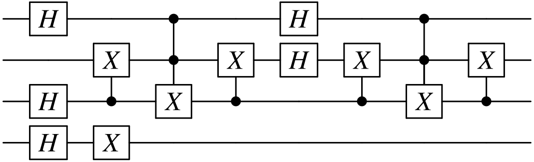
Improved Synthesis of Toffoli-Hadamard Circuits

Matthew Amy[1], Andrew N. Glaudell[2], Sarah Meng Li[3,4], Neil J. Ross[5]

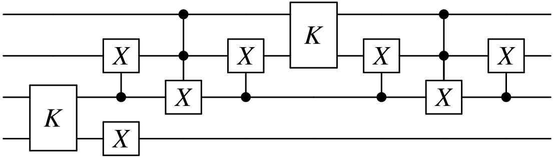
[1] School of Computing Science, Simon Fraser University
[2] Photonic Inc.
[3] Institute for Quantum Computing, University of Waterloo
[4] Department of Combinatorics and Optimization, University of Waterloo
[5] Department of Mathematics and Statistics, Dalhousie University

Paper on arXiv

Restricted Clifford+T Circuits[1]



A Toffoli-Hadamard Circuit



A Toffoli-K Circuit

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

Basic Gates

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

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

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

Our Motivations

- A family of quantum circuits $\Longleftrightarrow$ A group of matrices.

Our Motivations

- A family of quantum circuits $\Longleftrightarrow$ A group of matrices.
- Studying matrix groups is a way to study quantum circuits.

Our Motivations

- A family of quantum circuits $\Longleftrightarrow$ A group of matrices.
- Studying matrix groups is a way to study quantum circuits.
- For the matrix group associated with the Toffoli-Hadamard circuits, use a convenient set of generators and study the factorization of group elements into a sequence of these generators.

Our Motivations

- A family of quantum circuits $\Longleftrightarrow$ A group of matrices.
- Studying matrix groups is a way to study quantum circuits.
- For the matrix group associated with the Toffoli-Hadamard circuits, use a convenient set of generators and study the factorization of group elements into a sequence of these generators.

  $\Rightarrow$ ***The exact synthesis algorithm***

Our Motivations

- A family of quantum circuits $\Longleftrightarrow$ A group of matrices.
- Studying matrix groups is a way to study quantum circuits.
- For the matrix group associated with the Toffoli-Hadamard circuits, use a convenient set of generators and study the factorization of group elements into a sequence of these generators.
  $\Rightarrow$ ***The exact synthesis algorithm***
- A factorization is optimal if the sequence is a shortest possible sequence.

Our Motivations

- A family of quantum circuits $\Longleftrightarrow$ A group of matrices.
- Studying matrix groups is a way to study quantum circuits.
- For the matrix group associated with the Toffoli-Hadamard circuits, use a convenient set of generators and study the factorization of group elements into a sequence of these generators.

  $\Rightarrow$ ***The exact synthesis algorithm***
- A factorization is optimal if the sequence is a shortest possible sequence.
- Each generator can be expressed as a short circuit.

Our Motivations

- A family of quantum circuits $\Longleftrightarrow$ A group of matrices.
- Studying matrix groups is a way to study quantum circuits.
- For the matrix group associated with the Toffoli-Hadamard circuits, use a convenient set of generators and study the factorization of group elements into a sequence of these generators.

  $\Rightarrow$ ***The exact synthesis algorithm***
- A factorization is optimal if the sequence is a shortest possible sequence.
- Each generator can be expressed as a short circuit.

  $\Rightarrow$ A good solution to this factorization problem yields a good synthesis.

Our Results

The Local Synthesis Algorithm

- The gate complexity of the exactly synthesized circuit: $O(2^n \log(n) k)$

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

Our Results

The Local Synthesis Algorithm

- The gate complexity of the exactly synthesized circuit: $O(2^n \log(n) k)$

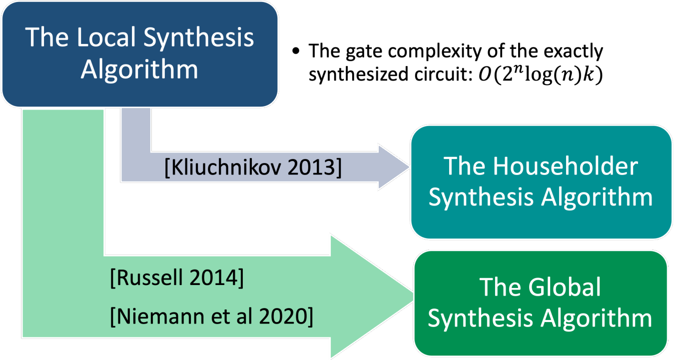
[Kliuchnikov 2013]

The Householder Synthesis Algorithm

- $O(\mathrm{n}^2 \log(n) k)$

[1] Kliuchnikov, V. (2013). Synthesis of unitaries with Clifford+ T circuits. arXiv preprint arXiv:1306.3200.

Our Results



- The gate complexity of the exactly synthesized circuit: $O(2^n \log(n) k)$
- $O(\mathrm{n}^2 \log(n) k)$
- $O(\log(n) k)$

[1] Kliuchnikov, V. (2013). Synthesis of unitaries with Clifford+ T circuits. arXiv preprint arXiv:1306.3200.
[2] Russell, T. (2014). The exact synthesis of 1-and 2-qubit Clifford+ T circuits. arXiv preprint arXiv:1408.6202.
[3] Niemann, P., Wille, R., & Drechsler, R. (2020). Advanced exact synthesis of Clifford+ T circuits. Quantum Information Processing, 19, 1-23.

