Improved Synthesis of Toffoli-Hadamard Circuits

Matthew Amy[1], Andrew N. Glaudell[2], Sarah Meng Li[3], Neil J. Ross[2]

[1]School of Computing Science, Simon Fraser University
[2]Photonic Inc.
[3]Institute for Quantum Computing, University of Waterloo
[4]Department of Mathematics and Statistics, Dalhousie University

April 3rd, 2023

Background

Quantum Algorithm: Carrying out a well-defined task.

Background

Quantum Algorithm: Carrying out a well-defined task.

Quantum Compilation: Program $\Rightarrow$ Sequence of elementary quantum gates.

Background

Quantum Algorithm: Carrying out a well-defined task.

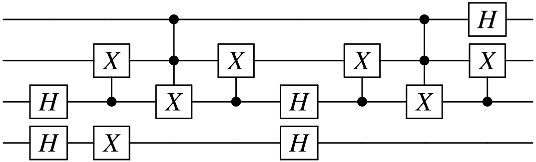
Quantum Compilation: Program $\Rightarrow$ Sequence of elementary quantum gates.

Implementation: Mapping unitary operations to physical architectures.

Restricted Clifford+T Circuits[1]

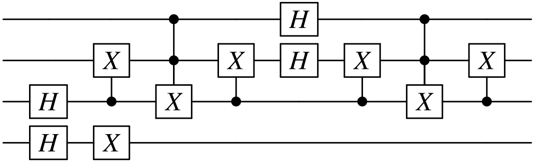
Toffoli-Hadamard circuits are quantum circuits over the gate set

$$\{X, CX, CCX, H\}.$$

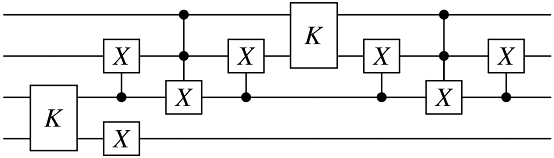


[1]Amy, M., Glaudell, A. N., & Ross, N. J. (2020). Number-theoretic characterizations of some restricted Clifford+T circuits. Quantum, 4, 252.

Restricted Clifford+T Circuits



A Toffoli-Hadamard Circuit



A Toffoli-K Circuit

Basic Gates

$$(-1) = [-1]$$

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad K = H \otimes H = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \left[\begin{array}{c|c} I_2 & \mathbf{0} \\ \hline \mathbf{0} & X \end{array}\right], \quad CCX = \left[\begin{array}{c|c} I_6 & \mathbf{0} \\ \hline \mathbf{0} & X \end{array}\right]$$

The Circuit-Matrix Correspondence

- A family of quantum circuits corresponds to a group of matrices.

- Studying matrix groups is a way to study quantum circuits.

Orthogonal Scaled Dyadic Matrices

- $\mathcal{L}_n$ is the group of ***orthogonal scaled dyadic matrices*** , which consists of $n \times n$ orthogonal matrices of the form $M/\sqrt{2}^k$, where M is an integer matrix and k is a nonnegative integer.

Orthogonal Scaled Dyadic Matrices

- $\mathcal{L}_n$ is the group of ***orthogonal scaled dyadic matrices*** , which consists of $n \times n$ orthogonal matrices of the form $M/\sqrt{2}^k$, where M is an integer matrix and k is a nonnegative integer.

- Example: $V \in \mathcal{L}_4$

$$V = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

Orthogonal Dyadic Matrices

- $\mathbb{Z}\left[\frac{1}{2}\right] = \left\{\frac{u}{2^q} | u \in \mathbb{Z}, q \in \mathbb{N}\right\}$ is the ring of ***dyadic fractions***.

Orthogonal Dyadic Matrices

- $\mathbb{Z}\left[\frac{1}{2}\right] = \left\{\frac{u}{2^q} | u \in \mathbb{Z}, q \in \mathbb{N}\right\}$ is the ring of ***dyadic fractions***.
- $\mathrm{O}_n(\mathbb{Z}\left[\frac{1}{2}\right])$ is the group of ***orthogonal dyadic matrices***, which consists of $n \times n$ orthogonal matrices of the form $M/2^k$, where M is an integer matrix and k is a nonnegative integer. For short, we denote it as $\mathcal{O}_n$.

Orthogonal Dyadic Matrices

- $\mathbb{Z}\left[\frac{1}{2}\right] = \left\{\frac{u}{2^q} | u \in \mathbb{Z}, q \in \mathbb{N}\right\}$ is the ring of ***dyadic fractions***.

- $\mathrm{O}_n(\mathbb{Z}\left[\frac{1}{2}\right])$ is the group of ***orthogonal dyadic matrices***, which consists of $n \times n$ orthogonal matrices of the form $M/2^k$, where M is an integer matrix and k is a nonnegative integer. For short, we denote it as $\mathcal{O}_n$.

- Example: $U \in \mathcal{O}_5$

$$U = \begin{bmatrix} 3/4 & 1/4 & -1/4 & 1/4 & 1/2 \\ 1/4 & 3/4 & 1/4 & -1/4 & -1/2 \\ -1/4 & 1/4 & 3/4 & 1/4 & 1/2 \\ 1/4 & -1/4 & 1/4 & 3/4 & -1/2 \\ -1/2 & 1/2 & -1/2 & 1/2 & 0 \end{bmatrix}$$

Orthogonal Dyadic Matrices

- $\mathbb{Z}\left[\frac{1}{2}\right] = \left\{\frac{u}{2^q} | u \in \mathbb{Z}, q \in \mathbb{N}\right\}$ is the ring of ***dyadic fractions***.

- $\mathrm{O}_n(\mathbb{Z}\left[\frac{1}{2}\right])$ is the group of ***orthogonal dyadic matrices***, which consists of $n \times n$ orthogonal matrices of the form $M/2^k$, where M is an integer matrix and k is a nonnegative integer. For short, we denote it as $\mathcal{O}_n$.

- Example: $U \in \mathcal{O}_5$

$$U = \begin{bmatrix} 3/4 & 1/4 & -1/4 & 1/4 & 1/2 \\ 1/4 & 3/4 & 1/4 & -1/4 & -1/2 \\ -1/4 & 1/4 & 3/4 & 1/4 & 1/2 \\ 1/4 & -1/4 & 1/4 & 3/4 & -1/2 \\ -1/2 & 1/2 & -1/2 & 1/2 & 0 \end{bmatrix}$$

Orthogonal Dyadic Matrices

- $\mathbb{Z}\left[\frac{1}{2}\right] = \left\{\frac{u}{2^q} | u \in \mathbb{Z}, q \in \mathbb{N}\right\}$ is the ring of ***dyadic fractions***.

- $\mathrm{O}_n(\mathbb{Z}\left[\frac{1}{2}\right])$ is the group of ***orthogonal dyadic matrices***, which consists of $n \times n$ orthogonal matrices of the form $M/2^k$, where M is an integer matrix and k is a nonnegative integer. For short, we denote it as $\mathcal{O}_n$.

- Example: $U \in \mathcal{O}_5$

$$U = \frac{1}{2^2} \begin{bmatrix} 3 & 1 & -1 & 1 & 2 \\ 1 & 3 & 1 & -1 & -2 \\ -1 & 1 & 3 & 1 & 2 \\ 1 & -1 & 1 & 3 & -2 \\ -2 & 2 & -2 & 2 & 0 \end{bmatrix}$$

Orthogonal (Scaled) Dyadic Matrices

- $\mathcal{L}_n$ is the group of ***orthogonal scaled dyadic matrices*** , which consists of $n \times n$ orthogonal matrices of the form $M/\sqrt{2}^k$, where M is an integer matrix and k is a nonnegative integer.

