Boolean functions in quantum computation

Ashley Montanaro

School of Mathematics,
University of Bristol

7 July 2017

`arXiv:1607.08473` and `arXiv:0810.2435`

Journal of Physics A, vol. 50, no. 8, 084002, 2017

Chicago Journal of Theoretical Computer Science 2010

University of Bristol

EPSRC
Engineering and Physical Sciences
Research Council

Quantum computing

A quantum computer is a machine designed to use quantum mechanics to outperform any "standard" computer based only on classical physics.

Quantum computing

A quantum computer is a machine designed to use quantum mechanics to outperform any "standard" computer based only on classical physics.

Known applications of quantum computers include:

- Simulation of quantum-mechanical systems;
- Integer factorisation, hence breaking the RSA cryptosystem;
- Unstructured search and optimisation;
- …

Quantum computing

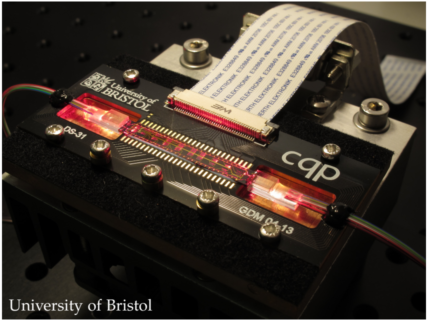
A quantum computer is a machine designed to use quantum mechanics to outperform any “standard” computer based only on classical physics.

Known applications of quantum computers include:

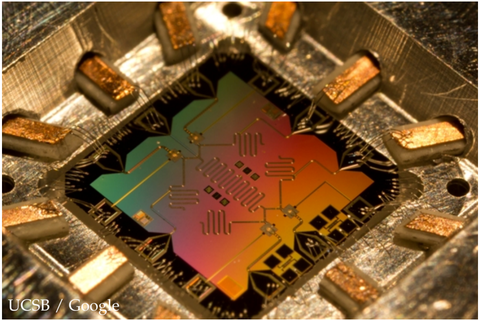
- Simulation of quantum-mechanical systems;
- Integer factorisation, hence breaking the RSA cryptosystem;
- Unstructured search and optimisation;
- …

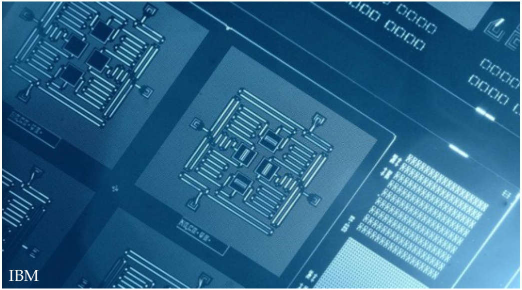
For many more, see the Quantum Algorithm Zoo (`math.nist.gov/quantum/zoo/`), which currently cites 361 papers on quantum algorithms…

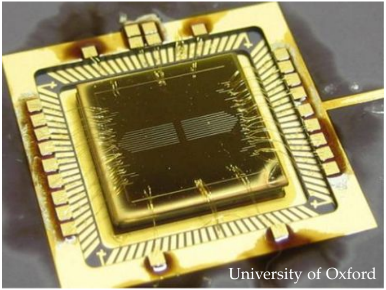
Quantum computers



University of Bristol


UCSB / Google


IBM


University of Oxford

This talk

In this talk I will discuss two connections between the theory of boolean functions and the theory of quantum computation:

This talk

In this talk I will discuss two connections between the theory of boolean functions and the theory of quantum computation:

- How low-degree polynomials over $\mathbb{F}_2$ can be used to understand quantum circuits;

This talk

In this talk I will discuss two connections between the theory of boolean functions and the theory of quantum computation:

- How low-degree polynomials over $\mathbb{F}_2$ can be used to understand quantum circuits;
- How quantum algorithms naturally give rise to a quantum generalisation of boolean functions.

This talk

In this talk I will discuss two connections between the theory of boolean functions and the theory of quantum computation:

- How low-degree polynomials over $\mathbb{F}_2$ can be used to understand quantum circuits;
- How quantum algorithms naturally give rise to a quantum generalisation of boolean functions.

