Softwareprogrammierung für Quantenrechner

Lösen von Färbungsproblemen (z. B. Sudoku-Rätsel) mit Quantenalgorithmen

FH MÜNSTER
University of Applied Sciences

Ausgangslage

Quantenrechner werden immer leistungsfähiger und versprechen seit geraumer Zeit eine grundlegende Veränderung für die Informatik. Beispielsweise könnte ein Quantenrechner die bestehenden Verschlüsselungsfunktionen in nicht exponentieller Zeit entschlüsseln. Quantenrechner basieren auf den Gesetzen der Quantenmechanik. Die Recheneinheiten bestehen aus sogenannten „Qubits“, die sich im Gegensatz zu klassischen Bits in mehreren Zuständen gleichzeitig befinden können (siehe Abbildung 1).

In diesem Forschungs- und Entwicklungsprojekt wird untersucht, wie nach aktuellem Stand mit Quantenrechnern programmiert werden kann, und welche Grenzen dabei auftreten und existieren.

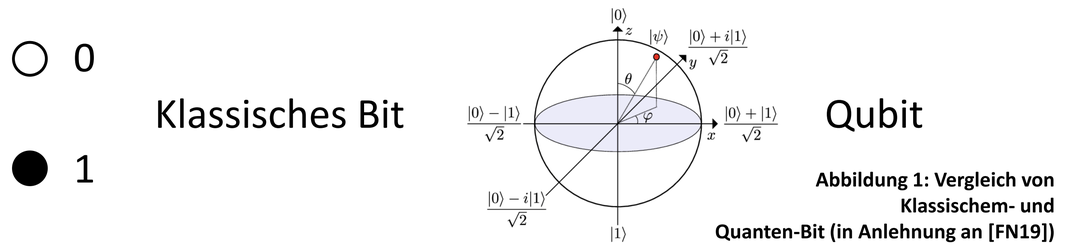

Abbildung 1: Vergleich von Klassischem- und Quanten-Bit (in Anlehnung an [FN19])

Vorgehen

- Einarbeitung in die Quanteninformatik
- Analyse bestehender Programmierplattformen
- Sudoku als Kernproblem
- Erstellung des Sudokus in klassischer Programmierung
- Lösung über Grover-Algorithmus auf Quantensimulator
- Entwicklung verschiedener Ansätze in Kleingruppen

Entwicklung von Quantenalgorithmen

Im Gegensatz zu klassischen Algorithmen sind Quantenalgorithmen probabilistisch und nicht-deterministisch. Für die Ermittlung des Ergebnisses bedeutet dies, dass der gleiche Quantenalgorithmus unterschiedliche Ergebnisse liefern kann. Erst bei mehrfacher Ausführung des Quantenalgorithmus kristallisiert sich das korrekte Ergebnis heraus, indem dieses am häufigsten auftritt. Nachdem ein Quantenschaltkreis ausgeführt wurde, kann das Ergebnis durch Messungen ermittelt werden. Dies führt nach den physikalischen Gesetzen dazu, dass jeder Quantenzustand in einen klassischen Zustand zerfällt. Hierdurch kann eine eindeutige Lösung ermittelt werden. Quantenalgorithmen dieser Art werden anhand von Quantenschaltkreisen implementiert.

Ein Quantenschaltkreis beschreibt eine bestimmte Abfolge von Quantengattern, welche auf Qubits angewendet werden. Abbildung 3 zeigt einen Quantenschaltkreis. Jede horizontale Linie stellt ein Qubit dar. Die auf den Linien abgebildeten Elemente stellen Quantengatter dar. Gegenwärtig ist Quantenhardware noch teuer und fehleranfällig, da es noch nicht möglich ist die theoretischen Konzepte physikalisch exakt abzubilden. Aus den genannten Gründen kann für die Entwicklung und Ausführung von Quantenschaltkreisen auf Simulatoren zurückgegriffen werden. Zur Erstellung und Simulation von Quantenschaltkreisen können SDKs verschiedener Anbieter verwendet werden. Im Projekt wurde dazu unter anderen IBM-Qiskit [Qis23] verwendet.

Vorstellung des Grover-Algorithmus

Der Grover-Algorithmus (siehe Abbildung 2) ist ein weitverbreiteter und bekannter Quantenalgorithmus. Er dient der Suche in einer unsortierten „Datenbank“. Ein klassischer Algorithmus benötigt für die Suche nach einem Element in einer unsortierten Liste mit N Einträgen eine Laufzeit von $\mathcal{O}(N)$. Der Grover-Algorithmus benötigt hierfür nur eine Laufzeit von $\mathcal{O}(\sqrt{N})$. Dies wird ermöglicht durch eine Orakel-Funktion und den Diffuser. Das Orakel „kennt“ gewissermaßen das gesuchte Element und markiert dieses. Der Diffuser erhöht die Wahrscheinlichkeit, dass das markierte Element vom Quantenrechner ermittelt wird. [Ho22]

