ATPL 2024: Hybrid Quantum-Classical Programming.

Lecture 5: Clifford Algebra, Stabilizers, and separating Classical Computations from Quantum

James Avery (avery@diku.dk)

December 2, 2024

Outline

Table of Contents

Overview

Recall from Lecture 1:

Gottesman-Knill Theorem

Theorem: A quantum circuit using only the following elements can be simulated efficiently on a classical computer:

1. Preparation of qubits in computational basis states,
2. Clifford gates (exactly), and
3. Measurements in the computational basis (stochastically).

and

Extension of Gottesman-Knill by Aaronson and Gottesman (2004)

A general n-qubit quantum circuit comprised of M Clifford gates and T non-Clifford gates can be simulated in space $\mathcal{O}\left(n^2 + n^2 2^{\gamma T}\right)$ and time $\mathcal{O}\left(n^2 + nM + nT\,2^{\gamma T}\right)$.

Overview

Recall from Lecture 1:

Gottesman-Knill Theorem

Theorem: A quantum circuit using only the following elements can be simulated efficiently on a classical computer:

1. Preparation of qubits in computational basis states,
2. Clifford gates (exactly), and
3. Measurements in the computational basis (stochastically).

and

Extension of Gottesman-Knill by Aaronson and Gottesman (2004)

A general n-qubit quantum circuit comprised of M Clifford gates and T non-Clifford gates can be simulated in space $\mathcal{O}\left(n^2 + n^2 2^{\gamma T}\right)$ and time $\mathcal{O}\left(n^2 + nM + nT\,2^{\gamma T}\right)$.

How do we actually do this?

Overview

Recall from Lecture 1:

Gottesman-Knill Theorem

Theorem: A quantum circuit using only the following elements can be simulated efficiently on a classical computer:

1. Preparation of qubits in computational basis states,
2. Clifford gates (exactly), and
3. Measurements in the computational basis (stochastically).

and

Extension of Gottesman-Knill by Aaronson and Gottesman (2004)

A general n-qubit quantum circuit comprised of M Clifford gates and T non-Clifford gates can be simulated in space $\mathcal{O}\left(n^2 + n^2 2^{\gamma T}\right)$ and time $\mathcal{O}\left(n^2 + nM + nT\,2^{\gamma T}\right)$.

How do we actually do this? Can we exploit this for hybrid QC?

Goal

Given a quantum program that is a mixture of Clifford and non-Clifford operations, we want to

1. Exploit commutation/rewrite rules to propagate the Clifford operations to the front of the program,
2. Compute them efficiently on a classical computer,
3. Minimize the resulting state,
4. Input to quantum device, and
5. Perform the non-Clifford operations on the quantum device.

Table of Contents

The Pauli Spin Matrices

The Pauli Spin-matrices

$$\mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

generate the Special Unitary group $SU(2)$: Any 3D rotation around an axis $\mathbf{n}$ can be neatly written:

$$\mathbf{R_n}(\theta) = e^{i\frac{\theta}{2}(n_x X + n_y Y + n_z Z)}$$

The Pauli Spin Matrices

The Pauli Spin-matrices

$$\mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

generate the Special Unitary group $SU(2)$: Any 3D rotation around an axis $\mathbf{n}$ can be neatly written:

$$\mathbf{R_n}(\theta) = e^{i\frac{\theta}{2}(n_x X + n_y Y + n_z Z)}$$

$\boldsymbol{\sigma} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ and $\mathbf{I}$ generate the full group of unitaries $U(2)$:

Any 2×2 unitary $\mathbf{U} \in U(2)$ can be written as

$$\mathbf{U} = e^{i(\phi I + s_x \mathbf{X} + s_y \mathbf{Y} + s_z \mathbf{Z})} = e^{i\phi} e^{i\mathbf{s}\cdot\boldsymbol{\sigma}}$$

for some $(\phi, s_x, s_y, s_z) \in \mathbb{R}^4$.

The Pauli Spin Matrices

The Pauli Spin-matrices

$$\mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

generate the Special Unitary group $SU(2)$: Any 3D rotation around an axis $\mathbf{n}$ can be neatly written:

$$\mathbf{R_n}(\theta) = e^{i\frac{\theta}{2}(n_x X + n_y Y + n_z Z)}$$

$\boldsymbol{\sigma} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ and $\mathbf{I}$ generate the full group of unitaries $U(2)$:

Any 2×2 unitary $\mathbf{U} \in U(2)$ can be written as

$$\mathbf{U} = e^{i(\phi I + s_x \mathbf{X} + s_y \mathbf{Y} + s_z \mathbf{Z})} = e^{i\phi} e^{i\mathbf{s}\cdot\boldsymbol{\sigma}}$$

for some $(\phi, s_x, s_y, s_z) \in \mathbb{R}^4$.

