Simulating Multi-valued Grover Search in Cirq

Project Statement

Alexander J. Heilman & Andy Phillips

November 5, 2021

Overview

- The highly acclaimed Grover search algorithm is capable of 'searching' binary databases for a single solution

Overview

- The highly acclaimed Grover search algorithm is capable of 'searching' binary databases for a single solution
- Relatively new generalizations of Grover's search algorithm apply to multi-valued functions [2][3]

Overview

- The highly acclaimed Grover search algorithm is capable of 'searching' binary databases for a single solution
- Relatively new generalizations of Grover's search algorithm apply to multi-valued functions [2][3]
- Google's Cirq SDK allows simulation of qudit circuits

Overview

- The highly acclaimed Grover search algorithm is capable of 'searching' binary databases for a single solution
- Relatively new generalizations of Grover's search algorithm apply to multi-valued functions [2][3]
- Google's Cirq SDK allows simulation of qudit circuits
- New and emerging field of many-valued qudit algorithms offers lots of space to work on novel projects

Binary Grover Search (Simple Overview)

$$|+\rangle^{\otimes 3} = \begin{array}{c} |000\rangle \\ |001\rangle \\ |010\rangle \\ |011\rangle \\ |100\rangle \\ |101\rangle \\ |110\rangle \\ |111\rangle \end{array} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Binary Grover Search (Simple Overview)

$$|+\rangle^{\otimes 3} = \begin{array}{c} |000\rangle \\ |001\rangle \\ |010\rangle \\ |011\rangle \\ |100\rangle \\ |101\rangle \\ |110\rangle \\ |111\rangle \end{array} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \xrightarrow{\mathcal{O}} \begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Binary Grover Search (Simple Overview)

$$|+\rangle^{\otimes 3} = \begin{array}{c} |000\rangle \\ |001\rangle \\ |010\rangle \\ |011\rangle \\ |100\rangle \\ |101\rangle \\ |110\rangle \\ |111\rangle \end{array} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \xrightarrow{\mathcal{O}} \begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \xrightarrow[\text{search}]{\text{Grover}} |010\rangle$$

Binary Grover Search (Simple Overview)

$$|+\rangle^{\otimes 3} = \begin{array}{c} |000\rangle \\ |001\rangle \\ |010\rangle \\ |011\rangle \\ |100\rangle \\ |101\rangle \\ |110\rangle \\ |111\rangle \end{array} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \xrightarrow{\mathcal{O}} \begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \xrightarrow[\text{search}]{\text{Grover}} |010\rangle$$

Note that here we are omitting normalization constants and repeated iterations of the oracle and Grover diffusion operator

Natural Extension of Grover Search

- The binary Grover search algorithm is often touted as a 'database search'

Natural Extension of Grover Search

- The binary Grover search algorithm is often touted as a 'database search'
- Databases should be able to hold more than one type of value we care about

Natural Extension of Grover Search

- The binary Grover search algorithm is often touted as a 'database search'

- Databases should be able to hold more than one type of value we care about (Note that I'm not referring to multiple binary solutions, this is solved by amplitude amplification)

Natural Extension of Grover Search

- The binary Grover search algorithm is often touted as a 'database search'
- Databases should be able to hold more than one type of value we care about (Note that I'm not referring to multiple binary solutions, this is solved by amplitude amplification)
- A natural generalization then is to encode several different 'types of solutions' in our 'database' and search for one

Natural Extension of Grover Search

- The binary Grover search algorithm is often touted as a 'database search'
- Databases should be able to hold more than one type of value we care about (Note that I'm not referring to multiple binary solutions, this is solved by amplitude amplification)
- A natural generalization then is to encode several different 'types of solutions' in our 'database' and search for one
- These different 'types' can be naturally encoded by the dth roots of unity as opposed to just 1 and -1

Multi-Valued Grover Search

The typical Grover search algorithm is effectively used to find a set of basis states marked in a complete superposition of basis states by a relative amplitude of -1.