A general principle

Although no large-scale general-purpose quantum computer has yet been built, quantum computation can already be used as a theoretical tool to study other areas of science and mathematics, without the need for an actual quantum computer.

Quantum computation

An exceptionally brief introduction:

- In a quantum algorithm, we start in some initial state, perform some quantum evolution, then measure and see some outcome (probabilistically).

Quantum computation

An exceptionally brief introduction:

- In a quantum algorithm, we start in some initial state, perform some quantum evolution, then measure and see some outcome (probabilistically).
- Associate each bit-string $x \in \{0, 1\}^n$ with an orthogonal basis vector in $\mathbb{C}^{2^n}$. This corresponds to a system of n qubits (quantum bits).

Quantum computation

An exceptionally brief introduction:

- In a quantum algorithm, we start in some initial state, perform some quantum evolution, then measure and see some outcome (probabilistically).
- Associate each bit-string $x \in \{0, 1\}^n$ with an orthogonal basis vector in $\mathbb{C}^{2^n}$. This corresponds to a system of n qubits (quantum bits).
- Then a quantum algorithm corresponds to a $2^n \times 2^n$ unitary matrix U, i.e. $UU^\dagger = I$.

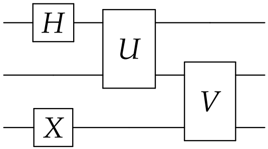
Quantum computation

An exceptionally brief introduction:

- In a quantum algorithm, we start in some initial state, perform some quantum evolution, then measure and see some outcome (probabilistically).
- Associate each bit-string $x \in \{0,1\}^n$ with an orthogonal basis vector in $\mathbb{C}^{2^n}$. This corresponds to a system of n qubits (quantum bits).
- Then a quantum algorithm corresponds to a $2^n \times 2^n$ unitary matrix U, i.e. $UU^\dagger = I$.
- If we apply U to a system initially in state $x \in \{0,1\}^n$ and then measure, the probability we see measurement outcome $y \in \{0,1\}^n$ is precisely $|U_{yx}|^2$.

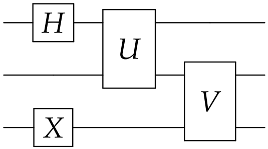
The quantum circuit model

Quantum algorithms are implemented as quantum circuits made up of elementary operations known as quantum gates.



The quantum circuit model

Quantum algorithms are implemented as quantum circuits made up of elementary operations known as quantum gates.

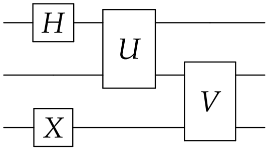


Each gate is a small unitary matrix itself, extended to acting on the whole space via the tensor (Kronecker) product with the identity matrix; e.g. the above circuit corresponds to the matrix

$$(I \otimes V)(U \otimes I)(H \otimes I \otimes X)$$

The quantum circuit model

Quantum algorithms are implemented as quantum circuits made up of elementary operations known as quantum gates.



Each gate is a small unitary matrix itself, extended to acting on the whole space via the tensor (Kronecker) product with the identity matrix; e.g. the above circuit corresponds to the matrix

$$(I \otimes V)(U \otimes I)(H \otimes I \otimes X)$$

Fundamental problem

For C in a given class of quantum circuits, compute $|C_{yx}|^2$.

Quantum circuits

The class of quantum circuits discussed today: those whose gates are picked from the set

$$\{H, Z, CZ, CCZ\}$$

where:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \quad \text{(aka “Hadamard”)}$$

Quantum circuits

The class of quantum circuits discussed today: those whose gates are picked from the set

$$\{H, Z, CZ, CCZ\}$$

where:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{(aka "Hadamard")}$$

$$Z = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}, \quad CZ = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix}, \quad CCZ = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & -1 \end{pmatrix}$$

and CCZ is an 8×8 matrix.

Quantum circuits

The class of quantum circuits discussed today: those whose gates are picked from the set

$$\{H, Z, CZ, CCZ\}$$

where:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{(aka ``Hadamard'')}$$

$$Z = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}, \quad CZ = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix}, \quad CCZ = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & -1 \end{pmatrix}$$

and CCZ is an 8×8 matrix.