This implies: Any 1-qubit quantum program is of this form! Adding CNOT, this generates all quantum programs.

The Pauli Group $\mathcal{P}_n$

The Pauli matrices form a *group*

$$\mathcal{P}_1 = \left\{ i^l \sigma \middle| \sigma \in \{\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}, l \in \{0, 1, 2, 3\} \right\}$$

under matrix multiplication if we include identity $\mathbf{I}$ and a phase i^k:

The Pauli Group $\mathcal{P}_n$

The Pauli matrices form a *group*

$$\mathcal{P}_1 = \left\{ i^l \sigma \middle| \sigma \in \{\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}, l \in \{0,1,2,3\} \right\}$$

under matrix multiplication if we include identity **I** and a phase i^k:

Multiplication table

	I	X	Y	Z
I	I	X	Y	Z
X	X	I	iZ	$-iY$
Y	Y	$-iZ$	I	iX
Z	Z	iY	$-iX$	I

The Pauli Group $\mathcal{P}_n$

The Pauli matrices form a *group*

$$\mathcal{P}_1 = \left\{ i^l \sigma \middle| \sigma \in \{\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}, l \in \{0,1,2,3\} \right\}$$

under matrix multiplication if we include identity **I** and a phase i^k:

Multiplication table

	I	X	Y	Z
I	I	X	Y	Z
X	X	I	iZ	$-iY$
Y	Y	$-iZ$	I	iX
Z	Z	iY	$-iX$	I

This extends to the n qubit Pauli group:

$$\mathcal{P}_n = \left\{ i^l \bigotimes_{k=0}^{n-1} \sigma_k \middle| \sigma_k \in \{\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}, l \in \{0,1,2,3\} \right\}$$

Note that there is only one phase i^l, out front, due to multilinearity of $\otimes$.

Table of Contents

Heisenberg representation of states

Just like there are different ways of representing linear operators (e.g. matrix, bra/ket, tensor-network), there are different ways to represent states (vectors in a Hilbert space):

- Coefficients $\mathbf{a} \in \mathbb{C}^n$ in some basis $(\psi = \sum_k a_k \phi_k)$,

Heisenberg representation of states

Just like there are different ways of representing linear operators (e.g. matrix, bra/ket, tensor-network), there are different ways to represent states (vectors in a Hilbert space):

- Coefficients $\mathbf{a} \in \mathbb{C}^n$ in some basis ($\psi = \sum_k a_k \phi_k$),
- Bra $\langle\psi|$ or ket $|\psi\rangle$ (defined by inner products $\langle\psi|\psi'\rangle$),

Heisenberg representation of states

Just like there are different ways of representing linear operators (e.g. matrix, bra/ket, tensor-network), there are different ways to represent states (vectors in a Hilbert space):

- Coefficients $\mathbf{a} \in \mathbb{C}^n$ in some basis ($\psi = \sum_k a_k \phi_k$),
- Bra $\langle\psi|$ or ket $|\psi\rangle$ (defined by inner products $\langle\psi|\psi'\rangle$),
- Today: **Symmetry!** (Defined xby stabilizer group)

Heisenberg representation of states

Just like there are different ways of representing linear operators (e.g. matrix, bra/ket, tensor-network), there are different ways to represent states (vectors in a Hilbert space):

- Coefficients $\mathbf{a} \in \mathbb{C}^n$ in some basis ($\psi = \sum_k a_k \phi_k$),
- Bra $\langle\psi|$ or ket $|\psi\rangle$ (defined by inner products $\langle\psi|\psi'\rangle$),
- Today: **Symmetry!** (Defined xby stabilizer group)

This is the key insight that allows us to compute many quantum programs quickly on a classical computer (and is also the foundation of much of quantum error correction theory).

Example: A single Pauli spin-symmetry uniquely determines a 1-qubit state

Because of normalization $\langle\psi|\psi\rangle = 1$, one linear equation is enough to uniquely determine a 1-qubit state $|\psi\rangle$:

$$
\begin{aligned}
|\psi\rangle = X|\psi\rangle \iff \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_0 \end{bmatrix} \\
\iff a_0 &= a_1 \\
\iff |\psi\rangle &\simeq \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}
\end{aligned}
$$

Example: A single Pauli spin-symmetry uniquely determines a 1-qubit state

Because of normalization $\langle\psi|\psi\rangle = 1$, one linear equation is enough to uniquely determine a 1-qubit state $|\psi\rangle$:

$$
\begin{aligned}
|\psi\rangle = -X|\psi\rangle \iff & \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = -\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = -\begin{bmatrix} a_1 \\ a_0 \end{bmatrix} \\
\iff & a_0 = -a_1 \\
\iff & |\psi\rangle \simeq \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}
\end{aligned}
$$