- $\mathcal{O}_n$ is the group of ***orthogonal dyadic matrices***, which consists of $n \times n$ orthogonal matrices of the form $M/2^k$, where M is an integer matrix and k is a nonnegative integer.

- $\mathcal{O}_n \subset \mathcal{L}_n$.

Example: $V \in \mathcal{L}_4$

$$V = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

Example: $U \in \mathcal{O}_4$

$$U = \frac{1}{2} \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 \end{bmatrix}$$

Orthogonal (Scaled) Dyadic Matrices

- $\mathcal{L}_n$ is the group of ***orthogonal scaled dyadic matrices*** , which consists of $n \times n$ orthogonal matrices of the form $M/\sqrt{2}^k$, where M is an integer matrix and k is a nonnegative integer.

- $\mathcal{O}_n$ is the group of ***orthogonal dyadic matrices***, which consists of $n \times n$ orthogonal matrices of the form $M/2^k$, where M is an integer matrix and k is a nonnegative integer.

- $\mathcal{O}_n \subset \mathcal{L}_n$.

Example: $V \in \mathcal{L}_4$

$$V = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

Example: $U \in \mathcal{O}_4$

$$U = \frac{1}{\sqrt{2}^2} \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 \end{bmatrix}$$

Constructive Membership Problem (CMP)

Let $\mathcal{G}$ be a group and let $\mathcal{S}$ be a set of generators for $\mathcal{G}$. The *constructive membership problem for* $\mathcal{G}$ *and* $\mathcal{S}$, denoted $\mathcal{P}(\mathcal{G}, \mathcal{S})$, is the following:

Given $g \in \mathcal{G}$, find a sequence of generators $s_1, \ldots, s_\ell \in \mathcal{S}$ such that

$$s_1 \cdot \ldots \cdot s_\ell = g,$$

where $\cdot$ is the group operation.

- The smaller the ℓ, the better the solution.

Constructive Membership Problem (CMP)

Let $\mathcal{G}$ be a group and let $\mathcal{S}$ be a set of generators for $\mathcal{G}$. The *constructive membership problem for $\mathcal{G}$ and $\mathcal{S}$*, denoted $\mathcal{P}(\mathcal{G}, \mathcal{S})$, is the following:

Given $g \in \mathcal{G}$, find a sequence of generators $s_1, \ldots, s_\ell \in \mathcal{S}$ such that

$$s_1 \cdot \ldots \cdot s_\ell = g,$$

where $\cdot$ is the group operation.

- The smaller the ℓ, the better the solution.
- A solution is optimal if the sequence is a shortest possible sequence.

Constructive Membership Problem (CMP)

Let $\mathcal{G}$ be a group and let $\mathcal{S}$ be a set of generators for $\mathcal{G}$. The *constructive membership problem for $\mathcal{G}$ and $\mathcal{S}$*, denoted $\mathcal{P}(\mathcal{G}, \mathcal{S})$, is the following:

Given $g \in \mathcal{G}$, find a sequence of generators $s_1, \ldots, s_\ell \in \mathcal{S}$ such that

$$s_1 \cdot \ldots \cdot s_\ell = g,$$

where $\cdot$ is the group operation.

- The smaller the ℓ, the better the solution.
- A solution is optimal if the sequence is a shortest possible sequence.
- An algorithm to solve the CMP is called an ***exact synthesis algorithm***.

The Circuit-Matrix Correspondence I

Theorem (Solutions to CMP: The AGR Algorithm[1])

For an n-dimensional orthogonal matrix U,

- *it can be exactly represented by a circuit over $\{X, CX, CCX, H\}$ iff $U \in \mathcal{L}_n$.*
- *it can be exactly represented by a circuit over $\{X, CX, CCX, K\}$ iff $U \in \mathcal{O}_n$.*

The gate complexity of the AGR algorithm in both cases is $O(2^n \log(n) k)$.

- A good solution to CMP yields shorter quantum circuits.

[1] Amy, M., Glaudell, A. N., & Ross, N. J. (2020). Number-theoretic characterizations of some restricted Clifford+T circuits. Quantum, 4, 252.

The Circuit-Matrix Correspondence I

Theorem (Solutions to CMP: The AGR Algorithm[1])

For an n-dimensional orthogonal matrix U,

- *it can be exactly represented by a circuit over $\{X, CX, CCX, H\}$ iff $U \in \mathcal{L}_n$.*
- *it can be exactly represented by a circuit over $\{X, CX, CCX, K\}$ iff $U \in \mathcal{O}_n$.*

The gate complexity of the AGR algorithm in both cases is $O(2^n \log(n)k)$.

- A good solution to CMP yields shorter quantum circuits.
- **Can we find a good solution to the CMP for $\mathcal{O}_n$ and $\mathcal{L}_n$?**

[1] Amy, M., Glaudell, A. N., & Ross, N. J. (2020). Number-theoretic characterizations of some restricted Clifford+T circuits. Quantum, 4, 252.

Two-level Operator: $U_{[\alpha,\beta]}$

Definition

Let $U = \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix}$. The action of $U_{[\alpha,\beta]}$, $1 \leq \alpha < \beta \leq n$, is defined as

$$U_{[\alpha,\beta]} v = w, \text{ where } \begin{cases} \begin{bmatrix} w_\alpha \\ w_\beta \end{bmatrix} = U \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix}, \\ w_i = v_i, i \notin \{\alpha, \beta\}. \end{cases}$$

Example:

Let $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Then $X_{[2,3]} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and $X_{[2,3]} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_3 \\ v_2 \\ v_4 \end{bmatrix}$.

Four-level Operator: $U_{[\alpha,\beta,\gamma,\delta]}$

Similarly, we can create a four-level operator by embedding a 4×4 matrix U into an $n \times n$ identity matrix.

Let $K = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$. Then $K_{[1,2,4,6]} = \begin{bmatrix} 1/2 & 1/2 & 0 & 1/2 & 0 & 1/2 \\ 1/2 & -1/2 & 0 & 1/2 & 0 & -1/2 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & -1/2 & 0 & -1/2 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1/2 & -1/2 & 0 & -1/2 & 0 & 1/2 \end{bmatrix}$.

$$K_{[1,2,4,6]} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} = \begin{bmatrix} (v_1 + v_2 + v_4 + v_6)/2 \\ (v_1 - v_2 + v_4 - v_6)/2 \\ v_3 \\ (v_1 + v_2 - v_4 - v_6)/2 \\ v_5 \\ (v_1 - v_2 - v_4 + v_6)/2 \end{bmatrix}.$$

The Circuit-Matrix Correspondence II

$$\mathcal{F}_n = \left\{(-1)_{[\alpha]}, X_{[\alpha,\beta]}, K_{[\alpha,\beta,\gamma,\delta]}, I_{n/2} \otimes H : 1 \leq \alpha < \beta < \gamma < \delta \leq n\right\}.$$
$$\mathcal{G}_n = \left\{(-1)_{[\alpha]}, X_{[\alpha,\beta]}, K_{[\alpha,\beta,\gamma,\delta]} : 1 \leq \alpha < \beta < \gamma < \delta \leq n\right\}.$$

Theorem (Solutions to CMP: The AGR Algorithm[1])

Let U be an $n \times n$ matrix.