Fact: This set of gates is universal for quantum computation.

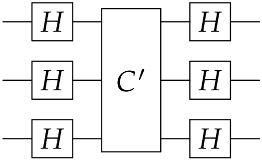
Understanding this class of circuits

We will show that, if C is picked from this class of circuits, the amplitudes C_{yx} of the corresponding unitary matrix can be written in a very concise form.

Understanding this class of circuits

We will show that, if C is picked from this class of circuits, the amplitudes C_{yx} of the corresponding unitary matrix can be written in a very concise form.

First assume that C begins and ends with a column of Hadamard gates:

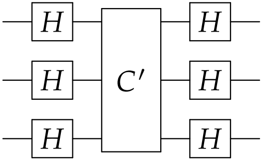


for some circuit C'.

Understanding this class of circuits

We will show that, if C is picked from this class of circuits, the amplitudes C_{yx} of the corresponding unitary matrix can be written in a very concise form.

First assume that C begins and ends with a column of Hadamard gates:

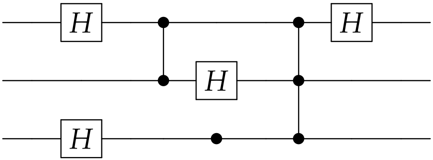


for some circuit C'.

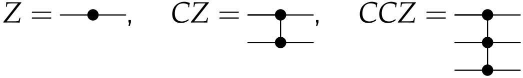
This is without loss of generality, as we can always add pairs of Hadamards to the beginning or end of each line ($H^2 = I$).

From a circuit to a polynomial

Now consider the internal part C', e.g.:



where we use the notation



From a circuit to a polynomial

Form a polynomial over $\mathbb{F}_2$ from the circuit as follows:

- Attach a variable to the left of each wire, and to the right of each Hadamard gate.

From a circuit to a polynomial

Form a polynomial over $\mathbb{F}_2$ from the circuit as follows:

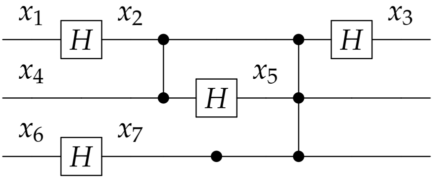
- Attach a variable to the left of each wire, and to the right of each Hadamard gate.
- Add a term multiplying together variables connected by a gate (of any kind).

From a circuit to a polynomial

Form a polynomial over $\mathbb{F}_2$ from the circuit as follows:

- Attach a variable to the left of each wire, and to the right of each Hadamard gate.
- Add a term multiplying together variables connected by a gate (of any kind).

For example:



corresponds to the polynomial

$$x_1x_2 + x_2x_3 + x_4x_5 + x_6x_7 + x_2x_4 + x_2x_5x_7 + x_7.$$

From a circuit to a polynomial

Assume C acts on ℓ qubits and contains h internal Hadamard gates.

Let f_C be the polynomial corresponding to C. Then f_C is a function of $n = h + \ell$ variables.

From a circuit to a polynomial

Assume C acts on ℓ qubits and contains h internal Hadamard gates.

Let f_C be the polynomial corresponding to C. Then f_C is a function of $n = h + \ell$ variables.

Write

$$\mathrm{gap}(f_C) := \sum_{x \in \{0,1\}^n} (-1)^{f_C(x)} = |\{x : f_C(x) = 0\}| - |\{x : f_C(x) = 1\}|.$$

From a circuit to a polynomial

Assume C acts on ℓ qubits and contains h internal Hadamard gates.

Let f_C be the polynomial corresponding to C. Then f_C is a function of $n = h + \ell$ variables.

Write

$$\mathrm{gap}(f_C) := \sum_{x \in \{0,1\}^n} (-1)^{f_C(x)} = |\{x : f_C(x) = 0\}| - |\{x : f_C(x) = 1\}|.$$

Claim

$$C_{0^n 0^n} = \frac{\mathrm{gap}(f_C)}{2^{h/2+\ell}}.$$

From a circuit to a polynomial

Assume C acts on ℓ qubits and contains h internal Hadamard gates.