Example: A single Pauli spin-symmetry uniquely determines a 1-qubit state

Because of normalization $\langle\psi|\psi\rangle = 1$, one linear equation is enough to uniquely determine a 1-qubit state $|\psi\rangle$:

$$
\begin{aligned}
|\psi\rangle = Y\,|\psi\rangle \iff \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} &= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = i \begin{bmatrix} -a_1 \\ a_0 \end{bmatrix} \\
\iff a_0 = -i\,a_1 &\iff a_1 = i\,a_0 \\
\iff |\psi\rangle &\simeq \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}
\end{aligned}
$$

Example: A single Pauli spin-symmetry uniquely determines a 1-qubit state

Because of normalization $\langle\psi|\psi\rangle = 1$, one linear equation is enough to uniquely determine a 1-qubit state $|\psi\rangle$:

$$\begin{aligned} |\psi\rangle = -Y\,|\psi\rangle \iff \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} &= -\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = i \begin{bmatrix} a_1 \\ -a_0 \end{bmatrix} \\ &\iff a_0 = i\,a_1 \\ &\iff |\psi\rangle \simeq \frac{1}{\sqrt{2}} \begin{bmatrix} i \\ 1 \end{bmatrix} \end{aligned}$$

Example: A single Pauli spin-symmetry uniquely determines a 1-qubit state

Because of normalization $\langle\psi|\psi\rangle = 1$, one linear equation is enough to uniquely determine a 1-qubit state $|\psi\rangle$:

$$
\begin{aligned}
|\psi\rangle = Z\,|\psi\rangle \iff & \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_0 \\ -a_1 \end{bmatrix} \\
\iff & a_1 = -a_1 \\
\iff & a_1 = 0 \\
\iff & |\psi\rangle \simeq \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\end{aligned}
$$

Example: A single Pauli spin-symmetry uniquely determines a 1-qubit state

Because of normalization $\langle\psi|\psi\rangle = 1$, one linear equation is enough to uniquely determine a 1-qubit state $|\psi\rangle$:

$$
\begin{aligned}
|\psi\rangle = -Z\,|\psi\rangle \iff & \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} -a_0 \\ a_1 \end{bmatrix} \\
\iff & a_0 = -a_0 \\
\iff & a_0 = 0 \\
\iff & |\psi\rangle \simeq \begin{bmatrix} 0 \\ 1 \end{bmatrix}
\end{aligned}
$$

Example: Stabilizer representation of 2-qubit states

$$|\psi\rangle = X \otimes Z\,|\psi\rangle \iff \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \right) \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix}$$

$$\iff \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = \begin{bmatrix} a_{10} \\ -a_{11} \\ a_{00} \\ -a_{01} \end{bmatrix}$$

$$\iff a_{00} = a_{10} \wedge a_{01} = -a_{11}$$

One stabilizer group element determines 2 out of 4 constraints necessary to fully determine a 2-qubit state!

Stabilizer representation of 2-qubit states

$$|\psi\rangle = Z \otimes Y\,|\psi\rangle \iff \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \otimes \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \right) \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix}$$

$$\iff \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = \begin{bmatrix} 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 0 & i \\ 0 & 0 & -i & 0 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = \begin{bmatrix} -i\,a_{01} \\ i\,a_{00} \\ i\,a_{11} \\ -i\,a_{10} \end{bmatrix}$$

$$\iff a_{01} = i\,a_{00} \wedge a_{10} = i\,a_{11}$$

Stabilizer representation of 2-qubit states

So, a state that is stabilized by $X \otimes Z$ and $Z \otimes Y$ satisfies

$$\left.\begin{array}{l} |\psi\rangle = X \otimes Z\,|\psi\rangle : a_{00} = a_{10} \wedge a_{01} = -a_{11} \\ |\psi\rangle = Z \otimes Y\,|\psi\rangle : a_{01} = i\,a_{00} \wedge a_{10} = i\,a_{11} \end{array}\right\} \Longleftrightarrow \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = \begin{bmatrix} 1 \\ \\ \\ \\ \end{bmatrix}$$

Stabilizer representation of 2-qubit states

So, a state that is stabilized by $X \otimes Z$ and $Z \otimes Y$ satisfies

$$\left.\begin{array}{l} |\psi\rangle = X \otimes Z\,|\psi\rangle : \boldsymbol{a_{00} = a_{10}} \wedge a_{01} = -a_{11} \\ |\psi\rangle = Z \otimes Y\,|\psi\rangle : a_{01} = i\,a_{00} \wedge a_{10} = i\,a_{11} \end{array}\right\} \iff \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = \begin{bmatrix} 1 \\ \\ 1 \\ \\ \end{bmatrix}$$