Abbildung 2: Bestandteile des Grover-Algorithmus

Ergebnis

Im Rahmen des Projekts wurde ein Python-Programm entwickelt, welches für beliebige Sudokus einen Quantenschaltkreis generieren kann. Dabei wurden unterschiedliche Ansätze benutzt, um die Anzahl der Qubits gering zu halten. Ein Ansatz besteht darin die Qubits in vorgefertigte Superpositionen zu versetzen. Ein anderer Ansatz besteht darin, die Qubits durch Regellisten zu verringern. Die Abbildung 3 zeigt einen Quantenschaltkreis, welcher unter Nutzung der Grover-Algorithmus ein 2x2-Sudoku lösen kann. Über die QR-Codes in der Fußzeile des Plakats können die erstellten Anwendungen inklusive Dokumentation gefunden werden.

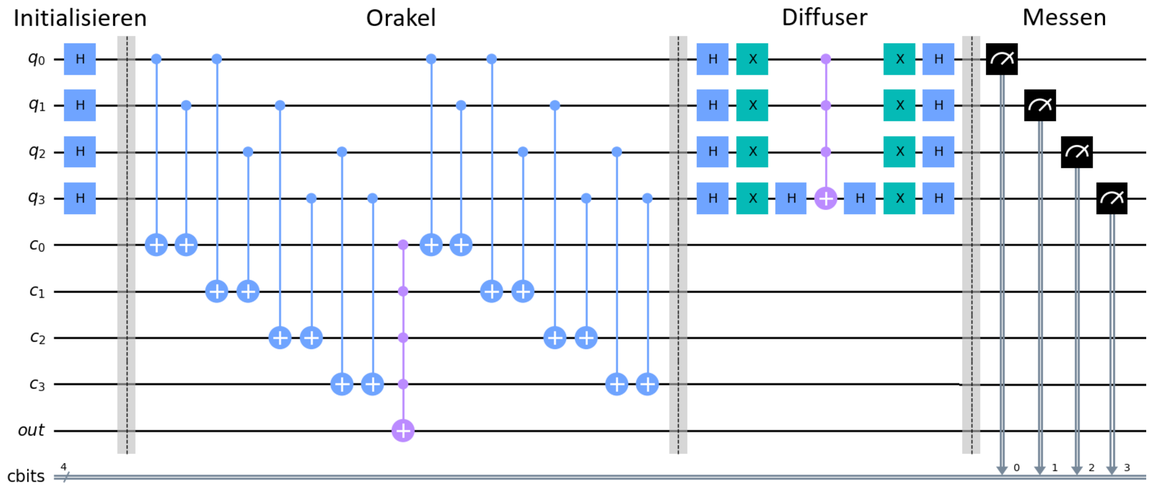

Abbildung 3: Quantenschaltkreis zur Lösung von 2x2 Sudokus

Fazit

Die Studierenden haben sich im Laufe des Projektes in die Materie des Quantencomputings eingearbeitet. Um das Wissen zu festigen wurden Quantenalgorithmen zur Lösung von Sudoku-Rätseln unter Verwendung des Grover-Algorithmus entwickelt. Die Erstellung von Quantenalgorithmen unterscheidet sich stark von der klassischen Programmierung, wodurch sich deren Kenntnisse nur geringfügig auf das Quantencomputing übertragen lassen. Zur Entwicklung von Quantenalgorithmen müssen zum jetzigen Zeitpunkt Quantengatter manuell erstellt werden, wodurch sich die Entwicklung von Quantenanwendungen komplex gestaltet. Zwar gibt es vielversprechende Ansätze um die Komplexität zu abstrahieren – für die Programmierung jedoch ist weiterhin ein Verständnis der physikalischen Grundlagen notwendig. Zudem ermöglicht es die aktuelle Quantenhardware noch nicht, realitätsnahe und komplexe Probleme zu lösen. Hierzu müssten Quantenrechner mehr Qubits und eine geringere Fehleranfälligkeit aufweisen.

Teilnehmer Jannik Bendfeld, Jannik Bergjan, Tizian Lengemann, Lennart Peters, Yannick Retzlaff, Sebastian Schmitt, Felix Schulze Sindern, Julian Tjaden, Marc Uphues

Literatur
[Qis23] IBM Qiskit Dokumentation https://qiskit.org/documentation/getting_started.html Abruf: 10.02.23
[Hom22] Homeister, Matthias; Quantum Computing verstehen: Grundlagen - Anwendungen – Perspektiven, 6. Auflage, Wiesbaden. Springer Fachmedien Wiesbaden; Springer Vieweg, 2022.
[FN19] Frisk Kockum, Anton & Nori, Franco; Quantum Bits with Josephson Junctions; 2019

Betreuung
Prof. Dr. Sebastian Thöne
Fachbereich Wirtschaft
Fachhochschule Münster