ACCQ 206 - Simon’s algorithm, Quantum Fourier transform and Shor’s algorithm

Alex Bredariol Grilo
Alex.Bredariol-Grilo@lip6.fr
Quantum operations

Evolution of quantum states is described by unitary operators

- \( UU^\dagger = U^\dagger U = I \)
  - For every quantum state \( |\psi\rangle \), \( U |\psi\rangle \) is also a quantum state
  - Reversible: no information loss

- Equivalent models of quantum computation:
  - Quantum Turing Machines
  - Quantum circuits
  - Adiabatic quantum computation
  - Measurement-based quantum computation
  - ...
Universal gateset

Definition

$\varepsilon$-approximation An $n$-qubit unitary $U$ $\varepsilon$-approximates an $n$-qubit unitary $U'$ if

$$\max_{|\psi\rangle\in\mathbb{C}^{2^n}} ||U|\psi\rangle - U'|\psi\rangle|| \leq \varepsilon.$$ 

Definition

Universal gateset A gateset $G$ is universal if for every unitary $U$, there exists a unitary $U'$ composed by gates in $G$ such that $U'$ $\varepsilon$-approximates $U$.

Lemma

The following gatesets are universal:

- $\{1$-qubit gates, CNOT\}
- $\{\text{CNOT}, H, T\}$
- $\{H, \text{CCNOT}\}$ (for unitaries with real entries)
Quantum circuits

\[ \mathcal{Q} = \{ g_1, \ldots, g_n \} \]
Deutsch-Josza algorithm (1992)

Problem

Given oracle access to \( f : \{0, 1\}^n \rightarrow \{0, 1\} \) with the promise that:
- \( f \) is constant
- \( f \) is balanced

Find out which is the case.

Quantum parallelism: \( |\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle \)
Deutsch-Josza algorithm (1992)

Problem
Given oracle access to $f : \{0, 1\}^n \to \{0, 1\}$ with the promise that:
- $f$ is constant
- $f$ is balanced
Find out which is the case.

Quantum parallelism:

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0, 1\}^n} |x\rangle |f(x)\rangle$$

$$H^n |0 \cdots 0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0, 1\}^n} (-1)^{f(x)} |x\rangle$$

$$(H \circ H)^2 = \Sigma$$
Simon’s algorithm (1994)

Problem

Given oracle access to a function $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ such that:

$$\exists s \neq 0^n \text{ such that } f(x) = f(y) \text{ iff } y \in \{x, s \oplus x\}.$$ 

Find $s \in \{0, 1\}^n$. 

```plaintext
0...0
s → 1(0...0)
0...1 → 1(0...1)
s00...0 → ...
```

$2^n / 2$ images
Simon’s algorithm

Problem

Given oracle access to a function $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ such that:

$$\exists s \neq 0^n \text{ such that } f(x) = f(y) \text{ iff } y \in \{x, s \oplus y\}.$$ 

Find $s$.

Randomized algorithms: $\begin{array}{l}
\exists i, j \text{ s.t. } f(i) = 1/i_j \\
\text{have hand on query complexity to find } s \text{ is } \Theta(\sqrt{n})
\end{array}$
Simon’s algorithm

Lemma
With a single quantum query, we can compute a random $d \in \{0, 1\}^n$ such that

$$d \cdot s = 0.$$ 

Theorem
There is a quantum algorithm that retrieves $s$ with high probability with $O(n)$ queries.

Proof

\[
\begin{align*}
\text{linearly independent } d_1, \ldots, d_n \quad \text{such that} \quad d_1 \cdot s &= 0, \\
&\quad \quad \vdots \\
&\quad \quad d_n \cdot s = 0 \\
\Rightarrow \quad \text{gives you } s
\end{align*}
\]

with $p = \frac{i^n}{n}$.

\[\sum d_i s_i \equiv 0 \mod(2) \Rightarrow 0\]
Simon’s algorithm

Lemma
With a single quantum query, we can compute a random \( d \in \{0, 1\}^n \) such that

\[ d \cdot s = 0. \]

Theorem
There is a quantum algorithm that retrieves \( s \) with high probability with \( O(n) \) queries.