Stabilizer representation of 2-qubit states

So, a state that is stabilized by $X \otimes Z$ and $Z \otimes Y$ satisfies

$$\left.\begin{array}{l} |\psi\rangle = X \otimes Z\,|\psi\rangle : a_{00} = a_{10} \wedge a_{01} = -a_{11} \\ |\psi\rangle = Z \otimes Y\,|\psi\rangle : \boldsymbol{a_{01} = i\, a_{00}} \wedge a_{10} = i\, a_{11} \end{array}\right\} \iff \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = \begin{bmatrix} 1 \\ i \\ 1 \\ \end{bmatrix}$$

Stabilizer representation of 2-qubit states

So, a state that is stabilized by $X \otimes Z$ and $Z \otimes Y$ satisfies

$$\left.\begin{array}{l} |\psi\rangle = X \otimes Z\,|\psi\rangle : a_{00} = a_{10} \wedge \boldsymbol{a_{01} = -a_{11}} \\ |\psi\rangle = Z \otimes Y\,|\psi\rangle : a_{01} = i\,a_{00} \wedge a_{10} = i\,a_{11} \end{array}\right\} \iff \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = \begin{bmatrix} 1 \\ i \\ 1 \\ -i \end{bmatrix}$$

Stabilizer representation of 2-qubit states

So, a state that is stabilized by $X \otimes Z$ and $Z \otimes Y$ satisfies

$$\left.\begin{array}{l} |\psi\rangle = X \otimes Z\,|\psi\rangle : a_{00} = a_{10} \wedge a_{01} = -a_{11} \\ |\psi\rangle = Z \otimes Y\,|\psi\rangle : a_{01} = i\,a_{00} \wedge a_{10} = i\,a_{11} \end{array}\right\} \iff \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ i \\ 1 \\ -i \end{bmatrix}$$

Stabilizer representation of 2-qubit states

So, a state that is stabilized by $X \otimes Z$ and $Z \otimes Y$ satisfies

$$\left.\begin{array}{l} |\psi\rangle = X \otimes Z\,|\psi\rangle : a_{00} = a_{10} \wedge a_{01} = -a_{11} \\ |\psi\rangle = Z \otimes Y\,|\psi\rangle : a_{01} = i\,a_{00} \wedge a_{10} = i\,a_{11} \end{array}\right\} \iff \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ i \\ 1 \\ -i \end{bmatrix}$$

or, written out in th the computational basis:

$$|\psi\rangle = \frac{1}{2}\left(|00\rangle + i\,|01\rangle + |10\rangle - i\,|11\rangle\right)$$

Table of Contents

Stabilizer group and generators

Definition

The **stabilizer group** of a quantum state $|\psi\rangle$ is the subgroup $\mathcal{S}(\psi) \subseteq \mathcal{P}_n$ of the Pauli group that stabilizes $|\psi\rangle$:

$$\mathcal{S}(\psi) = \{P \in \mathcal{P}_n | P|\psi\rangle = |\psi\rangle\}$$

Stabilizer group and generators

Definition

The **stabilizer group** of a quantum state $|\psi\rangle$ is the subgroup $\mathcal{S}(\psi) \subseteq \mathcal{P}_n$ of the Pauli group that stabilizes $|\psi\rangle$:

$$\mathcal{S}(\psi) = \{P \in \mathcal{P}_n | P|\psi\rangle = |\psi\rangle\}$$

Properties

1. $P^2 = I$ for every $P \in \mathcal{S}(\psi)$
   ...because $P^2 = i^l I$ by construction, so
   $i^l|\psi\rangle = P^2|\psi\rangle = |\psi\rangle \implies i^l = 1.$
2. $\mathcal{S}(\psi)$ is *Abelian* (all members commute)
   ...because either $PQ = QP$ or $PQ = -QP$, and the latter would imply $|\psi\rangle = PQ|\psi\rangle = -QP|\psi\rangle = -|\psi\rangle$.

An $\mathcal{O}\left(2n^2\right)$ Representation of Stabilizer States

We call a state $|\psi\rangle$ that is *uniquely determined* by its stabilizer group $\mathcal{S}\left(\psi\right)$ a **stabilizer state**.

Theorem 1 (García et al. (2014) p.5 rephrased)

For any n-qubit state $|\psi\rangle$

1. $S(|\psi\rangle) \simeq \mathbb{Z}_2^k$ for some $k \leq n$. ($\leftarrow$ bits!)
2. $|\psi\rangle$ is a stabilizer state if and only if $k = n$.