- *$U \in \mathcal{L}_n$ iff U can be written as a product of elements of $\mathcal{F}_n$.*
- *$U \in \mathcal{O}_n$ iff U can be written as a product of elements of $\mathcal{G}_n$.*

- When $n = 2^m$, every operator in $\mathcal{G}_n$ and $\mathcal{F}_n$ can be exactly represented by $O(\log(n))$ operators in $\{X, CX, CCX, K\}$ and $\{X, CX, CCX, H\}$, respectively.

[1]Amy, M., Glaudell, A. N., & Ross, N. J. (2020). Number-theoretic characterizations of some restricted Clifford+T circuits. Quantum, 4, 252.

The Least Denominator Exponent (LDE)

Let $t \in \mathbb{Z}\left[\frac{1}{2}\right]$. $t = \frac{a}{2^k}$, where $a \in \mathbb{Z}$ and $k \in \mathbb{N}$. k is a *denominator exponent* for t. The minimal such k is called the ***least denominator exponent*** of t, written $\operatorname{lde}(t)$.

Example: LDE of a column vector

$$v = \frac{1}{2^7}\begin{bmatrix}54\\62\\98\\2\\2\\2\\2\\2\end{bmatrix} = \frac{2}{2^7}\begin{bmatrix}27\\31\\49\\1\\1\\1\\1\\1\end{bmatrix} = \frac{1}{2^6}\begin{bmatrix}27\\31\\49\\1\\1\\1\\1\\1\end{bmatrix}$$

$$\operatorname{lde}(v) = 6$$

Example: LDE of a matrix

$$U = \frac{1}{2}\begin{bmatrix}-1 & 1 & 1 & 0 & 1 & 0 & 0 & 0\\ -1 & -1 & 0 & 1 & 0 & 1 & 0 & 0\\ -1 & 1 & -1 & 0 & -1 & 0 & 0 & 0\\ -1 & -1 & 0 & -1 & 0 & -1 & 0 & 0\\ 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0\\ 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\end{bmatrix}$$

$$\operatorname{lde}(U) = 1$$

The Least Denominator Exponent (LDE)

Let $t \in \mathbb{Z}\left[\frac{1}{2}\right]$. $t = \frac{a}{2^k}$, where $a \in \mathbb{Z}$ and $k \in \mathbb{N}$. k is a *denominator exponent* for t. The minimal such k is called the ***least denominator exponent*** of t, written $\mathrm{lde}(t)$.

Example: LDE of a column vector

$$v = \frac{1}{2^7}\begin{bmatrix}54\\62\\98\\2\\2\\2\\2\\2\end{bmatrix} = \frac{2}{2^7}\begin{bmatrix}27\\31\\49\\1\\1\\1\\1\\1\end{bmatrix} = \frac{1}{2^6}\begin{bmatrix}27\\31\\49\\1\\1\\1\\1\\1\end{bmatrix}$$

$$\mathrm{lde}(v) = 6$$

Example: LDE of a matrix

$$U = \frac{1}{2}\begin{bmatrix}-1 & 1 & 1 & 0 & 1 & 0 & 0 & 0\\-1 & -1 & 0 & 1 & 0 & 1 & 0 & 0\\-1 & 1 & -1 & 0 & -1 & 0 & 0 & 0\\-1 & -1 & 0 & -1 & 0 & -1 & 0 & 0\\0 & 0 & 1 & 1 & -1 & -1 & 0 & 0\\0 & 0 & 1 & -1 & -1 & 1 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 0 & 2 & 0\\0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\end{bmatrix}$$

$$\mathrm{lde}(U) = 1$$

The Least Denominator Exponent (LDE)

Let $t \in \mathbb{Z}\left[\frac{1}{2}\right]$. $t = \frac{a}{2^k}$, where $a \in \mathbb{Z}$ and $k \in \mathbb{N}$. k is a *denominator exponent* for t. The minimal such k is called the ***least denominator exponent*** of t, written $\mathrm{lde}(t)$.

Example: LDE of a column vector

$$v = \frac{1}{2^7}\begin{bmatrix}54\\62\\98\\2\\2\\2\\2\\2\end{bmatrix} = \frac{2}{2^7}\begin{bmatrix}27\\31\\49\\1\\1\\1\\1\\1\end{bmatrix} = \frac{1}{2^6}\begin{bmatrix}27\\31\\49\\1\\1\\1\\1\\1\end{bmatrix}$$

$$\mathrm{lde}(v) = 6$$

Example: LDE of a matrix

$$U = \frac{1}{2}\begin{bmatrix}-1 & 1 & 1 & 0 & 1 & 0 & 0 & 0\\-1 & -1 & 0 & 1 & 0 & 1 & 0 & 0\\-1 & 1 & -1 & 0 & -1 & 0 & 0 & 0\\-1 & -1 & 0 & -1 & 0 & -1 & 0 & 0\\0 & 0 & 1 & 1 & -1 & -1 & 0 & 0\\0 & 0 & 1 & -1 & -1 & 1 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 0 & 2 & 0\\0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\end{bmatrix}$$

$$\mathrm{lde}(U) = 1$$

Lemma (Base Case)

Let $v \in \mathbb{Z}\left[\frac{1}{2}\right]^n$ *be a unit vector. Let* $k = \operatorname{lde}(v)$*. If* $k = 0$*, then* $v = \pm e_j$ *for some* $j \in \{1, \ldots, n\}$.

Lemma (Count)

Let $v \in \mathbb{Z}\left[\frac{1}{2}\right]^n$ *be a unit vector, and* $\operatorname{lde}(v) = k > 0$*. Let* $w = 2^k v$*. Then the number of odd entries in* w *is a multiple of* 4.

Lemma (Parity Reduction)

Let u_1, u_2, u_3, u_4 *be odd integers. Then there exist* $\tau_1, \tau_2, \tau_3, \tau_4 \in \mathbb{Z}_2$ *such that*

$$K_{[1,2,3,4]}(-1)^{\tau_1}_{[1]}(-1)^{\tau_2}_{[2]}(-1)^{\tau_3}_{[3]}(-1)^{\tau_4}_{[4]} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} u_1' \\ u_2' \\ u_3' \\ u_4' \end{bmatrix}, \ u_1', u_2', u_3', u_4' \text{ are even integers.}$$

The AGR Algorithm (I)

Example: Input: $v \in \mathbb{Z}[\frac{1}{2}]^8$ Output: G_1, G_2, G_3 Result: $G_3 \cdot G_2 \cdot G_1 \cdot v = e_1$

$$v: \frac{1}{4}\begin{pmatrix} -1 \\ 1 \\ -1 \\ -1 \\ 3 \\ 1 \\ 1 \\ 1 \end{pmatrix} \xrightarrow{G_1 = K_{[1,2,3,4]}(-1)_{[4]}(-1)_{[3]}(-1)_{[1]}} v': \frac{1}{4}\begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 3 \\ 1 \\ 1 \\ 1 \end{pmatrix} \xrightarrow{G_2 = K_{[5,6,7,8]}(-1)_{[5]}}$$

$\mathrm{lde}(v) = 2$ $\mathrm{lde}(v') = 2$

$$v'': \frac{1}{4}\begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2 \\ -2 \\ -2 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ -1 \end{pmatrix} \xrightarrow{G_3 = K_{[1,6,7,8]}(-1)_{[8]}(-1)_{[7]}(-1)_{[6]}} v''': \frac{1}{2}\begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = e_1.$$