Multi-Valued Grover Search

The typical Grover search algorithm is effectively used to find a set of basis states marked in a complete superposition of basis states by a relative amplitude of -1.

$$|+\rangle^{\otimes n} \xrightarrow{\mathcal{O}} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{f(k)} |k\rangle$$

Multi-Valued Grover Search

The typical Grover search algorithm is effectively used to find a set of basis states marked in a complete superposition of basis states by a relative amplitude of -1.

$$|+\rangle^{\otimes n} \xrightarrow{\mathcal{O}} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{f(k)} |k\rangle$$

A generalization of this is to find basis states marked with one of many relative amplitudes, which if equally spaced are the roots of unity $\omega_d = e^{2\pi i/d}$

Multi-Valued Grover Search

The typical Grover search algorithm is effectively used to find a set of basis states marked in a complete superposition of basis states by a relative amplitude of -1.

$$|+\rangle^{\otimes n} \xrightarrow{\mathcal{O}} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{f(k)} |k\rangle$$

A generalization of this is to find basis states marked with one of many relative amplitudes, which if equally spaced are the roots of unity $\omega_d = e^{2\pi i/d}$

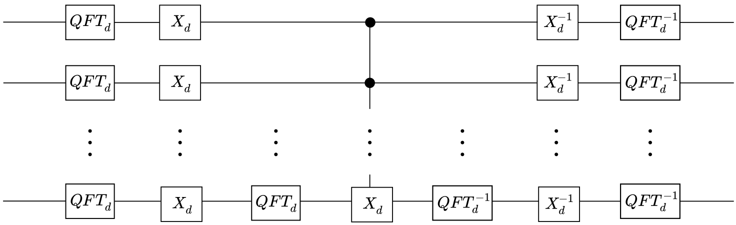
$$|+_d\rangle^{\otimes n} \xrightarrow{\mathcal{O}} \frac{1}{\sqrt{d^n}} \sum_{k=0}^{d^n-1} \omega_d^{f(k)} |k\rangle$$

Multi-Valued Grover Search

Maximilian Hunt and Samuel Hunt have recently published *Grovers Algorithm and Many-Valued Quantum Logic* (December 2020, [2]).

Multi-Valued Grover Search

Maximilian Hunt and Samuel Hunt have recently published *Grovers Algorithm and Many-Valued Quantum Logic* (December 2020, [2]). They generalize the Grover diffusion operator to qudits and multi-valued functions using the circuit below:



For gates see `https://alexheilman.com/qis/qudits`

Why Qudits?/What Qudits?

Qudits are generalization of qubits in that they're a quantum state of finite dimension d.

Why Qudits?/What Qudits?

Qudits are generalization of qubits in that they're a quantum state of finite dimension d.

$$|\psi_d\rangle = \sum_{n=0}^{d-1} c_n|n\rangle$$

Where qubits are just when $d = 2$.

Why Qudits?/What Qudits?

Qudits are generalization of qubits in that they're a quantum state of finite dimension d.

$$|\psi_d\rangle = \sum_{n=0}^{d-1} c_n |n\rangle$$

Where qubits are just when $d = 2$. They facilitate multi-valued logic as single qu$d$it gates then allow us to encapsulate dth order operations.

Why Qudits?/What Qudits?

Qudits are generalization of qubits in that they're a quantum state of finite dimension d.

$$|\psi_d\rangle = \sum_{n=0}^{d-1} c_n |n\rangle$$

Where qubits are just when $d = 2$. They facilitate multi-valued logic as single qu$d$it gates then allow us to encapsulate dth order operations.As an example, the generalized $\mathbf{Z}$ gate for $d = 3$

$$\mathbf{Z_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \omega & 0 \\ 0 & 0 & \omega^2 \end{bmatrix}$$