Let f_C be the polynomial corresponding to C. Then f_C is a function of $n = h + \ell$ variables.

Write

$$\mathrm{gap}(f_C) := \sum_{x \in \{0,1\}^n} (-1)^{f_C(x)} = |\{x : f_C(x) = 0\}| - |\{x : f_C(x) = 1\}|.$$

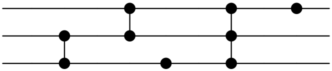
Claim

$$C_{0^n 0^n} = \frac{\mathrm{gap}(f_C)}{2^{h/2+\ell}}.$$

(All other amplitudes can be obtained too:
$C_{yx} = \mathrm{gap}(f_C + L_{x,y})/2^{h/2+\ell}$ for some linear function $L_{x,y}$.)

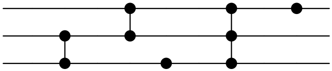
Proof idea

The special case where C' only contains Z, CZ, CCZ gates, e.g.:



Proof idea

The special case where C' only contains Z, CZ, CCZ gates, e.g.:

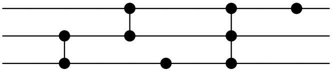


Let superscripts of Z, CZ, CCZ denote the qubits on which they act. Then, for any $x \in \{0,1\}^\ell$,

$$Z^i_{xx} = (-1)^{x_i}, \quad CZ^{ij}_{xx} = (-1)^{x_i x_j}, \quad CCZ^{ijk}_{xx} = (-1)^{x_i x_j x_k}.$$

Proof idea

The special case where C' only contains Z, CZ, CCZ gates, e.g.:



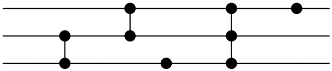
Let superscripts of Z, CZ, CCZ denote the qubits on which they act. Then, for any $x \in \{0,1\}^\ell$,

$$Z^i_{xx} = (-1)^{x_i}, \quad CZ^{ij}_{xx} = (-1)^{x_i x_j}, \quad CCZ^{ijk}_{xx} = (-1)^{x_i x_j x_k}.$$

As these gates are diagonal, we can obtain C'_{xx} by multiplying these expressions for different gates in C.

Proof idea

The special case where C' only contains Z, CZ, CCZ gates, e.g.:



Let superscripts of Z, CZ, CCZ denote the qubits on which they act. Then, for any $x \in \{0,1\}^\ell$,

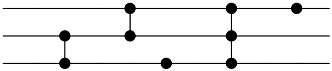
$$Z^i_{xx} = (-1)^{x_i}, \;\; CZ^{ij}_{xx} = (-1)^{x_i x_j}, \;\; CCZ^{ijk}_{xx} = (-1)^{x_i x_j x_k}.$$

As these gates are diagonal, we can obtain C'_{xx} by multiplying these expressions for different gates in C.

Each gate corresponds to a term in f_C as defined above. So $C'_{xx} = (-1)^{f_C(x)}$, and hence

Proof idea

The special case where C' only contains Z, CZ, CCZ gates, e.g.:



Let superscripts of Z, CZ, CCZ denote the qubits on which they act. Then, for any $x \in \{0,1\}^\ell$,

$$Z^i_{xx} = (-1)^{x_i}, \quad CZ^{ij}_{xx} = (-1)^{x_i x_j}, \quad CCZ^{ijk}_{xx} = (-1)^{x_i x_j x_k}.$$

As these gates are diagonal, we can obtain C'_{xx} by multiplying these expressions for different gates in C.

Each gate corresponds to a term in f_C as defined above. So $C'_{xx} = (-1)^{f_C(x)}$, and hence

$$(H^{\otimes \ell} C' H^{\otimes \ell})_{0^\ell 0^\ell} = \frac{1}{2^\ell} \sum_{x \in \{0,1\}^\ell} C'_{xx} = \frac{1}{2^\ell} \sum_{x \in \{0,1\}^\ell} (-1)^{f_C(x)} = \frac{\mathrm{gap}(f_C)}{2^\ell}.$$

Some easy observations

What can we say about this connection between circuits and polynomials?

Some easy observations

What can we say about this connection between circuits and polynomials?