$\mathrm{lde}(v'') = 1$ $\mathrm{lde}(v''') = 0$

The AGR Algorithm (II)

- The algorithm proceeds one column at a time, reducing each column to a corresponding basis vector.
- While outputting a word $\overrightarrow{G_\ell}$ after each iteration, the algorithm recursively acts on the input matrix until it is reduced to the identity matrix $\mathbb{I}$.

$$M \xrightarrow{\overrightarrow{G_1}} \left(\begin{array}{ccc|c} & & & 0 \\ & M' & & \vdots \\ & & & 0 \\ \hline 0 & \cdots & 0 & 1 \end{array}\right) \xrightarrow{\overrightarrow{G_2}} \left(\begin{array}{ccc|cc} & & & 0 & 0 \\ & M'' & & \vdots & \vdots \\ & & & 0 & 0 \\ \hline 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 1 \end{array}\right) \xrightarrow{\overrightarrow{G_3}} \cdots \xrightarrow{\overrightarrow{G_\ell}} \mathbb{I}$$

$$\overrightarrow{G_\ell} \cdot \cdots \cdot \overrightarrow{G_1} M = \mathbb{I} \Rightarrow M = \overrightarrow{G_1}^{-1} \cdot \cdots \cdot \overrightarrow{G_\ell}^{-1}$$

The AGR Algorithm (II)

- The algorithm proceeds one column at a time, reducing each column to a corresponding basis vector.
- While outputting a word $\overrightarrow{G_\ell}$ after each iteration, the algorithm recursively acts on the input matrix until it is reduced to the identity matrix $\mathbb{I}$.

$$M \xrightarrow{\overrightarrow{G_1}} \left(\begin{array}{ccc|c} & & & 0 \\ & M' & & \vdots \\ & & & 0 \\ \hline 0 & \cdots & 0 & 1 \end{array}\right) \xrightarrow{\overrightarrow{G_2}} \left(\begin{array}{ccc|cc} & & & 0 & 0 \\ & M'' & & \vdots & \vdots \\ & & & 0 & 0 \\ \hline 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 1 \end{array}\right) \xrightarrow{\overrightarrow{G_3}} \cdots \xrightarrow{\overrightarrow{G_\ell}} \mathbb{I}$$

$\mathbf{e_n}$

$$\overrightarrow{G_\ell} \cdot \cdots \cdot \overrightarrow{G_1} M = \mathbb{I} \Rightarrow M = \overrightarrow{G_1}^{-1} \cdot \cdots \cdot \overrightarrow{G_\ell}^{-1}$$

The AGR Algorithm (II)

- The algorithm proceeds one column at a time, reducing each column to a corresponding basis vector.
- While outputting a word $\overrightarrow{G_\ell}$ after each iteration, the algorithm recursively acts on the input matrix until it is reduced to the identity matrix $\mathbb{I}$.

$$M \xrightarrow{\overrightarrow{G_1}} \left(\begin{array}{ccc|c} & & & 0 \\ & M' & & \vdots \\ & & & 0 \\ \hline 0 & \cdots & 0 & 1 \end{array}\right) \xrightarrow{\overrightarrow{G_2}} \left(\begin{array}{ccc|cc} & & & 0 & 0 \\ & M'' & & \vdots & \vdots \\ & & & 0 & 0 \\ \hline 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 1 \end{array}\right) \xrightarrow{\overrightarrow{G_3}} \cdots \xrightarrow{\overrightarrow{G_\ell}} \mathbb{I}$$

$\mathbf{e_n}$ $\quad$ $\mathbf{e_{n-1}}$

$$\overrightarrow{G_\ell} \cdot \cdots \cdot \overrightarrow{G_1} M = \mathbb{I} \Rightarrow M = \overrightarrow{G_1}^{-1} \cdot \cdots \cdot \overrightarrow{G_\ell}^{-1}$$

Gate Complexity of the AGR Algorithm

Lemma

Let $\mathbf{u} \in \mathbb{Z}\left[\frac{1}{2}\right]^n$ *with* $\mathrm{lde}(\mathbf{u}) = k$*. The number of generators in* $\mathcal{G}_n$ *to reduce* $\mathbf{u}$ *to* $\mathbf{e}_j$ *is* $O(nk)$.

Theorem

Let $U \in \mathcal{O}_n$ *with* $\mathrm{lde}(U) = k$*.* U *can be exactly represented by* $O(2^n k)$ *generators over* $\mathcal{G}_n$.

Proof. Let $f_{\mathbf{u}_i}$ be the cost of reducing $\mathbf{u}_i$ to $\mathbf{e}_i$.

- Each row operation may increase the lde of any column in U by 1.
- During reduction, the lde of any other column may increase up to $2k$.

$$f_{\mathbf{u}_1} = O\left(nk\right), \quad f_{\mathbf{u}_2} = O\left((n-1)2k\right), \quad f_{\mathbf{u}_3} = O\left((n-2)2^2k\right), \quad \ldots, \quad f_{\mathbf{u}_n} = O\left(2^{n-1}k\right).$$

$$S_n = \sum_{i=1}^{n} f_{\mathbf{u}_i} = \sum_{i=1}^{n} (n-i+1)2^{i-1}k = O(2^n k).$$

□

The Householder Algorithm[2]

With **ancillary qubits**, the gate complexity of the exact synthesis for $\mathcal{L}_n$ over $\mathcal{F}_n$ is reduced from $O(2^n k)$ to $O(n^2 k)$.

[2]Vadym Kliuchnikov (2013). "Synthesis of unitaries with Clifford+ T circuits". In: *arXiv preprint arXiv:1306.3200*.

The Householder Algorithm[2]

With **ancillary qubits**, the gate complexity of the exact synthesis for $\mathcal{L}_n$ over $\mathcal{F}_n$ is reduced from $O(2^n k)$ to $O(n^2 k)$.

Definition

For an n-dimensional unit vector $|\psi\rangle$, the reflection operator around $|\psi\rangle$ is

$$R_{|\psi\rangle} = I - 2\,|\psi\rangle\,\langle\psi|\,.$$

[2]Vadym Kliuchnikov (2013). "Synthesis of unitaries with Clifford+ T circuits". In: *arXiv preprint arXiv:1306.3200*.

The Householder Algorithm[2]

With **ancillary qubits**, the gate complexity of the exact synthesis for $\mathcal{L}_n$ over $\mathcal{F}_n$ is reduced from $O(2^n k)$ to $O(n^2 k)$.

Definition

For an n-dimensional unit vector $|\psi\rangle$, the reflection operator around $|\psi\rangle$ is

$$R_{|\psi\rangle} = I - 2\,|\psi\rangle\langle\psi|\,.$$

Proposition: Gate Complexity of the Reflection Operator

Let $|\psi\rangle = \mathbf{v}/\sqrt{2}^k$ be an n-dimensional unit vector with $\mathrm{lde}_{\sqrt{2}}(|\psi\rangle) = k$, $\mathbf{v}$ is an integer vector. The reflection operator $R_{|\psi\rangle}$ can be exactly represented by $O(nk)$ generators over $\mathcal{F}_n$.

[2] Vadym Kliuchnikov (2013). “Synthesis of unitaries with Clifford+ T circuits”. In: *arXiv preprint arXiv:1306.3200*.