Generalized Grover Search (Simple Overview)

$$|+_3\rangle^{\otimes 2} = \begin{array}{c c} |00\rangle & \\ |01\rangle & \\ |02\rangle & \\ |10\rangle & \\ |11\rangle & \\ |12\rangle & \\ |20\rangle & \\ |21\rangle & \\ |22\rangle & \end{array} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Generalized Grover Search (Simple Overview)

$$|+_3\rangle^{\otimes 2} = \begin{array}{c} |00\rangle \\ |01\rangle \\ |02\rangle \\ |10\rangle \\ |11\rangle \\ |12\rangle \\ |20\rangle \\ |21\rangle \\ |22\rangle \end{array} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \xrightarrow{\mathcal{O}} \begin{bmatrix} 1 \\ \omega^2 \\ \omega^2 \\ \omega \\ \omega^2 \\ 1 \\ 1 \\ \omega^2 \\ 1 \end{bmatrix}$$

Generalized Grover Search (Simple Overview)

$$|+_3\rangle^{\otimes 2} = \begin{array}{c} |00\rangle \\ |01\rangle \\ |02\rangle \\ |10\rangle \\ |11\rangle \\ |12\rangle \\ |20\rangle \\ |21\rangle \\ |22\rangle \end{array} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \xrightarrow{\mathcal{O}} \begin{bmatrix} 1 \\ \omega^2 \\ \omega^2 \\ \omega \\ \omega^2 \\ 1 \\ 1 \\ \omega^2 \\ 1 \end{bmatrix} \xrightarrow[\text{search}]{\text{Generalized Grover}} |10\rangle$$

Generalized Grover Search (Simple Overview)

$$|+_3\rangle^{\otimes 2} = \begin{array}{c} |00\rangle \\ |01\rangle \\ |02\rangle \\ |10\rangle \\ |11\rangle \\ |12\rangle \\ |20\rangle \\ |21\rangle \\ |22\rangle \end{array} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \xrightarrow{\mathcal{O}} \begin{bmatrix} 1 \\ \omega^2 \\ \omega^2 \\ \omega \\ \omega^2 \\ 1 \\ 1 \\ \omega^2 \\ 1 \end{bmatrix} \xrightarrow[\text{search}]{\text{Generalized Grover}} |10\rangle$$

Again we omit normalization and gritty details

Simulation

- Google's Cirq allows us to simulate qudit circuits

Simulation

- Google's Cirq allows us to simulate qudit circuits
- Defined generalized qutrit (i.e. $d = 3$) gates

Simulation

- Google's Cirq allows us to simulate qudit circuits
- Defined generalized qutrit (i.e. $d = 3$) gates
- Figured out several simulation techniques in Cirq

Simulation

- Google's Cirq allows us to simulate qudit circuits
- Defined generalized qutrit (i.e. $d = 3$) gates
- Figured out several simulation techniques in Cirq
- Close to functioning simulator for two qutrits (i.e. $d = 3$ $n = 2$)

Simulation

- Google's Cirq allows us to simulate qudit circuits
- Defined generalized qutrit (i.e. $d = 3$) gates
- Figured out several simulation techniques in Cirq
- Close to functioning simulator for two qutrits (i.e. $d = 3$ $n = 2$), results next time

Simulation

- Google's Cirq allows us to simulate qudit circuits
- Defined generalized qutrit (i.e. $d = 3$) gates
- Figured out several simulation techniques in Cirq
- Close to functioning simulator for two qutrits (i.e. $d = 3$ $n = 2$), results next time (hopefully)