Proof

An $\mathcal{O}\left(2n^2\right)$ Representation of Stabilizer States

We call a state $|\psi\rangle$ that is *uniquely determined* by its stabilizer group $\mathcal{S}\left(\psi\right)$ a **stabilizer state**.

Theorem 1 (García et al. (2014) p.5 rephrased)

For any n-qubit state $|\psi\rangle$

1. $S(|\psi\rangle) \simeq \mathbb{Z}_2^k$ for some $k \leq n$. ($\leftarrow$ bits!)
2. $|\psi\rangle$ is a stabilizer state if and only if $k = n$.

Proof

1. By the Fundamental Theorem of Finite Abelian Groups, any Abelian group with $P^2 = I$ is isomorphic to $\mathbb{Z}_2^k$ for some k.

An $\mathcal{O}\left(2n^2\right)$ Representation of Stabilizer States

We call a state $|\psi\rangle$ that is *uniquely determined* by its stabilizer group $\mathcal{S}\left(\psi\right)$ a **stabilizer state**.

Theorem 1 (García et al. (2014) p.5 rephrased)

For any n-qubit state $|\psi\rangle$

1. $S(|\psi\rangle) \simeq \mathbb{Z}_2^k$ for some $k \leq n$. $(\leftarrow$ bits!$)$
2. $|\psi\rangle$ is a stabilizer state if and only if $k = n$.

Proof

1. By the Fundamental Theorem of Finite Abelian Groups, any Abelian group with $P^2 = I$ is isomorphic to $\mathbb{Z}_2^k$ for some k.
2. . . . next slide.

An $\mathcal{O}\left(2n^2\right)$ Representation of Stabilizer States

1. Every $P \in \mathcal{S}\left(|\psi\rangle\right)$ is a tensor product $P = \pm P_1 \otimes \cdots \otimes P_n$ of n 2×2 Pauli matrices, which each have eigenvalues 1 and -1.

An $\mathcal{O}\left(2n^2\right)$ Representation of Stabilizer States

1. Every $P \in \mathcal{S}\left(|\psi\rangle\right)$ is a tensor product $P = \pm P_1 \otimes \cdots \otimes P_n$ of n 2×2 Pauli matrices, which each have eigenvalues 1 and -1.
2. So P also has eigenvalues $1, -1$, each with 2^{n-1}-dimensiaonal eigenspaces...

An $\mathcal{O}\left(2n^2\right)$ Representation of Stabilizer States

1. Every $P \in \mathcal{S}\left(|\psi\rangle\right)$ is a tensor product $P = \pm P_1 \otimes \cdots \otimes P_n$ of n 2×2 Pauli matrices, which each have eigenvalues 1 and -1.
2. So P also has eigenvalues $1, -1$, each with 2^{n-1}-dimensiaonal eigenspaces. . .
3. . . . which means that the linear equation $P\,|\psi\rangle = |\psi\rangle$ splits the 2^n dimensional state space in half, restricting it to the 2^{n-1}-dimensional subspace with eigenvalue 1.

An $\mathcal{O}\left(2n^2\right)$ Representation of Stabilizer States

1. Every $P \in \mathcal{S}\left(|\psi\rangle\right)$ is a tensor product $P = \pm P_1 \otimes \cdots \otimes P_n$ of n 2×2 Pauli matrices, which each have eigenvalues 1 and -1.
2. So P also has eigenvalues $1, -1$, each with 2^{n-1}-dimensiaonal eigenspaces...
3. ...which means that the linear equation $P\,|\psi\rangle = |\psi\rangle$ splits the 2^n dimensional state space in half, restricting it to the 2^{n-1}-dimensional subspace with eigenvalue 1.
4. Rinse and repeat: each independent generator P of $\mathcal{S}\left(|\psi\rangle\right)$ halves the dimension of the subspace $\rightsquigarrow$ n independent generators uniquely determine $|\psi\rangle$.

An $\mathcal{O}\left(2n^2\right)$ Representation of Stabilizer States

1. Every $P \in \mathcal{S}\left(|\psi\rangle\right)$ is a tensor product $P = \pm P_1 \otimes \cdots \otimes P_n$ of n 2×2 Pauli matrices, which each have eigenvalues 1 and -1.
2. So P also has eigenvalues $1, -1$, each with 2^{n-1}-dimensiaonal eigenspaces...
3. ...which means that the linear equation $P|\psi\rangle = |\psi\rangle$ splits the 2^n dimensional state space in half, restricting it to the 2^{n-1}-dimensional subspace with eigenvalue 1.
4. Rinse and repeat: each independent generator P of $\mathcal{S}\left(|\psi\rangle\right)$ halves the dimension of the subspace $\rightsquigarrow$ n independent generators uniquely determine $|\psi\rangle$.
5. Thus $|\mathcal{S}\left(|\psi\rangle\right)| = 2^k$ with $k \leq n$, and $k = n$ if and only if $|\psi\rangle$ is a stabilizer state.