Unitary Simulation

Let $U \in \mathcal{L}_n$. Then U can be simulated using the unitary

$$U' = |+\rangle \langle -| \otimes U + |-\rangle \langle +| \otimes U^\dagger.$$

Moreover, $U' \in \mathcal{L}_{2n}$ and U' can be factored as a product $U' = \prod_{j=1}^{n} R_{|\omega_j^-\rangle}$ of reflection operators around vectors

$$|\omega_j^-\rangle = \frac{\left(|-\rangle |j\rangle - |+\rangle |\mathbf{u}_j\rangle\right)}{\sqrt{2}},$$

$\mathbf{u}_j$ is the j-th column vector in U and $|j\rangle$ is the j-th computational basis vector.

Unitary Simulation

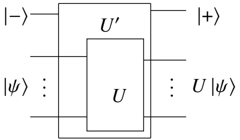
Let $U \in \mathcal{L}_n$. Then U can be simulated using the unitary

$$U' = |+\rangle\langle-| \otimes U + |-\rangle\langle+| \otimes U^{\dagger}.$$

Moreover, $U' \in \mathcal{L}_{2n}$ and U' can be factored as a product $U' = \prod_{j=1}^{n} R_{|\omega_j^-\rangle}$ of reflection operators around vectors

$$|\omega_j^-\rangle = \frac{\left(|-\rangle|j\rangle - |+\rangle|\mathbf{u}_j\rangle\right)}{\sqrt{2}},$$

$\mathbf{u}_j$ is the j-th column vector in U and $|j\rangle$ is the j-th computational basis vector.



Unitary Simulation

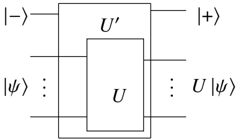
Let $U \in \mathcal{L}_n$. Then U can be simulated using the unitary

$$U' = |+\rangle\langle-| \otimes U + |-\rangle\langle+| \otimes U^\dagger.$$

Moreover, $U' \in \mathcal{L}_{2n}$ and U' can be factored as a product $U' = \prod_{j=1}^{n} R_{|\omega_j^-\rangle}$ of reflection operators around vectors

$$|\omega_j^-\rangle = \frac{\left(|-\rangle|j\rangle - |+\rangle|\mathbf{u}_j\rangle\right)}{\sqrt{2}},$$

$\mathbf{u}_j$ is the j-th column vector in U and $|j\rangle$ is the j-th computational basis vector.



- $I = \sum_{j=1}^{n} \left(|\omega_j^+\rangle\langle\omega_j^+| + |\omega_j^-\rangle\langle\omega_j^-|\right)$ The completeness relation.

Unitary Simulation

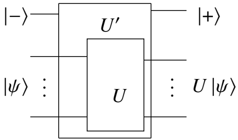
Let $U \in \mathcal{L}_n$. Then U can be simulated using the unitary

$$U' = |+\rangle\langle-| \otimes U + |-\rangle\langle+| \otimes U^\dagger.$$

Moreover, $U' \in \mathcal{L}_{2n}$ and U' can be factored as a product $U' = \prod_{j=1}^{n} R_{|\omega_j^-\rangle}$ of reflection operators around vectors

$$|\omega_j^-\rangle = \frac{\left(|-\rangle|j\rangle - |+\rangle|\mathbf{u}_j\rangle\right)}{\sqrt{2}},$$

$\mathbf{u}_j$ is the j-th column vector in U and $|j\rangle$ is the j-th computational basis vector.



- $I = \sum_{j=1}^{n}\left(|\omega_j^+\rangle\langle\omega_j^+| + |\omega_j^-\rangle\langle\omega_j^-|\right)$ The completeness relation.
- $U' = \sum_{j=1}^{n}\left(|\omega_j^+\rangle\langle\omega_j^+| - |\omega_j^-\rangle\langle\omega_j^-|\right)$ The spectral theorem.

Unitary Simulation

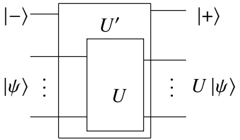
Let $U \in \mathcal{L}_n$. Then U can be simulated using the unitary

$$U' = |+\rangle\langle -| \otimes U + |-\rangle\langle +| \otimes U^\dagger.$$

Moreover, $U' \in \mathcal{L}_{2n}$ and U' can be factored as a product $U' = \prod_{j=1}^{n} R_{|\omega_j^-\rangle}$ of reflection operators around vectors

$$|\omega_j^-\rangle = \frac{\big(|-\rangle|j\rangle - |+\rangle|\mathbf{u}_j\rangle\big)}{\sqrt{2}},$$

$\mathbf{u}_j$ is the j-th column vector in U and $|j\rangle$ is the j-th computational basis vector.



- $I = \sum_{j=1}^{n}\left(|\omega_j^+\rangle\langle\omega_j^+| + |\omega_j^-\rangle\langle\omega_j^-|\right)$ The completeness relation.
- $U' = \sum_{j=1}^{n}\left(|\omega_j^+\rangle\langle\omega_j^+| - |\omega_j^-\rangle\langle\omega_j^-|\right)$ The spectral theorem.

$$I - U' = 2\sum_{j=1}^{n}|\omega_j^-\rangle\langle\omega_j^-| \Rightarrow U' = I - 2\sum_{j=1}^{n}|\omega_j^-\rangle\langle\omega_j^-| = \prod_{j=1}^{n} R_{|\omega_j^-\rangle}.$$

Gate Complexity of the Householder Algorithm

Theorem

Let $U \in \mathcal{L}_n$ with $\mathrm{lde}_{\sqrt{2}}(U) = k$. Then U can be represented by $O(n^2 k)$ generators over $\mathcal{F}_n$ using the Householder algorithm.

Gate Complexity of the Householder Algorithm

Theorem

Let $U \in \mathcal{L}_n$ with $\mathrm{lde}_{\sqrt{2}}(U) = k$. Then U can be represented by $O(n^2 k)$ generators over $\mathcal{F}_n$ using the Householder algorithm.

Proof. We showed that U can be simulated by U' where

$$U' = |+\rangle \langle -| \otimes U + |-\rangle \langle +| \otimes U^\dagger = \prod_{j=1}^{n} R_{|\omega_j^-\rangle}.$$

Moreover, each $R_{|\omega_j^-\rangle}$ can be exactly represented by $O(nk)$ generators from $\mathcal{F}_n$. Therefore, to represent U, we need $n \cdot O(nk) = O(n^2 k)$ generators over $\mathcal{F}_n$. □

The Global Synthesis Algorithm

- The AGR algorithm carries out matrix factorization **locally** - it synthesizes one column at a time.

The Global Synthesis Algorithm

- The AGR algorithm carries out matrix factorization **locally** - it synthesizes one column at a time.
- When n is fixed, both AGR and householder algorithms have the same worst-case gate complexity – linear in k.

The Global Synthesis Algorithm

- The AGR algorithm carries out matrix factorization **locally** - it synthesizes one column at a time.
- When n is fixed, both AGR and householder algorithms have the same worst-case gate complexity – linear in k.
- Next, we will take a **global** view of each matrix. This results in a **smaller** gate count in practice.

The Global Synthesis Algorithm

- The AGR algorithm carries out matrix factorization **locally** - it synthesizes one column at a time.
- When n is fixed, both AGR and householder algorithms have the same worst-case gate complexity – linear in k.
- Next, we will take a **global** view of each matrix. This results in a **smaller** gate count in practice.

$$O(n^2 k) \implies O(k)$$

The Global Synthesis Algorithm

- The AGR algorithm carries out matrix factorization **locally** - it synthesizes one column at a time.

