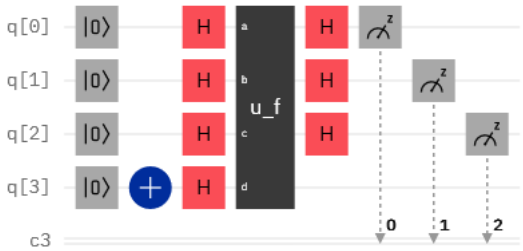
Danach werden die resultierenden Eingabebits $|+\rangle$ und das Ancilla Bit $|-\rangle$ in die Abbildung von f auf die Basisvektoren $|0\rangle$ und $|1\rangle$ gegeben und dort verrechnet. Das Ergebnis der Eingabe wird auf das Ancilla bit binär addiert[1]. Dies bildet insgesamt das Oracle U_f welches auf alle Bits angewandt wird:



Das Ancilla-Bit, auf dem theoretisch indirekt das Ergebniss steht wird nun weggeworfen, da es nicht weiter gebraucht wird. Auf die restlichen Qubits werden wieder Hadamard-Gates angewandt. Danach werden die drei Eingangsqubits gemessen:

Ist die Funktion balanciert, so ist die Wahrscheinlichkeit $|000\rangle$[2], also auf jedem Qubit 0 zu messen gleich 0. Kann also nie gemessen werden. Ist die Funktion konstant, so ist die Wahrscheinlichkeit $|000\rangle$ zu messen 1[2]. Daher wird auf jedem Qubit immer 0 gemessen. So kann das Deutsch-Jozsa-Problem mit nur einer einzigen Messung der Funktion gelöst werden.

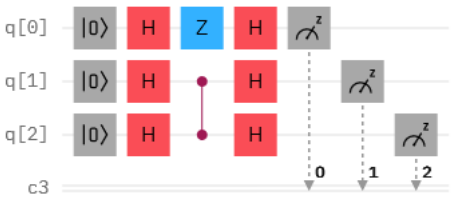
Leider erschließt sich mir die Funktion des Ancilla-Bits nicht. Bei Quantencomputern müssen alle Operationen umkehrbar sein. Ancilla-Bits werden z.B. verwendet,



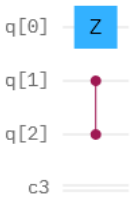
wenn dies nicht der Fall ist[3]. Weiterhin werden sie verwendet, um Circuits zu vereinfachen usw.[3].

Tatsächlich funktioniert der Algorithmus wie oben dargestellt nicht, da mit einer Wahrscheinlichkeit von ca. 25% $|000\rangle$ gemessen wird.

Ein funktionierender Versuchsaufbau ist:



wobei



die Funktion $f(x) = x_0 \oplus x_1 x_2$ darstellt.

Leider habe ich kein genaues Verständniss, wie genau das Ancilla-Bit in eine beliebige Funktion bzw. in Lösungen des Deutsch-Jozsa-Problems integriert werden könnte. Zudem hat auch der Kursleiter mit dem ich in Korrespondenz stand keine zufriedenstellende Antwort gegeben und hat mir zwar eine Rechnung angeboten, sich jedoch seitdem nichtmehr gemeldet.

Zukünftige Änderungen an diesem Dokument können ggf. eingesehen werden unter: `https://github.com/DeLumberjack/Schuelerakademie.git`

[1] Wikipedia, "Deutsch–jozsa algorithm." https://en.wikipedia.org/wiki/Deutsch%E2%80%93Jozsa_algorithm, Juli 2022. Zuletzt aufgerufen am 27/08/2022.

[2] Wikipedia, "Deutsch-jozsa-algorithmus." https://de.wikipedia.org/wiki/Deutsch-Jozsa-Algorithmus,, März 2022. Zuletzt aufgerufem am 27/08/2022.

[3] Wikipedia, "Ancilla bit." https://en.wikipedia.org/wiki/Ancilla_Bit,, September 2021. Zuletzt aufgerufen am 27/08/2022.