Proof
When we sample \( d_1, d_2, \ldots, d_{i-1} \) is LI wr. \( 1 - \frac{2i}{2^n} \)
\( d_1, \ldots, d_{i-1} \) always \( \in \text{span}(d_1, \ldots, d_{i-1}) \).
Simon’s algorithm - sampling $d$ s.t. $d \cdot s = 0$

\[ |0\rangle \rightarrow^n H \otimes n \rightarrow^n U_f \rightarrow^n H \otimes n \rightarrow^3 d \]

\[ |0\rangle \rightarrow^n H \otimes n \rightarrow^n U_f \rightarrow^n H \otimes n \rightarrow^3 d \]

**Analysis**

1. $H^0 |0, 0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0, 1\}^n} |x\rangle$

2. $U_f \left( \frac{1}{\sqrt{2^n}} \sum_{x} |x\rangle |0\rangle \right) \rightarrow \frac{1}{\sqrt{2^n}} \sum_{x} |x\rangle |f(x)\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0, 1\}^n} \left( f_1^{-1}(y) + f_2^{-1}(y) \right) |y\rangle$

\[ f_1^{-1}(y) = x_0 \quad f_2^{-1}(y) = x_2 \]
Simon’s algorithm

Analysis (cont.)
Simon’s algorithm

Analysis (cont.)
Simon’s algorithm

Analysis (cont.)

If we measure the second register, the outcome is a random \( y \in \text{Im}(f) \), and the post-measurement state is

\[
\frac{1}{|f|} \left( |f^{-1}(y)\rangle + |f^{-1}(y)\rangle \otimes |y\rangle \right)
\]

3) How
Simon’s algorithm

Analysis (cont.)

\[ H^\otimes n \left( \frac{1}{\sqrt{d}} |2\rangle + \frac{1}{\sqrt{d}} |2 \oplus s\rangle \right) = \frac{1}{\sqrt{d}} \left( H^\otimes n |2\rangle + \frac{1}{\sqrt{d}} H^\otimes n |2 \oplus s\rangle \right) = \]

\[ = \frac{1}{\sqrt{d}} \sum_p \left( -1 \right)^{d \cdot s} |p\rangle = \frac{1}{\sqrt{d}} \sum_p \left( -1 \right)^{d \cdot s} \prod_{i=1}^{d} \left( 1 + (-1)^{d \cdot s} \right) |p\rangle \]

\[ \text{if } d \cdot s = 1 \]
\[ (4) = (-1)^{d \cdot 2} (1 + (-1)) = 0 \]
\[ \text{if } d \cdot s = 0 \]
\[ (4) = (-1)^{d \cdot 2} (1 + 1) = 2(-1)^{d \cdot 2} \]
Simon’s algorithm

Analysis (cont.)

\[
\left\{ \frac{1}{\sqrt{d}} \sum_{d': d \cdot d' = 0} 2(-1)^{d \cdot d'} |d\rangle \right\} = \frac{1}{\sqrt{d}} \sum_{d', s = 0} (-1)^{d \cdot d'} |d\rangle
\]

state after step 3

4) After measuring the measurement \( d: d' s = 0 \) is the outcome of

\[
\left( \frac{(-1)^{d \cdot d'}}{\sqrt{2^{n-1}}} \right)^2 = \frac{1}{2^{n-1}}
\]
Simon’s algorithm

Analysis (cont.)
Simon’s algorithm - recap

Problem

Given oracle access to a function \( f : \{0,1\}^n \rightarrow \{0,1\}^n \) such that:

\[ \exists s \neq 0^n \text{ such that } f(x) = f(y) \text{ iff } y \in \{x, s \oplus y\}. \]

Find \( s \).

The following circuit samples random \( d \) such that

\[ \forall 1 \leq i \leq n, d_i \cdot s = 0. \]
Simon’s algorithm - applications

Authentication schemes that are secure classically are broken in the quantum query model using Simon’s algorithm.

Mahadev. Classical Verification of Quantum Computations. FOCS 2018
Breakthrough result on delegation of quantum computation by classical clients, it uses a variant of Simon’s algorithm to make the server perform an unknown measurement.
Factoring

Problem

For a composite number $N$, find a non-trivial factor of $N$.

Classical algorithms:

$$d(\log N)^{\alpha}$$

- if heuristic, $\alpha = 1/3$
- if worst case, $\alpha = 1/2 \implies 2^{\Omega(\log \log N)}$

Belief: factoring is hard

So most crypto constructions are based on hardness of
Shor’s algorithm (1994) - overview

$\text{poly}(\log N)$ - time quantum alg. for factoring $N$

1. "classical part" as reduction from factoring to period finding
2. "quantum part" to period finding w/ $O$ Fourier transform
Definition
A function $f : \mathbb{F} \to \mathbb{F}$ is periodic if $\exists r$ such that $f(a) = f(b)$ iff $b = a + kr \pmod{p}$.