- When n is fixed, both AGR and householder algorithms have the same worst-case gate complexity – linear in k.

- Next, we will take a **global** view of each matrix. This results in a **smaller** gate count in practice.

$$O(n^2 k) \Longrightarrow O(k)$$

- Define a global synthesis method for $U \in \mathcal{L}_8$, then **leverage** this to find a global synthesis method for $U' \in \mathcal{O}_8$.

Intuition

$U \in \mathcal{L}_8$. Write $U = \frac{1}{\sqrt{2}^k} M$ with k minimal. There exists $\overrightarrow{G_1}, \dots, \overrightarrow{G_k}$ over $\mathcal{F}$, such that

$$\frac{1}{\sqrt{2}^k} M \xrightarrow{\overrightarrow{G_1}} \frac{1}{\sqrt{2}^{k-1}} M' \xrightarrow{\overrightarrow{G_2}} \frac{1}{\sqrt{2}^{k-2}} M'' \xrightarrow{\overrightarrow{G_3}} \cdots \xrightarrow{\overrightarrow{G_k}} \mathbb{I}.$$

Therefore,

$$\overrightarrow{G_k} \cdot \dots \cdot \overrightarrow{G_1} U = \mathbb{I} \implies U = \overrightarrow{G_1}^{-1} \cdot \dots \cdot \overrightarrow{G_k}^{-1}.$$

Preliminaries

Binary Pattern

Let $U \in \mathcal{L}_n$. Write $U = \frac{1}{\sqrt{2}^k} M$ with k minimal. The residue mod 2 of M is called the ***binary pattern*** of U, denoted as $\overline{U}$.

Example: $U \in \mathcal{L}_5$

$$U = \frac{1}{\sqrt{2}^4} \begin{bmatrix} 3 & 1 & -1 & 1 & 2 \\ 1 & 3 & 1 & -1 & -2 \\ -1 & 1 & 3 & 1 & 2 \\ 1 & -1 & 1 & 3 & -2 \\ -2 & 2 & -2 & 2 & 0 \end{bmatrix} \to \overline{U} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Number Theoretic Property I

Weight Lemma

Let $U \in \mathcal{L}_n$ and $\mathrm{lde}_{\sqrt{2}}(U) = k \geq 2$. Let $\mathbf{u}$ be an arbitrary column vector in $\overline{U}$. Then

$$|\{u_i; u_i = 1, 1 \leq i \leq n\}| \equiv 0(4).$$

In other words, in each column of $\overline{U}$, the 1's occur in quadruples.

Number Theoretic Property I

Weight Lemma

Let $U \in \mathcal{L}_n$ and $\mathrm{lde}_{\sqrt{2}}(U) = k \geq 2$. Let $\mathbf{u}$ be an arbitrary column vector in $\overline{U}$. Then

$$|\{u_i; u_i = 1, 1 \leq i \leq n\}| \equiv 0(4).$$

In other words, in each column of $\overline{U}$, the 1's occur in quadruples.

Proof. Let $\mathbf{v}$ be a column vector in U and $\mathbf{v} = \frac{1}{\sqrt{2}^k}\mathbf{w}$, where $\mathbf{w} \in \mathbb{Z}^n$. Since $\langle \mathbf{v}, \mathbf{v} \rangle = 1$, $\langle \mathbf{w}, \mathbf{w} \rangle = 2^k$ and thus $\sum w_i^2 = 2^k$. When $k \geq 2$, $\sum w_i^2 \equiv 0(4)$. Note that

$$w_i^2 \equiv 1(4) \iff w_i \equiv 1(2), \quad w_i^2 \equiv 0(4) \iff w_i \equiv 0(2).$$

Hence the number of odd entries in $\mathbf{w}$ is a multiple of 4. □

Number Theoretic Property II

Intuition: The 1's in any two distinct columns of $\overline{U}$ collide evenly many times.

Collision Lemma

Let $U \in \mathcal{L}_n$ and $\mathrm{lde}_{\sqrt{2}}(U) = k > 0$. Any two distinct columns in $\overline{U}$ must have evenly many 1's in common.

Example: Evenly many collisions

$$u_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, u_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Example: Oddly many collisions

$$u_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, u_4 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Number Theoretic Property II

Intuition: The 1's in any two distinct columns of $\overline{U}$ collide evenly many times.

Collision Lemma

Let $U \in \mathcal{L}_n$ and $\mathrm{lde}_{\sqrt{2}}(U) = k > 0$. Any two distinct columns in $\overline{U}$ must have evenly many 1's in common.

Example: Evenly many collisions

$$u_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, u_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Example: Oddly many collisions

$$u_3 = \begin{bmatrix} \color{red}{1} \\ \color{red}{1} \\ \color{red}{1} \\ 1 \\ 0 \\ 0 \end{bmatrix}, u_4 = \begin{bmatrix} \color{red}{1} \\ \color{red}{1} \\ \color{red}{1} \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Binary Patterns of $\mathcal{L}_8$

Theorem

There exists a set $\mathcal{P}$ of 14 binary patterns such that if $U \in \mathcal{L}_8$ and $\mathrm{lde}(U) \geq 2$, then $\overline{U} \in \mathcal{P}$ (up to row and column permutations, as well as taking transpose).

Proof. By a long case distinction using the Weight and Collision Lemmas.

Binary patterns that are "nice".

$$A=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\end{bmatrix},\quad B=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&0&0&0&0\\1&1&1&1&0&0&0&0\\1&1&1&1&0&0&0&0\\1&1&1&1&0&0&0&0\end{bmatrix},\ldots,K=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$$

Binary patterns that are "not nice".

$$L=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&0&0&0&0\\1&1&0&0&1&1&0&0\\1&1&0&0&0&0&1&1\\1&0&1&0&1&0&1&0\\1&0&1&0&0&1&0&1\\1&0&0&1&1&0&0&1\\1&0&0&1&0&1&1&0\end{bmatrix},\quad M=\begin{bmatrix}1&1&1&1&0&0&0&0\\1&1&0&0&1&1&0&0\\1&0&1&0&1&0&1&0\\1&0&0&1&0&1&1&0\\0&1&1&0&1&0&0&1\\0&1&0&1&0&1&0&1\\0&0&1&1&0&0&1&1\\0&0&0&0&1&1&1&1\end{bmatrix},\quad N=\begin{bmatrix}1&1&1&1&0&0&0&0\\1&1&0&0&1&1&0&0\\1&0&1&0&1&0&1&0\\1&0&0&1&0&1&1&0\\0&1&1&0&0&1&1&0\\0&1&0&1&1&0&1&0\\0&0&1&1&1&1&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$$

Binary patterns that are "nice".

$$A=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\end{bmatrix},\quad B=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&0&0&0&0\\1&1&1&1&0&0&0&0\\1&1&1&1&0&0&0&0\\1&1&1&1&0&0&0&0\end{bmatrix},\ldots,K=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$$

Binary patterns that are "not nice".