An $\mathcal{O}\left(2n^2\right)$ Representation of Stabilizer States

1. Every $P \in \mathcal{S}\left(|\psi\rangle\right)$ is a tensor product $P = \pm P_1 \otimes \cdots \otimes P_n$ of n 2×2 Pauli matrices, which each have eigenvalues 1 and -1.
2. So P also has eigenvalues $1, -1$, each with 2^{n-1}-dimensiaonal eigenspaces...
3. ...which means that the linear equation $P\,|\psi\rangle = |\psi\rangle$ splits the 2^n dimensional state space in half, restricting it to the 2^{n-1}-dimensional subspace with eigenvalue 1.
4. Rinse and repeat: each independent generator P of $\mathcal{S}\left(|\psi\rangle\right)$ halves the dimension of the subspace $\rightsquigarrow$ n independent generators uniquely determine $|\psi\rangle$.
5. Thus $\left|\mathcal{S}\left(|\psi\rangle\right)\right| = 2^k$ with $k \leq n$, and $k = n$ if and only if $|\psi\rangle$ is a stabilizer state.

The n independent Pauli group elements that uniquely determine a stabiliser state can be used as a representation of the state instead of the 2^n coefficients.

We'll see how we can directly act on this representation with quantum gates, so long as these are *Clifford gates*.

2-qubit stabilizer states and their generators

	STATE	GEN'TORS	∠	STATE	GEN'TORS	∠	STATE	GEN'TORS	∠	STATE
SEPARABLE	$1,1,1,1$	IX, XI	$\pi/3$	$1,-1,1,-1$	-IX, XI	$\pi/3$	$1,1,-1,-1$	IX, -XI	$\pi/3$	$1,-1,-1,1$
	$1,1,i,i$	IX, YI	$\pi/3$	$1,-1,i,-i$	-IX, YI	$\pi/3$	$1,1,-i,-i$	IX, -YI	$\pi/3$	$1,-1,-i,i$
	$1,1,0,0$	IX, ZI	$\pi/4$	$1,-1,0,0$	-IX, ZI	$\pi/4$	$0,0,1,1$	IX, -ZI	$\perp$	$0,0,1,-1$
	$1,i,1,i$	IY, XI	$\pi/3$	$1,-i,1,-i$	-IY, XI	$\pi/3$	$1,i,-1,-i$	IY, -XI	$\pi/3$	$1,-i,-1,i$
	$1,i,i,-1$	IY, YI	$\pi/3$	$1,-i,i,1$	-IY, YI	$\pi/3$	$1,i,-i,1$	IY, -YI	$\pi/3$	$1,-i,-i,-1$
	$1,i,0,0$	IY, ZI	$\pi/4$	$1,-i,0,0$	-IY, ZI	$\pi/4$	$0,0,1,i$	IY, -ZI	$\perp$	$0,0,1,-i$
	$1,0,1,0$	IZ, XI	$\pi/4$	$0,1,0,1$	-IZ, XI	$\perp$	$1,0,-1,0$	IZ, -XI	$\pi/4$	$0,1,0,-1$
	$1,0,i,0$	IZ, YI	$\pi/4$	$0,1,0,i$	-IZ, YI	$\perp$	$1,0,-i,0$	IZ, -YI	$\pi/4$	$0,1,0,-i$
	$\mathbf{1,0,0,0}$	**IZ, ZI**	0	$\mathbf{0,1,0,0}$	**-IZ, ZI**	$\perp$	$\mathbf{0,0,1,0}$	**IZ, -ZI**	$\perp$	$\mathbf{0,0,0,1}$
ENTANGLED	$0,1,1,0$	XX, YY	$\perp$	$1,0,0,-1$	-XX, YY	$\pi/4$	$1,0,0,1$	XX, -YY	$\pi/4$	$0,1,-1,0$
	$1,0,0,i$	XY, YX	$\pi/4$	$0,1,i,0$	-XY, YX	$\perp$	$0,1,-i,0$	XY, -YX	$\perp$	$1,0,0,-i$
	$1,1,1,-1$	XZ, ZX	$\pi/3$	$1,1,-1,1$	-XZ, ZX	$\pi/3$	$1,-1,1,1$	XZ, -ZX	$\pi/3$	$1,-1,-1,-1$
	$1,i,1,-i$	XZ, ZY	$\pi/3$	$1,i,-1,i$	-XZ, ZY	$\pi/3$	$1,-i,1,i$	XZ, -ZY	$\pi/3$	$1,-i,-1,-i$
	$1,1,i,-i$	YZ, ZX	$\pi/3$	$1,1,-i,i$	-YZ, ZX	$\pi/3$	$1,-1,i,i$	YZ, -ZX	$\pi/3$	$1,-1,-i,-i$
	$1,i,i,1$	YZ, ZY	$\pi/3$	$1,i,-i,-1$	-YZ, ZY	$\pi/3$	$1,-i,i,-1$	YZ, -ZY	$\pi/3$	$1,-i,-i,1$