Example
For every $a, N$, $f_a(x) = a^x \pmod{N}$ is periodic.

- $f_a(0) = 1 \pmod{N}$
- $f_a(1) = a \pmod{N}$
- $f_a(i) = a^i \pmod{N}$
- $f_a(i + kr) = a^i \pmod{N}$ for any $y \geq 1$, $z \in \mathbb{N}$ and $f_a(z) \neq f_a(y)$.
Shor’s algorithm - from period finding to factoring

**Algorithm 1: Factoring from period finding**

Pick an unif. random \( a \in \{2, \ldots, N\} \);

\[
\text{if } \gcd(a, N) > 1 \text{ then}
\]

\[\text{return } \gcd(a, N) ;\]

\[
\text{end}
\]

Find the period \( r \) of \( f_a(x) = a^x \pmod{N} \);

\[
\text{if } r \text{ is odd or } a^{r/2} = \pm 1 \pmod{N} \text{ then}
\]

\[\text{return } \bot ;\]

\[
\text{else}
\]

\[\text{return } \max \left( \gcd(N, a^{r/2} + 1), \gcd(N, a^{r/2} - 1) \right) ;\]

\[
\text{end}
\]

\( \text{we already have a non-trivial factor of } N \) (we got lucky!)
Shor's algorithm - from period finding to factoring

Algorithm 1: Factoring from period finding

Pick an unif. random \( a \in \{2, \ldots, N\} \);

\[
\text{if } \gcd(x, N) > 1 \text{ then}
\]
\[
\quad \text{return } \gcd(a, N) ;
\]
\[
\text{end}
\]

Find the period \( r \) of \( f_a(x) = a^x \pmod{N} \);

\[
\text{if } r \text{ is odd or } a^{r/2} = \pm 1 \pmod{N} \text{ then}
\]
\[
\quad \text{return } \perp ;
\]
\[
\text{else}
\]
\[
\quad \text{return } \max(\gcd(N, a^{r/2} + 1), \gcd(N, a^{r/2} - 1)) ;
\]
\[
\text{end}
\]
Fourier Transform

Original signal

\[ f(t) = \cos(2\pi(3t))e^{-\pi t^2} \]

Fourier transform

\[ \hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i x \xi} \, dx \]
Discrete Fourier Transform

\[ x_1, \ldots, x_N \in \mathbb{C} \quad \overset{\text{DFT}}{\longrightarrow} \quad y_1, \ldots, y_N \in \mathbb{C} \]

\[ y_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{-2\pi i jk/N} \]

\[ F_N = \frac{1}{\sqrt{N}} \begin{pmatrix}
1 & 1 & \cdots & 1 \\
1 & \omega & \cdots & \omega^{i-1} \\
1 & \omega^{i-1} & \cdots & \omega^{i(N-1)} \\
1 & \omega^{N-1} & \cdots & \omega^{(N-1)^2} \\
\end{pmatrix} \begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_N \\
\end{pmatrix} = \begin{pmatrix}
y_1 \\
y_2 \\
\vdots \\
y_N \\
\end{pmatrix} \]

Naive DFT computation: \( O(N^2) \) on time \( O(N \log N) \)
Quantum Fourier Transform

\[
\begin{bmatrix}
\alpha_0 \\
\vdots \\
\alpha_{N-1}
\end{bmatrix}
\xrightarrow{\text{QFT}}
\begin{bmatrix}
\beta_0 \\
\vdots \\
\beta_{N-1}
\end{bmatrix}
\]

\[
\sum_j \beta_j |j\rangle
\]

\[
\beta_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \alpha_k e^{-2\pi i j k N}
\]

We can compute QFT using \(O((\log N)^4)\) "simple" gates.
Shor's algorithm - quantum algorithm for period finding

Pick $q = 2^d$ such that $N^2 < q < 2N^2$.