$$L=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&0&0&0&0\\1&1&0&0&1&1&0&0\\1&1&0&0&0&0&1&1\\1&0&1&0&1&0&1&0\\1&0&1&0&0&1&0&1\\1&0&0&1&1&0&0&1\\1&0&0&1&0&1&1&0\end{bmatrix},\quad M=\begin{bmatrix}1&1&1&1&0&0&0&0\\1&1&0&0&1&1&0&0\\1&0&1&0&1&0&1&0\\1&0&0&1&0&1&1&0\\0&1&1&0&1&0&0&1\\0&1&0&1&0&1&0&1\\0&0&1&1&0&0&1&1\\0&0&0&0&1&1&1&1\end{bmatrix},\quad N=\begin{bmatrix}1&1&1&1&0&0&0&0\\1&1&0&0&1&1&0&0\\1&0&1&0&1&0&1&0\\1&0&0&1&0&1&1&0\\0&1&1&0&0&1&1&0\\0&1&0&1&1&0&1&0\\0&0&1&1&1&1&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$$

Binary patterns that are "nice".

$$A = \begin{bmatrix} 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1 \end{bmatrix}, \quad B = \begin{bmatrix} 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&0&0&0&0\\ 1&1&1&1&0&0&0&0\\ 1&1&1&1&0&0&0&0\\ 1&1&1&1&0&0&0&0 \end{bmatrix}, \ldots, K = \begin{bmatrix} 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0 \end{bmatrix}$$

Binary patterns that are "not nice".

$$L = \begin{bmatrix} \color{red}{1}&\color{red}{1}&\color{red}{1}&\color{red}{1}&\color{red}{1}&\color{red}{1}&\color{red}{1}&\color{red}{1}\\ 1&1&1&1&0&0&0&0\\ 1&1&0&0&1&1&0&0\\ 1&1&0&0&0&0&1&1\\ 1&0&1&0&1&0&1&0\\ 1&0&1&0&0&1&0&1\\ 1&0&0&1&1&0&0&1\\ 1&0&0&1&0&1&1&0 \end{bmatrix}, \quad M = \begin{bmatrix} 1&1&1&1&0&0&0&0\\ 1&1&0&0&1&1&0&0\\ 1&0&1&0&1&0&1&0\\ 1&0&0&1&0&1&1&0\\ 0&1&1&0&1&0&0&1\\ 0&1&0&1&0&1&0&1\\ 0&0&1&1&0&0&1&1\\ 0&0&0&0&1&1&1&1 \end{bmatrix}, \quad N = \begin{bmatrix} 1&1&1&1&0&0&0&0\\ 1&1&0&0&1&1&0&0\\ 1&0&1&0&1&0&1&0\\ 1&0&0&1&0&1&1&0\\ 0&1&1&0&0&1&1&0\\ 0&1&0&1&1&0&1&0\\ 0&0&1&1&1&1&0&0\\ 0&0&0&0&0&0&0&0 \end{bmatrix}$$

Definition

A matrix $\overline{U} \in \mathbb{Z}_2^{8\times 8}$ is ***row-paired*** if identical rows occur evenly many times.

Definition

A matrix $\overline{U} \in \mathbb{Z}_2^{8\times 8}$ is ***column-paired*** if identical columns occur evenly many times.

Remark: We demonstrate an example and a counterexample when $n = 4$.

Example: Row-paired and column-paired

$$\overline{U} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Example: Only column-paired

$$\overline{V} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Definition

A matrix $\overline{U} \in \mathbb{Z}_2^{8\times 8}$ is ***row-paired*** if identical rows occur evenly many times.

Definition

A matrix $\overline{U} \in \mathbb{Z}_2^{8\times 8}$ is ***column-paired*** if identical columns occur evenly many times.

Remark: We demonstrate an example and a counterexample when $n = 4$.

Example: Row-paired and column-paired

$$\overline{U} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Example: Only column-paired

$$\overline{V} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Definition

A matrix $\overline{U} \in \mathbb{Z}_2^{8\times 8}$ is ***row-paired*** if identical rows occur evenly many times.

Definition

A matrix $\overline{U} \in \mathbb{Z}_2^{8\times 8}$ is ***column-paired*** if identical columns occur evenly many times.

Remark: We demonstrate an example and a counterexample when $n = 4$.

Example: Row-paired and column-paired

$$\overline{U} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Example: Only column-paired

$$\overline{V} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Definition

A matrix $\overline{U} \in \mathbb{Z}_2^{8\times 8}$ is ***row-paired*** if identical rows occur evenly many times.

Definition

A matrix $\overline{U} \in \mathbb{Z}_2^{8\times 8}$ is ***column-paired*** if identical columns occur evenly many times.

Remark: We demonstrate an example and a counterexample when $n = 4$.

Example: Row-paired and column-paired

$$\overline{U} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Example: Only column-paired

$$\overline{V} = \begin{bmatrix} \color{red}{0} & 1 & \color{red}{0} & 1 \\ \color{red}{1} & 1 & \color{red}{1} & 1 \\ \color{red}{0} & 1 & \color{red}{0} & 1 \\ \color{red}{0} & 1 & \color{red}{0} & 1 \end{bmatrix}$$

Definition

A matrix $\overline{U} \in \mathbb{Z}_2^{8\times 8}$ is ***row-paired*** if identical rows occur evenly many times.

Definition

A matrix $\overline{U} \in \mathbb{Z}_2^{8\times 8}$ is ***column-paired*** if identical columns occur evenly many times.

Remark: We demonstrate an example and a counterexample when $n = 4$.

Example: Row-paired and column-paired

$$\overline{U} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Example: Only column-paired

$$\overline{V} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

When the Binary Pattern is "Nice"

Theorem (Row-paired Reduction)

If $U \in \mathcal{L}_8$ and $\overline{U}$ is row-paired, then there exists $P \in S_8$ such that $\operatorname{lde}_{\sqrt{2}}(((I \otimes H)\, P)U) < \operatorname{lde}_{\sqrt{2}}(U)$.

Theorem (Column-paired Reduction)

If $U \in \mathcal{L}_8$ and $\overline{U}$ is column-paired, then there exists $P \in S_8$ such that $\operatorname{lde}_{\sqrt{2}}(U(P\,(I \otimes H))) < \operatorname{lde}_{\sqrt{2}}(U)$.

Remark: Below we sketch the proof for the Row-paired Reduction using a 6×6 matrix as an example.

Proof. Consider $U \in \mathcal{L}_6$ with $\operatorname{lde}_{\sqrt{2}}(U) = k$. Since $\overline{U}$ is row-paired, there exists $P \in S_6$ such that

$$PU = \frac{1}{\sqrt{2}^k}\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_6 \end{bmatrix}$$, where $r_1 \equiv r_2(2), r_3 \equiv r_4(2), r_5 \equiv r_6(2)$. Now

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$ and $$I \otimes H = \left[\begin{array}{c|c|c} H & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & H & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & H \end{array}\right]$$. Therefore,

$$(I \otimes H)\, PU = \frac{1}{\sqrt{2}^{k+1}}\begin{bmatrix} r_1 + r_2 \\ r_1 - r_2 \\ r_3 + r_4 \\ r_3 - r_4 \\ r_5 + r_6 \\ r_5 - r_6 \end{bmatrix} = \frac{2}{\sqrt{2}^{k+1}}\begin{bmatrix} r_1' \\ \vdots \\ r_6' \end{bmatrix} = \frac{1}{\sqrt{2}^{k-1}}\begin{bmatrix} r_1' \\ \vdots \\ r_6' \end{bmatrix}$$, where $r_1', \dots, r_6' \in \mathbb{Z}^{1\times 6}$.

Hence $\operatorname{lde}_{\sqrt{2}}(((I \otimes H)\, P)U) < \operatorname{lde}_{\sqrt{2}}(U)$, for some $P \in S_6$.