Representing Stabilizer states with $2n^2$ bits

Before we look into the Clifford circuits, let's look at how we can do calculations extremely efficiently with these representations of Stabilizer states.
Because $Y = iXZ$, we can represent all the Pauli matrices with two bits, x and z, as:

$$\boldsymbol{\sigma} = i^{x\,z} X^x Z^z$$

i.e., $[\mathbf{I}] = 00, [Z] = 01, [X] = 10, [Y] = 11$ (where $[\cdot]$ maps group elements to bit-string representation). So the product of two Pauli matrices becomes

$$\begin{aligned}
\boldsymbol{\sigma_1 \sigma_2} &= (i^{l_1} X^{x_1} Z^{z_1})(i^{l_2} X^{x_2} Z^{z_2}) \\
&= i^{l_1+l_2} X^{x_1} Z^{z_1} X^{x_2} Z^{z_2} \\
&= (-1)^{z_1\,x_2} i^{l_1+l_2} X^{x_1+x_2} Z^{z_1+z_2} \\
&= (phase) \cdot X^{x_1 \oplus x_2} Z^{z_1 \oplus z_2}
\end{aligned}$$

where $\oplus$ is bitwise XOR.

Representing Stabilizer states with $2n^2$ bits

Thus, we can represent a general n-qubit Pauli group element by a $2n+2$-bit string $(s, c, \mathbf{x}, \mathbf{z})$

$$[P] = [(-1)^s i^c X^{x_0} Z^{z_0} \otimes \cdots \otimes X^{x_{n-1}} Z^{z_{n-1}}] = (s, c, \mathbf{x}, \mathbf{z})$$

Representing Stabilizer States

We can represent a Stabilizer state $|\psi\rangle$ by an $n \times (2n+2)$ bit-matrix, where each row represents a generator of $\mathcal{S}\left(|\psi\rangle\right)$. This is called a *stabilizer tableau*.

Pauli-group operations on bit-strings

The two n-qubit stabilizers can be multiplied simply as

$$[P_1 P_2] = [P_1] \oplus [P_2] \oplus \texttt{sign-correction}$$

where $\texttt{sign-correction} = \oplus(\mathbf{z}_1 \text{ and } \mathbf{x}_2)$ only acts on the sign-bit.

This is extremely efficient on multiple levels! It's a single bitwise machine instruction applied to wide bit-strings, i.e. embarassingly data-parallel.

Example

Let $P_1 = -X \otimes I \otimes Z$ and $P_2 = -Z \otimes Z \otimes Y = -iZ \otimes Z \otimes (XZ)$.

Example

Let $P_1 = -X \otimes I \otimes Z$ and $P_2 = -Z \otimes Z \otimes Y = -iZ \otimes Z \otimes (XZ)$.

	sc	$x_0x_1x_2$	$z_0z_1z_2$
$[P_1]$	10	100	001
$[P_2]$	11	001	111
$[P_1] \oplus [P_2]$	01	101	110
$\oplus(zz^1$ and $\mathbf{x}^2)$	1		
$[P_1 P_2]$	11	101	110

Example

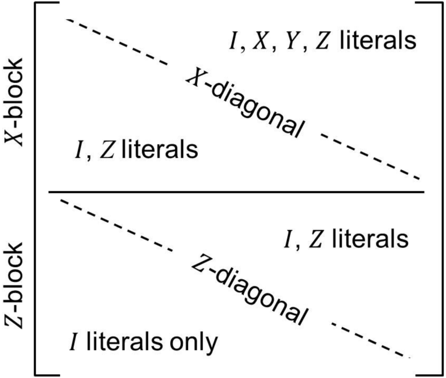
Let $P_1 = -X \otimes I \otimes Z$ and $P_2 = -Z \otimes Z \otimes Y = -iZ \otimes Z \otimes (XZ)$.