$|0\rangle^\otimes d \xrightarrow{QFT_q} U_{q^x} \text{(mod } N) \xrightarrow{QFT_q^{-1}} |z\rangle$ + classical post-processing

$x \in \{0,1\}^n \iff y \in [2^n]$ to implement

1) $QFT |0\rangle = \frac{1}{\sqrt{q}} \sum_{j=0}^{q-1} e^{2\pi ij/q} |j\rangle = H^{\otimes d} |0\rangle$

2) $U_{f_{\alpha}} \left( \frac{1}{\sqrt{q}} \sum_{j=0}^{q-1} |j\rangle |0\rangle^\otimes n \right) \rightarrow \frac{1}{\sqrt{q}} \sum_{j=0}^{q-1} |j\rangle |0^\alpha \text{ (mod } N)\rangle$
Shor's algorithm - quantum algorithm for period finding

Pick \( q = 2^d \) such that \( N^2 < q < 2N^2 \).

\[
\begin{align*}
|0\rangle \otimes d & \quad \xrightarrow{QFT_q} \quad |0\rangle \otimes d \quad \xrightarrow{U_a^x \text{ (mod } N)} \quad |z\rangle \quad \xrightarrow{QFT_q^{-1}} \quad |y\rangle + \text{ classical post-processing} \\
|0\rangle \otimes n & \quad \xrightarrow{\text{measure the 2nd register}} \quad \text{if outcome is } y \quad \frac{1}{\sqrt{q-1}} \sum_{j=0}^{q-1} |j\rangle |a^j \text{ (mod } N)\rangle
\end{align*}
\]
Shor’s algorithm - quantum algorithm for period finding

Pick $q = 2^d$ such that $N^2 < q < 2N^2$.

$$|0\rangle \otimes d \xrightarrow{QFT_q} U_{a^x} \pmod{N} \xrightarrow{QFT_q^{-1}} \sum_{z}\binom{y}{z} + \text{classical post-processing}$$

$$QFT^{-1}(\frac{1}{\sqrt{m}} \sum_{k} |k\rangle |y\rangle) = \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} \frac{1}{\sqrt{q}} \sum_{l=0}^{q-1} e^{\frac{2\pi i (kr + 5s)l}{q}} |l\rangle$$

$$= \frac{1}{\sqrt{mq}} \sum_{l=0}^{q-1} e^{2\pi i s/q} \left( \sum_{k=0}^{m-1} e^{2\pi i r/k} \right) |l\rangle$$

for different $s$'s
Shor’s algorithm - quantum algorithm for period finding

Pick $q = 2^d$ such that $N^2 < q < 2N^2$.

$$|0\rangle^\otimes d \xrightarrow{QFT_q} U_{a^x} \xrightarrow{} QFT_q^{-1} \xrightarrow{} z \xrightarrow{} + \text{classical post-processing}$$

$$|0\rangle^\otimes n$$

$$(\ast) \sum_{k=0}^{m-1} (e^{\frac{\pi i n k}{q}})^{x^k} \text{ for different values of } x$$

$$(\ast \ast) = \sum_{k=0}^{n-1} (\frac{m}{k}) = \frac{m!}{k!(m-k)!}$$

$$(\ast \ast \ast) = \frac{1 - e^{\frac{2\pi i n x}{q}}}{1 - e^{\frac{2\pi i}{q}}}$$
Shor's algorithm - quantum algorithm for period finding

Pick \( q = 2^d \) such that \( N^2 < q < 2N^2 \).

\[
\ket{0} \otimes d \xrightarrow{QFT_q} U_{a^x} \text{ (mod } N) \xrightarrow{QFT_q^{-1}} \zeta \]

\[
\ket{0} \otimes n \xrightarrow{QFT_q} U_{a^x} \text{ (mod } N) \xrightarrow{QFT_q^{-1}} \zeta
\]

Assume that \( r \) divide \( q \):

\[
e^{i\pi r/2q} = 1 \iff r = 0, 9, 18 \text{ in multiples of } 9
\]

\( \frac{r}{q} \) is an integer, \( \zeta \) amplitude on output state is

\[
\frac{m}{m^3} \quad \text{pads of memory such } \frac{m^3}{m^3} = \frac{1}{9}
\]

+ classical post-processing
Shor’s algorithm - quantum algorithm for period finding

Pick $q = 2^d$ such that $N^2 < q < 2N^2$.

$$|0\rangle^\otimes d \xrightarrow{QFT_q} U_{a^x} \text{ (mod } N) \xrightarrow{QFT_q^{-1}} z$$

$$|0\rangle^\otimes n \xrightarrow{y}$$

+ classical post-processing

Bottom line: in the "simple" case

$$\frac{1}{\log \log N}$$

Outcome of measurement of parameter of algo.

$$\sqrt{2} = \frac{c}{\sqrt{r}}$$

Period that we want to find

$$\frac{2}{9} = \frac{c}{\sqrt{r}}$$

$\log \log N$
Factoring in quantum polynomial time
Unstructured search

Computational problem

Let us assume that a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ has the following property:

$$\exists \text{ a unique } x^* \text{ such that } f(x^*) = 1.$$ 

Find $x^*$.