The repository can be found at
`https://github.com/tuc56407/QuditScripts`

```
---------BEGIN-----------
Circuit:
0 (d=3): ───[QFT]───F───[QFT]───[X]────────────o─────[X-]─────[QFT-]───M('q0')─────────────
                    │                          │
1 (d=3): ───[QFT]───F───[QFT]───[X]───[QFT]───[X]───[QFT-]───[X-]─────[QFT-]─────M('q1')───
-------------------------
measurements: q0=2 q1=2
output vector: (-0.143-0.99j)|22⟩
-------------------------
Results:
q0=0010220222220221111212212020020211021010121222112002022000221012112212102020220101122111211101222002
q1=0111220002221221212201202100021222020101022201010110122201222102212200212120220100022001202000022110
-------------------------
state at step 0: [0.333+0.j 0.333+0.j 0.333+0.j 0.333+0.j 0.333+0.j 0.333+0.j 0.333+0.j
 0.333+0.j 0.333+0.j]
state at step 1: [ 0.333+0.j     0.333+0.j     0.333+0.j    -0.167-0.289j -0.167-0.289j
 -0.167-0.289j -0.167-0.289j -0.167-0.289j -0.167+0.289j]
state at step 2: [ 0.   -0.385j  0.167-0.096j -0.167-0.096j  0.667+0.192j -0.167-0.096j
  0.   +0.192j  0.333+0.192j -0.   +0.192j  0.167-0.096j]
state at step 3: [ 0.167-0.096j  0.333+0.192j -0.   +0.192j -0.167-0.096j  0.   -0.385j
  0.167-0.096j  0.   +0.192j  0.667+0.192j -0.167-0.096j]
state at step 4: [ 0.289+0.167j  0.   +0.j     0.   -0.333j  0.   -0.333j  0.   -0.j
 -0.289+0.167j  0.289+0.167j -0.289+0.5j    0.   -0.333j]
state at step 5: [ 0.289+0.167j  0.   +0.j     0.   -0.333j -0.289+0.167j  0.   -0.333j
  0.   -0.j     0.289+0.167j -0.289+0.5j    0.   -0.333j]
state at step 6: [-0.167-0.096j -0.333+0.192j -0.   +0.192j  0.   +0.192j  0.667+0.192j
 -0.167-0.096j  0.167-0.096j  0.333+0.192j  0.   +0.192j]
state at step 7: [ 0.385+0.333j -0.096+0.167j  0.   +0.j    -0.481-0.167j -0.096+0.167j
  0.   +0.j    -0.481+0.167j  0.192-0.j    -0.289-0.167j]
state at step 8: [0.   +0.612j 0.354+0.612j 0.177+0.306j 0.   +0.j    0.   +0.j
 0.   +0.j    0.   +0.j    0.   +0.j    0.   +0.j   ]
state at step 9: [0. +0.j    0.5+0.866j 0. +0.j    0. +0.j    0. +0.j    0. +0.j
 0. +0.j    0. +0.j    0. +0.j   ]
```

Recap

- Grover search is for single binary entry in database

Recap

- Grover search is for single binary entry in database
- Multi-valued Grover search is recent development in field

Recap

- Grover search is for single binary entry in database
- Multi-valued Grover search is recent development in field
- Progress in simulating/testing new work in Cirq

What Hasn't Been Done?/Where to Go?

- The recent papers only generalize strict Grover search algorithm (i.e. oracles with only one relevant solution)

What Hasn't Been Done?/Where to Go?

- The recent papers only generalize strict Grover search algorithm (i.e. oracles with only one relevant solution)

- Natural to then generalize multi-valued amplitude amplification (Grover search for multiple solutions)

What Hasn't Been Done?/Where to Go?

- The recent papers only generalize strict Grover search algorithm (i.e. oracles with only one relevant solution)
- Natural to then generalize multi-valued amplitude amplification (Grover search for multiple solutions)
- With this it would be required to generalize quantum counting to the multi-valued paradigm

What Hasn't Been Done?/Where to Go?

- The recent papers only generalize strict Grover search algorithm (i.e. oracles with only one relevant solution)
- Natural to then generalize multi-valued amplitude amplification (Grover search for multiple solutions)
- With this it would be required to generalize quantum counting to the multi-valued paradigm
- These together would effectively give an algorithm for evaluating and counting roots of multivariate polynomial over finite fields (as they can be encoded in these types of quantum states[1])

References I

[1] Paul Appel, Alexander J Heilman, Ezekiel W Wertz, et al. "Finite-Function-Encoding Quantum States". In: *arXiv preprint arXiv:2012.00490* (2020).

[2] Samuel Hunt and Maximilien Gadouleau. "Grover's Algorithm and Many-Valued Quantum Logic". In: *arXiv preprint arXiv:2001.06316* (2020).

[3] Yale Fan. "Applications of multi-valued quantum algorithms". In: *arXiv preprint arXiv:0809.0932* (2008).

Links for work in progress
Code: `https://github.com/tuc56407/QuditScripts`
Overview: `https://alexheilman.com/qis/qudits`