- Every degree-3 polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ with no constant term has at least one corresponding quantum circuit, by considering the case where C' contains no Hadamard gates ...

Some easy observations

What can we say about this connection between circuits and polynomials?

- Every degree-3 polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ with no constant term has at least one corresponding quantum circuit, by considering the case where C' contains no Hadamard gates ...
- ... and this circuit is usually not unique, as e.g. $x_1 x_2$ can be obtained from either a CZ or Hadamard gate.

Some easy observations

What can we say about this connection between circuits and polynomials?

- Every degree-3 polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ with no constant term has at least one corresponding quantum circuit, by considering the case where C' contains no Hadamard gates ...
- ... and this circuit is usually not unique, as e.g. $x_1 x_2$ can be obtained from either a CZ or Hadamard gate.
- If f_C corresponds to a circuit C on ℓ qubits with h Hadamard gates, then

$$|\operatorname{gap}(f_C)| \leqslant 2^{h/2+\ell}$$

because $|C_{0^\ell 0^\ell}|^2 \leqslant 1$.

Consequences for simulating q. circuits

Quantum computers can be simulated by counting zeroes of degree-3 polynomials over $\mathbb{F}_2$!

Consequences for simulating q. circuits

Quantum computers can be simulated by counting zeroes of degree-3 polynomials over $\mathbb{F}_2$!

- …actually not so surprising: this problem was already known to be #P-complete [Ehrenfeucht and Karpinski '90].
- So this may not be a useful way of simulating general quantum circuits.

Consequences for simulating q. circuits

Quantum computers can be simulated by counting zeroes of degree-3 polynomials over $\mathbb{F}_2$!

- ...actually not so surprising: this problem was already known to be #P-complete [Ehrenfeucht and Karpinski '90].
- So this may not be a useful way of simulating general quantum circuits.

But we can simulate some special kinds of quantum circuits this way, e.g.:

- Those with no CCZ gates (as $\text{gap}(f)$ can be computed efficiently for degree-2 polynomials f) – this also follows from the Gottesman-Knill theorem.
- Those where there exists a transformation $L \in GL_n(\mathbb{F}_2)$ such that $f_C \circ L$ depends on only $O(\log n)$ variables.

A quantum version of boolean functions?

How can we generalise the concept of a boolean function $f : \{0, 1\}^n \to \{0, 1\}$ in a "quantum" way?

A quantum version of boolean functions?

How can we generalise the concept of a boolean function $f : \{0,1\}^n \to \{0,1\}$ in a "quantum" way?

First, change to considering $f : \{0,1\}^n \to \{\pm 1\}$.

A quantum version of boolean functions?

How can we generalise the concept of a boolean function $f : \{0,1\}^n \to \{0,1\}$ in a "quantum" way?

First, change to considering $f : \{0,1\}^n \to \{\pm 1\}$.

Then write such a function as a diagonal matrix, e.g. for $n = 2$:

$$\begin{pmatrix} f(00) & & & \\ & f(01) & & \\ & & f(10) & \\ & & & f(11) \end{pmatrix}$$

A quantum version of boolean functions?

How can we generalise the concept of a boolean function $f : \{0,1\}^n \to \{0,1\}$ in a "quantum" way?

First, change to considering $f : \{0,1\}^n \to \{\pm 1\}$.

Then write such a function as a diagonal matrix, e.g. for $n = 2$:

$$\begin{pmatrix} f(00) & & & \\ & f(01) & & \\ & & f(10) & \\ & & & f(11) \end{pmatrix}$$

This naturally suggests a generalisation:

Definition

A quantum boolean function is a $2^n \times 2^n$ unitary matrix whose eigenvalues are in the set $\{\pm 1\}$.

A quantum version of boolean functions

Is this a nontrivial definition?

A quantum version of boolean functions

Is this a nontrivial definition?

We can obtain quantum boolean functions from:

- Standard methods for implementing functions $f : \{0,1\}^n \to \{\pm 1\}$ on a quantum computer;
- Quantum algorithms solving decision problems;
- Quantum error-correcting codes;
- …in fact, any subspace of $\mathbb{C}^{2^n}$.

