Classical Simulation

CSE 490Q: Quantum Computation
Stabilizer States and Circuits

\[ I \otimes X \]

\[ I \otimes Z \]

\[ \text{use the fact that} \]

\[ \text{Diagram of operations} \]
Stabilizer States and Circuits

$I \otimes X = \begin{array}{c}
\text{CNOT} \\
\text{H}
\end{array}$

$I \otimes Z = \begin{array}{c}
\text{Z}
\end{array}$

use the fact that

$\begin{array}{c}
\text{H} \\
\text{H}
\end{array} = \begin{array}{c}
\text{H} \\
\text{H}
\end{array}$
Stabilizer States and Circuits

\[ H \, Z \, H = X \quad \text{or equiv.} \quad H \, Z = X \, H \]
Stabilizer States and Circuits

\[ H \cdot H = X \text{ or equiv. } Z \cdot H = H \cdot X \]
The Hidden Subgroup Problem

CSE 490Q: Quantum Computation
Shor’s (1994) breakthrough result was an efficient q. algorithm for \textit{factoring}.
- strongly believed to be classically hard
  - 200+ years of failed attempts to solve it
  - hardness is \textit{assumed} by cryptosystems like RSA
- not a toy problem, not a black box

- Fastest classical algorithm runs in \(O(2^{n/3})\) time
- Shor’s algorithm is \(O(n^3)\)
Post-Shor Algorithms

• Much later work focused on generalizing it
  • we will look at some of the generalizations (more than one!)
  • Shor’s paper still has insights that may not be fully understood still

• First generalization was the phase estimation algorithm
Alexei Kitaev, in Russia at the time, heard about Shor’s result. Unable to get the paper, he re-derived how he thought it must work. His result was actually a significant generalization of Shor’s.

The technique he invented is called “phase estimation”

(He also generalized the problem solved.)
Groups
Definition: A group $G$ is a set with...

- a special element $e \in G$
- an operation taking any $x, y \in G$ to some $xy \in G$
- an operation taking any $x \in G$ to some $x^{-1} \in G$

satisfying certain rules...

- $ex = x = xe$ for all $x \in G$ (“identity”)
- $x x^{-1} = e = x^{-1} x = e$ for all $x \in G$ (“inverses”)
- $(xy)z = x(yz)$ for all $x, y, z \in G$ (“associativity”)
public interface Group {
    Collection<Element> elements();
    Element identity();
}

public interface Element {
    Element multiply(Element y); // returns xy
    Element inverse(); // returns x^{-1}
}

// Specification would list the rules from the prior page...
Definition: A group $G$ is a set with...

- a special element $e \in G$
- an operation taking any $x, y \in G$ to some $xy \in G$
- an operation taking any $x \in G$ to some $x^{-1} \in G$

satisfying the following rules...

Definition: A group $G$ is abelian if $xy = yx$ for all $x, y \in G$.

- (In this case, we often write $x+y$ instead of $xy$.)
• \( \mathbb{R} \) is a group under multiplication, where 1 is identity and the inverse of \( x \) is \( 1/x \).

• \( \mathbb{Z} \) is not a group under multiplication because, e.g., 2 has no inverse.

• \( \mathbb{Z} \) is a group under addition, where 0 is the identity and the inverse of \( x \) is \( -x \).
  • \( x + (-x) = 0 = (-x) + x \)

• \( \mathbb{Z}_n \) is a group under addition, where multiplication is modulo \( N \).
  • Set is \{0, 1, ..., N-1\}
  • 0 is still the identity
  • the inverse of \( x \) is \( N - x \) since \( x + (N - x) = N \equiv 0 \) (mod \( N \))
\( \mathbb{Z}_N \) is a group under addition, where multiplication is modulo \( N \).
- Set is \( \{0, 1, \ldots, N-1\} \)
- 0 is still the identity
- If \( \gcd(x, N) = 1 \), then \( x \) has a multiplicative inverse \( x^{-1} \).
  - Bezout’s Theorem says there exist \( a \) and \( b \) such that \( ax + bN = 1 \), which says that \( ax \equiv 1 \pmod{N} \)

\( (\mathbb{Z}_N)^x \) is a group under multiplication.
- Set is \( \{ x \in \mathbb{Z}_N : \gcd(x, N) = 1 \} \)
- 1 is the identity
Those groups are all abelian. Here are some non-abelian groups.

- 2 x 2 unitary matrices form a group under multiplication
  - the identity is I
  - the inverse of U is $U^\dagger$

- The Pauli group is a (finite!) group under multiplication
  - elements are one of \{I, X, Y, Z\} scaled by one of \{1, -1, i, -i\}
  - inverses... homework problem
If $G$ and $H$ are groups, we can combine them to form a new group.

**Definition:** $G \times H$ is the group with elements \{ $(x, y) : x \in G$ and $y \in H$ \}

- identity is $(e_G, e_H)$, where $e_G$ is the identity of $G$ and $e_H$ of $H$
- multiplication is $(u, v)(x, y) = (ux, vy)$
- inverse is $(x, y)^{-1} = (x^{-1}, y^{-1})$

- Can take more products to build vectors.
- E.g., $G^4 = G \times G \times G \times G$, whose elements are of the form $(w, x, y, z)$
Finite abelian groups are actually not that complicated...

**Theorem:** If $G$ is a finite abelian group, then there are integers $N_1, \ldots, N_k \in \mathbb{Z}$ such that $G = \mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_k}$.

- Here "=" is not really equal but rather "isomorphism".
- There is a correspondence between elements of $G$ and $\mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_k}$
  - for every $g \in G$, there is an $f(g) \in \mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_k}$ that behaves identically
- Can do the arithmetic in $\mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_k}$ instead of $G$:
  - $f^{-1}(f(g)f(h)) = gh$ for all $g, h \in G$
This theorem gives us a great understanding of abelian groups.