Classical algorithms:

$d^n$ inputs of $f$
Grover search - amplitude amplification
Grover search - amplitude amplification

Phase inversion

\[ |x^*\rangle \rightarrow \frac{1}{\sqrt{2^n-1}} \sum_{y \neq x^*} |y\rangle \]
Grover search - amplitude amplification

Phase inversion

\[ |\phi\rangle = \frac{1}{\sqrt{2^n-1}} \sum_{y \neq x^*} |y\rangle \]
Grover search - amplitude amplification

Phase inversion

\[ |x^*\rangle \]

\[ |\phi\rangle \rightarrow \frac{1}{\sqrt{2^n-1}} \sum_{y \neq x^*} |y\rangle \]

\[ U_f |\phi\rangle \]

\[ U_f |i\rangle = (-1)^{f(i)} |i\rangle \]
Grover search - amplitude amplification

Phase inversion

\[ U_f \ket{\phi} = \frac{1}{\sqrt{2^n-1}} \sum_{y \neq x^*} \ket{y} \]

\[ U_f \ket{i} = (-1)^{f(i)} \ket{i} \]

Inversion about the mean
Grover search - amplitude amplification

Phase inversion

\[ |x^*\rangle \quad \Rightarrow \quad \frac{1}{\sqrt{2^n-1}} \sum_{y \neq x^*} |y\rangle \]

\[ U_f |\phi\rangle \]

\[ U_f |i\rangle = (-1)^{f(i)} |i\rangle \]

Inversion about the mean

\[ |x^*\rangle \quad \Rightarrow \quad |\phi\rangle \]
Grover search - amplitude amplification

Phase inversion

\[ |x^*\rangle \]

\[ \frac{1}{\sqrt{2^n}} \sum_{y \neq x^*} |y\rangle \]

\[ U_f |\phi\rangle \]

\[ U_f |i\rangle = (-1)^{f(i)} |i\rangle \]

Inversion about the mean

\[ |x^*\rangle \]

\[ |\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{y} |y\rangle \]

\[ |\phi\rangle \]
Grover search - amplitude amplification

Phase inversion

\[ |x^*\rangle \]

\[ |\phi\rangle \]

\[ U_f |\phi\rangle \]

\[ U_f |i\rangle = (-1)^{f(i)} |i\rangle \]

Inversion about the mean

\[ |x^*\rangle \]

\[ D |\phi\rangle \]

\[ |\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_y |y\rangle \]

\[ D |i\rangle = H^\otimes n (I - 2 |0\rangle \langle 0|) H^\otimes n |i\rangle \]
Grover search - amplitude amplification

Phase inversion

\[ |\psi\rangle = \frac{1}{\sqrt{2^n-1}} \sum_{y \neq x^*} |y\rangle \]

\[ U_f |\phi\rangle \]

**Grover’s algorithm**

Start from

\[ \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \]

and repeat these two procedures \( \sqrt{2^n} \) times.
Grover search and amplitude amplification

- The algorithm can be generalized for different settings: more than one marked element, amplification of success probability, quantum rewinding in quantum cryptographic protocols
- Quadratic speedup over classical algorithm
- Not so drastic, but it solves a generic problem
- It can be reframed into different frameworks: quantum walks, block encoding and Hamiltonian simulations, ...
Recent (somewhat) advances in quantum algorithms

- Quantum linear algebra
  - Linear system of equations
  - Semi-definite programming
  - Applications to Quantum Machine Learning: faster recommendation systems, ...

- Quantum learning theory
  - Learning theory: ML from the perspective of complexity theory
  - Exponential separation between some quantum and classical models
  - Connections to cryptography and circuit lower bounds

- Quantum chemistry
  - Using quantum computers to simulate quantum systems

- NISQ: Near-term intermediate scale quantum computers
  - What can we do with current Google, IBM, D-Wave devices?

- Quantum random walks
- Quantum Fourier sampling
- Block encodings