A quantum version of boolean functions

Is this a nontrivial definition?

We can obtain quantum boolean functions from:

- Standard methods for implementing functions $f : \{0,1\}^n \to \{\pm 1\}$ on a quantum computer;
- Quantum algorithms solving decision problems;
- Quantum error-correcting codes;
- ... in fact, any subspace of $\mathbb{C}^{2^n}$.

For any quantum boolean function F, $UFU^\dagger$ is also a quantum boolean function for any unitary matrix U.

A quantum version of boolean functions

We can also generalise some techniques used in the analysis of classical boolean functions, e.g. the Fourier (aka Walsh-Hadamard) transform.

A quantum version of boolean functions

We can also generalise some techniques used in the analysis of classical boolean functions, e.g. the Fourier (aka Walsh-Hadamard) transform.

If f is a quantum boolean function, then we can write

$$f = \sum_{s \in \{I,X,Y,Z\}^n} \hat{f}(s)\, \chi_s$$

where

$$\chi_s = s_1 \otimes s_2 \otimes \cdots \otimes s_n$$

and I, X, Y, Z are the Pauli matrices

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \;\; X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \;\; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \;\; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

A quantum version of boolean functions

We can also generalise some techniques used in the analysis of classical boolean functions, e.g. the Fourier (aka Walsh-Hadamard) transform.

If f is a quantum boolean function, then we can write

$$f = \sum_{s \in \{I,X,Y,Z\}^n} \hat{f}(s)\, \chi_s$$

where

$$\chi_s = s_1 \otimes s_2 \otimes \cdots \otimes s_n$$

and I, X, Y, Z are the Pauli matrices

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \;\; X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \;\; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \;\; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

(Classical boolean functions are the same, but only use I & Z.)

A quantum version of boolean functions

Is this an interesting definition?

A quantum version of boolean functions

Is this an interesting definition?

Using the Pauli expansion, we can prove some analogous results to those in the theory of classical boolean functions, e.g.:

- Testing linearity (= being a Pauli matrix);
- Learning an unknown quantum boolean function;
- The Friedgut-Kalai-Naor (FKN) theorem ("dictator-vs-constant").

A quantum version of boolean functions

Is this an interesting definition?

Using the Pauli expansion, we can prove some analogous results to those in the theory of classical boolean functions, e.g.:

- Testing linearity (= being a Pauli matrix);
- Learning an unknown quantum boolean function;
- The Friedgut-Kalai-Naor (FKN) theorem ("dictator-vs-constant").

…but many "combinatorial" results are harder to prove (e.g. because there are uncountably many quantum boolean functions!).

Conclusions and open problems

There are already some intriguing connections between the theory of boolean functions and the theory of quantum computation, with many more yet to be explored.

Conclusions and open problems

There are already some intriguing connections between the theory of boolean functions and the theory of quantum computation, with many more yet to be explored.

Some open problems:

- Can we use the connection between quantum circuits and degree-3 polynomials to find new simulation techniques for quantum algorithms?

Conclusions and open problems

There are already some intriguing connections between the theory of boolean functions and the theory of quantum computation, with many more yet to be explored.

Some open problems:

- Can we use the connection between quantum circuits and degree-3 polynomials to find new simulation techniques for quantum algorithms?
- Do interesting classes of polynomials correspond to interesting quantum circuits?

Conclusions and open problems

There are already some intriguing connections between the theory of boolean functions and the theory of quantum computation, with many more yet to be explored.

Some open problems:

- Can we use the connection between quantum circuits and degree-3 polynomials to find new simulation techniques for quantum algorithms?
- Do interesting classes of polynomials correspond to interesting quantum circuits?
- Can we prove a quantum analogue of the KKL theorem ("every boolean function has an influential variable")?

Conclusions and open problems

There are already some intriguing connections between the theory of boolean functions and the theory of quantum computation, with many more yet to be explored.

Some open problems:

- Can we use the connection between quantum circuits and degree-3 polynomials to find new simulation techniques for quantum algorithms?
- Do interesting classes of polynomials correspond to interesting quantum circuits?
- Can we prove a quantum analogue of the KKL theorem ("every boolean function has an influential variable")?

Thanks!