- any abelian group is just a vector of numbers using modular arithmetic
- if we can solve a problem for vectors and modular arithmetic, then in principle, we can solve that problem for all abelian groups (hint hint)

- Non-abelian groups can have much complex behavior...
• Element are $r^i s^j$ for some $i \in \mathbb{Z}_N$ and $j \in \mathbb{Z}_2$, where $r^N = 1$ and $s^2 = 1$
  • identity is $(1, 1) = (r^0, s^0)$
  • $r$ and $s$ satisfy $sr = r^{-1} s$

• Correspondence with $\mathbb{Z}_N \times \mathbb{Z}_2$, but multiplication is more complicated...
  • $(r^i s)(r^j s^k) = r^i r^{-j} s^k = r^{i+j} s^k$
  • c.f. $(r^i)(r^j s^k) = r^{i+j} s^k$

• The inverse of $r^i$ is $r^{N-i}$
• The inverse if $r^i s$ is $r^i s !!$
• 3 x 3 matrices with entries from $\mathbb{Z}_p$, with p prime

\[
\begin{bmatrix}
1 & a & c \\
0 & 1 & b \\
0 & 0 & 1 \\
\end{bmatrix}
\]

• Can check that a product of these is also of this form.
• Can check that this is the inverse:

\[
\begin{bmatrix}
1 & -a & -c + ab \\
0 & 1 & -b \\
0 & 0 & 1 \\
\end{bmatrix}
\]

• (Can generalize all of this to k x k matrices for k > 3.)
Subgroups
A subgroup is a subset of a group that is itself a group.

In particular, $S \subseteq G$ is a subgroup iff

- $e \in S$
- if $x \in S$, then $x^{-1} \in S$
- if $x \in S$ and $y \in S$, then $xy \in S$

So you can perform group operations on elements of $S$ and you will never see an element outside of $S$ (i.e., from $G - S$).
If $S \subseteq G$ is a subgroup, then a **coset** of $S$ is a set of the form

$$gS = \{ gs : s \in S \}$$

for some $g \in G$.

- This is not a subgroup (e.g., we do not have $e \in gS$).
- It is just a subset of $G$. 
Cosets

- Let $S \subseteq G$ be a subgroup.

**Theorem:** If $g, h \in G$, then we either have $gS = hS$ or $gS \cap hS = \emptyset$.

**Proof:** If $gS \cap hS$ is not empty, then there is some element in both sets that are the same. I.e., we have $gx = hy$ for some $x, y \in S$.

Multiplying on the right by $x^{-1}$, we have $g = hyx^{-1}$.

Hence, for any $s \in S$, we have $gs = hyx^{-1}s \in hS$ since $x, y, s \in S$.

This shows that $gS \subseteq hS$. Analogous argument shows $hS \subseteq gS$, so $gS = hS$. Q.E.D.
**Theorem**: If \( g, h \in G \), then we either have \( gS = hS \) or \( gS \cap hS = \emptyset \).

- So \( gS \) and \( hS \) are either the same set or completely disjoint.
- And every element in \( G \) is in some coset of \( S \) (e.g., \( g \in gS \) since it is \( g = ge \)).
- Hence, the cosets of \( S \) partition \( G \). I.e., we can write

\[
G = g_1S \cup g_1S \cup \ldots \cup g_kS
\]

for some choice of \( g_1, \ldots, g_k \in G \).
Hidden Subgroup Problem (HSP)
Problem: Given a function $f : G \rightarrow \{0,1\}^k$ (via an oracle) that is constant on cosets of some subgroup $H$, find the subgroup $H$.

• I.e., we are promised that $f(gh_1) = f(gh_2)$ for all $h_1, h_2 \in H$ and $g \in G$.
• Equivalently, $f(g) = f(gh)$ for all $h \in H$ and $g \in G$.

• Outputs need not have any meaning.

• To “find” the subgroup, we usually ask for a ”generating set” for it
  • set of elements such that any $h \in H$ can be written as a product of those
  • (usually done when we have informationally “determined” $H$)
**Problem:** Given a function $f : G \rightarrow \{0,1\}^k$ (via an oracle) that is constant on cosets of some subgroup $H$, find the subgroup $H$.

- Requires exponential time classically.

- E.g., consider $G = \mathbb{Z}_2^n = \mathbb{Z}_2 \times \ldots \times \mathbb{Z}_2$

- Promised that $f(x_1 + y_1, \ldots, x_n + y_n) = f(x_1, \ldots, x_n)$ for some vector $(y_1, \ldots, y_n)$ and all vectors $(x_1, \ldots, x_n)$

- But there are $2^n$ choices for what $(y_1, \ldots, y_n)$ could be.

- We’ll have to try all of them in the worst case to find the one that works.
**Problem:** Given a function $f : G \rightarrow \{0,1\}^k$ (via an oracle) that is constant on cosets of some subgroup $H$, find the subgroup $H$.

- Requires exponential time classically.

- E.g., consider $G = (\mathbb{Z}_N)^x$ and $f(x) = y^x \pmod{N}$

- If $y$ has period $r$, then $f(x+r) = y^{x+r} = y^x y^r = y^x = f(x)$.

- So $f$ is constant on cosets of $\mathbb{Z}_r$

- This is the period finding problem, so we have generalized Shor’s problem.
**Problem:** Given a function \( f : G \to \{0,1\}^k \) (via an oracle) that is constant on cosets of some subgroup \( H \), find the subgroup \( H \).

- Requires exponential time classically.

- We will see an efficient quantum algorithm for a large number of cases.
  - handle all abelian groups using on phase estimation

- (This is essentially what that Kitaev did after hearing about Shor’s result.)