When the Binary Pattern is "NOT Nice"

Theorem

Consider $U \in \mathcal{L}_8$ *and* $\overline{U}$ *is neither row-paired nor column-paired. Let* $U' = (I \otimes H)\, U\, (I \otimes H)$. *Then* $\overline{U'}$ *is row-paired and* $\mathrm{lde}_{\sqrt{2}}(U') \le \mathrm{lde}_{\sqrt{2}}(U)$.

Proof. By direct computation.

Global Synthesis for $\mathcal{L}_8$

Theorem

Let $U \in \mathcal{L}_8$ and $\mathrm{lde}_{\sqrt{2}}(U) = k$. Then there exists C over $\mathcal{F}$ such that $[\![C]\!] = U$ and the length of C is $O(k)$.

Proof. Let $U \in \mathcal{L}_8$, proceed by induction on k.

- $k \leq 1$, there exists C composed of $(-1)_{[\alpha]}, X_{[\alpha,\beta]}$ and $I \otimes H$ such that $[\![C]\!] = U$ and the length of C is $O(1)$.
- $k \geq 2$, $\overline{U}$ must be one of the 14 binary patterns.
  - If $\overline{U}$ is nice, then $\mathrm{lde}((I \otimes H)\, PU) \leq k - 1$ and proceed recursively with $(I \otimes H)\, PU$.
  - If $\overline{U}$ is not nice, then $(I \otimes H)\, U\, (I \otimes H)$ is nice so $\mathrm{lde}\,((I \otimes H)\, P\, (I \otimes H)\, U\, (I \otimes H)) \leq k - 1$ and proceed recursively with $(I \otimes H)\, P\, (I \otimes H)\, U\, (I \otimes H)$.

Generator Relations for $\mathcal{L}_8$ and $\mathcal{O}_8$[3]

- $\mathcal{L}_8$ is generated by $\mathcal{F} = \left\{(-1)_{[\alpha]}, X_{[\alpha,\beta]}, K_{[\alpha,\beta,\gamma,\delta]}, I \otimes H : 1 \leq \alpha < \beta < \gamma < \delta \leq 8\right\}$.
- $\mathcal{O}_8$ is generated by $\mathcal{G} = \left\{(-1)_{[\alpha]}, X_{[\alpha,\beta]}, K_{[\alpha,\beta,\gamma,\delta]} : 1 \leq \alpha < \beta < \gamma < \delta \leq 8\right\}$.

$$(I \otimes H)(I \otimes H) = \epsilon \quad (1)$$

$$(I \otimes H)(-1)_{[1]} = (-1)_{[1]} X_{[1,2]} (-1)_{[1]} (I \otimes H) \quad (2)$$

$$(I \otimes H) X_{[a,a+1]} = (-1)^{a+1}_{[a+1]} X^{a}_{[a,a+1]} K^{a}_{[a-1,a,a+1,a+2]} (I \otimes H) \quad (3)$$

$$(I \otimes H) K_{[1,2,3,4]} = K_{[1,2,3,4]} (I \otimes H) \quad (4)$$

Intuition: Commuting $I \otimes H$ with an element in $\mathcal{G}$ adds $O(1)$ gates.

[3] Sarah Meng Li, Neil J Ross, and Peter Selinger (2021). “Generators and relations for the group $O_n(\mathbb{Z}[1/2])$”. In: *arXiv preprint arXiv:2106.01175*.

Relations for $\mathcal{L}_8$

Lemma

For any M over $\mathcal{G}$, there exists M' over $\mathcal{G}$ such that $(I \otimes H)\, M = M'\, (I \otimes H)$. Moreover, if M has length $O(m)$, then M' has length $O(m)$.

Example:

$$
\begin{aligned}
(I \otimes H)K_{[1,2,3,4]}(-1)_{[1]}X_{[1,2]}(I \otimes H) &= K_{[1,2,3,4]}(I \otimes H)(-1)_{[1]}X_{[1,2]}(I \otimes H) \\
&= K_{[1,2,3,4]}(-1)_{[1]}X_{[1,2]}(-1)_{[1]}(I \otimes H)X_{[1,2]}(I \otimes H) \\
&= K_{[1,2,3,4]}(-1)_{[1]}X_{[1,2]}(-1)_{[1]}(-1)_{[2]}(I \otimes H)(I \otimes H) \\
&= K_{[1,2,3,4]}(-1)_{[1]}X_{[1,2]}(-1)_{[1]}(-1)_{[2]}.
\end{aligned}
$$

Global Synthesis for $\mathcal{O}_8$

Theorem

Let $U \in \mathcal{O}_8$ and $\mathrm{lde}(U) = k \geq 1$. Then there exists C over $\mathcal{G}$ such that $[[C]] = U$ and the length of C is $O(k)$.

Proof. Let $U \in \mathcal{O}_8$ and $\mathrm{lde}(U) = k$. Then $U \in \mathcal{L}_8$ with $\mathrm{lde}_{\sqrt{2}}(U) = 2k$. Using the global synthesis for $\mathcal{L}_8$, we can express U as a word W over $\mathcal{F}$ with evenly many occurrences of $I \otimes H$, and the length of W is $O(k)$. Consider any subword W_i of the form

$$(I \otimes H)\, C\, (I \otimes H)\,,$$

where C does not contain $I \otimes H$.

Global Synthesis for $\mathcal{O}_8$

Theorem

Consider $U \in \mathcal{O}_8$ and $\mathrm{lde}(U) = k \geq 1$. Then there exists C over $\mathcal{G}$ such that $[[C]] = U$ and the length of C is $O(k)$.

Proof Continued. Suppose the length of W_i is $O(k)$. Then

$$W_i = (I \otimes H)\, C\, (I \otimes H) \longrightarrow C'\, (I \otimes H)\, (I \otimes H) \longrightarrow C'$$

Wi can be rewritten as a word C' over $\mathcal{G}$ of length at most $3 * O(k)$ generators. Hence we can rewrite W as a word W' over $\mathcal{G}$ of length no more than $3 * O(k)$.

Future Work

- **Benchmark** our global synthesis algorithm with other state-of-the-art algorithms to compare their performance in practice.

Future Work

- **Benchmark** our global synthesis algorithm with other state-of-the-art algorithms to compare their performance in practice.

- **Design** a standalone global synthesis for $\mathcal{O}_8$, rather than relying on the corresponding result for $\mathcal{L}_8$ and the commutation of generators.

Future Work

- **Benchmark** our global synthesis algorithm with other state-of-the-art algorithms to compare their performance in practice.
- **Design** a standalone global synthesis for $\mathcal{O}_8$, rather than relying on the corresponding result for $\mathcal{L}_8$ and the commutation of generators.
- **Extend** the global synthesis to arbitrary dimensions: $\mathcal{O}_n$ and $\mathcal{L}_n$.

Future Work

- **Benchmark** our global synthesis algorithm with other state-of-the-art algorithms to compare their performance in practice.
- **Design** a standalone global synthesis for $\mathcal{O}_8$, rather than relying on the corresponding result for $\mathcal{L}_8$ and the commutation of generators.
- **Extend** the global synthesis to arbitrary dimensions: $\mathcal{O}_n$ and $\mathcal{L}_n$.
- **Present** the global synthesis results of $\mathcal{O}_n$ and $\mathcal{L}_n$ in terms of the restricted Clifford+T circuits over $\{X, CX, CCX, K\}$ and $\{X, CX, CCX, H\}$ respectively.


Thank you!