	sc	$x_0x_1x_2$	$z_0z_1z_2$
$[P_1]$	10	100	001
$[P_2]$	11	001	111
$[P_1] \oplus [P_2]$	01	101	110
$\oplus(zz^1$ and $\mathbf{x}^2)$	1		
$[P_1P_2]$	11	101	110

$$\begin{aligned} P_1P_2 &= (-)^1 i^1 (X^1Z^1) \otimes (X^0Z^1) \otimes (X^1Z^0) \\ &= -(iXZ) \otimes Z \otimes X \\ &= -Y \otimes Z \otimes X \end{aligned}$$

Canonical representations

Any given stabilizer state has many representations: any n independent generators of $\mathcal{S}\left(|\psi\rangle\right)$ will do. García et al. (2014) p.8-10 (Alg. 2) shows how to transform a stabilizer representation into a canonical form:



Whenever you see the words "canonical representation", you should immediately think: *Yes! I can now compare, sort, and hash efficiently.*

Table of Contents

Applying Quantum Gates to Stabilizer States

Now that we have an efficient represenation of stabilizer states through their stabilizer tableau, what operations can we perform on them, such that we get another stabilizer state?
Let $P_0, \ldots, P_{n-1}$ be the n independent generators of $\mathcal{S}(|\psi\rangle)$. If a quantum gate U yields another stabiliser state $U|\psi\rangle$, then its stabilizers are $UP_kU^\dagger$, because:

$$(UP_kU^\dagger)U|\psi\rangle = UP_k|\psi\rangle = U|\psi\rangle$$

I.e., the unitary operations that map stabilizer states to stabilizer states are exactly the ones that map Pauli group elements to Pauli group elements under conjugation.

The Clifford group

The *Clifford group* is defined as the group of unitary operators that map Pauli group elements to Pauli group elements under conjugation. These are the operations we can apply to stabilizer states to get another stabilizer state, and hence simulate efficiently on classical computers!

There are many possible choices of gates that generate the Clifford group: the standard ones are the Hadamard gate H, the phase gate P, and the $CNOT$ ($= CX$) gate, which act like follows:

GATE	INPUT	OUTPUT
H	X	Z
	Y	-Y
	Z	X
P	X	Y
	Y	-X
	Z	Z

GATE	INPUT	OUTPUT
$CNOT$	I_1X_2	I_1X_2
	X_1I_2	X_1X_2
	I_1Y_2	Z_1Y_2
	Y_1I_2	Y_1X_2
	I_1Z_2	Z_1Z_2
	Z_1I_2	Z_1I_2

Other standard gates that are in the Clifford group include $X, Y, Z, CZ, CY, CZ^{-}, CP$, and more.

Simulating Clifford circuits

García et al. (2014), Theorem 7:

Any n-qubit stabilizer state $|\psi\rangle$ can be obtained by applying a stabilizer circuit to the $|0\cdots 0\rangle$ computational-basis state.

I.e., we start with the zero-state, which has canonical stabilizer tableau

$$|0\cdots 0\rangle \simeq \begin{bmatrix} Z & I & \cdots & I \\ I & Z & \cdots & I \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \cdots & Z \end{bmatrix}$$

and then apply each Clifford operation U by conjugating every row $P_k \mapsto UP_kU^\dagger$ (using precomputed bitwise transformation rules in parallel).

1- and 2-qubit Clifford gates only act on 1 or 2 columns, which makes it possible to perform the updates in $\mathcal{O}(n)$ time per gate, for a total classical runtime of $\mathcal{O}(n^2 + nM)$ for M gates.

Table of Contents

Moving Clifford gates through non-Cliffords

The rewrite rules for "moving" Clifford gates through non-Clifford gates depend on the chosen gate instruction set.

Many models use Clifford + T-gates, where $T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \simeq R_z(-\pi/4)$.

However, this requires many T-gates to approximate arbitrary rotations, which are directly supported on some quantum architectures. Notably, for topological quantum devices, $R_z(\theta)$ is "free" since it's implemented by waiting.

Commutation rules for Clifford+$R_x(\pi/4)$, $R_z(\pi/4)$

$R_z(\theta)H = HR_x(\theta)$

$R_z(\theta)S = SR_z(\theta)$

$(R_z(\theta) \otimes I)\, CNOT = CNOT\, (R_z(\theta) \otimes I)$

$(I \otimes R_x(\theta))\, CNOT = CNOT\, (I \otimes R_x(\theta))$

$R_x(\theta)H = HR_z(\theta)$

$R_z(\theta)P = PR_z(\theta)$

$(I \otimes R_z(\pi/4))\, CNOT = CNOT$

$\cdots$ (Nontrivial to work out)

Literature

Aaronson, S. and Gottesman, D. (2004). Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70:052328.

García, H., Markov, I., and Cross, A. (2014). On the geometry of stabilizer states. *Quantum Info. Comput.*, 14(7 & 8):683–720.