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***.

- $\mathrm{O}_n\left(\mathbb{Z}\left[\frac{1}{2}\right]\right)$ 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 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***.
- $\mathrm{O}_n(\mathbb{Z}[\frac{1}{2}])$ 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: $U \in \mathcal{O}_5$

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

The Circuit-Matrix Correspondence I

Theorem (The AGR Algorithm[1])

For an n-dimensional orthogonal matrix U, it can be exactly represented by a circuit over $\{X, CX, CCX, K\}$ iff $U \in \mathcal{O}_n$.

[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 (The AGR Algorithm[1])

For an n-dimensional orthogonal matrix U, it can be exactly represented by a circuit over $\{X, CX, CCX, K\}$ iff $U \in \mathcal{O}_n$.

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

[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 (The AGR Algorithm[1])

For an n-dimensional orthogonal matrix U, it can be exactly represented by a circuit over $\{X, CX, CCX, K\}$ iff $U \in \mathcal{O}_n$.

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

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

[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 (The AGR Algorithm[1])

For an n-dimensional orthogonal matrix U, it can be exactly represented by a circuit over $\{X, CX, CCX, K\}$ iff $U \in \mathcal{O}_n$.

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

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

Theorem (The AGR Algorithm)

For an n-dimensional orthogonal matrix U, it can be written as a product of elements of $\mathcal{G}_n$ iff $U \in \mathcal{O}_n$.

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

The 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 $X_{[2,3]} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_3 \\ v_2 \\ v_4 \end{bmatrix}$.

The 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 AGR Algorithm

- 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

- 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

- 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 $\mathbb{I}$.

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

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

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

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

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

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

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

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

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

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 with* $\operatorname{lde}(v) = k$*. If* $k = 0$*,* $v = \pm e_j$ *for some* $j \in \{1, \ldots, n\}$*.*

Lemma (Base Case)

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

Lemma (Weight)

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

Lemma (Base Case)

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

Lemma (Weight)

Let $v \in \mathbb{Z}\left[\frac{1}{2}\right]^n$ *be a unit vector with* $\operatorname{lde}(v) = k$*. Let* $w = 2^k v$*. If* $k > 0$*, 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' *are even integers.*

Example: The Column Reduction

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

$$v: \quad \frac{1}{4}\begin{pmatrix} -1 \\ 1 \\ -1 \\ -1 \\ 3 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

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

Example: The Column Reduction

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

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

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

Example: The Column Reduction

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

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

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

$$v'' : \frac{1}{4}\begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2 \\ -2 \\ -2 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ -1 \end{pmatrix}$$

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

Example: The Column Reduction

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

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

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

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

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

Gate Complexity of the AGR Algorithm

Theorem

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

Gate Complexity of the AGR Algorithm

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 Sketch.

- Each row operation may increase the lde of any column in U by 1.

Gate Complexity of the AGR Algorithm

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 Sketch.

- 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 $2k$.

Gate Complexity of the AGR Algorithm

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 Sketch.

- 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 $2k$.

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

Gate Complexity of the AGR Algorithm

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 Sketch.

- 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 **one ancilla**, the gate complexity of exactly synthesizing $\mathcal{O}_n$ over $\mathcal{G}_n$ is reduced from $O(2^n k)$ to $O(n^2 k)$.

Definition

Let $|\psi\rangle$ be an n-dimensional unit vector. The reflection operator around $|\psi\rangle$ is

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

- $R_{|\psi\rangle} = R^{\dagger}_{|\psi\rangle}$ and $R^2_{|\psi\rangle} = (I - 2\,|\psi\rangle\langle\psi|)(I - 2\,|\psi\rangle\langle\psi|) = I.$

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

The Householder Algorithm[2]

With **one ancilla**, the gate complexity of exactly synthesizing $\mathcal{O}_n$ over $\mathcal{G}_n$ is reduced from $O(2^n k)$ to $O(n^2 k)$.

Definition

Let $|\psi\rangle$ be an n-dimensional unit vector. The reflection operator around $|\psi\rangle$ is

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

- $R_{|\psi\rangle} = R^{\dagger}_{|\psi\rangle}$ and $R^2_{|\psi\rangle} = (I - 2\,|\psi\rangle\langle\psi|)(I - 2\,|\psi\rangle\langle\psi|) = I.$
- $R_{|\psi\rangle}$ is unitary: $R_{|\psi\rangle}R^{\dagger}_{|\psi\rangle} = R^{\dagger}_{|\psi\rangle}R_{|\psi\rangle} = R^2_{|\psi\rangle} = I.$

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

The Householder Algorithm[2]

With **one ancilla**, the gate complexity of exactly synthesizing $\mathcal{O}_n$ over $\mathcal{G}_n$ is reduced from $O(2^n k)$ to $O(n^2 k)$.

Definition

Let $|\psi\rangle$ be an n-dimensional unit vector. The reflection operator around $|\psi\rangle$ is

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

- $R_{|\psi\rangle} = R^{\dagger}_{|\psi\rangle}$ and $R^2_{|\psi\rangle} = (I - 2\,|\psi\rangle\langle\psi|)(I - 2\,|\psi\rangle\langle\psi|) = I$.
- $R_{|\psi\rangle}$ is unitary: $R_{|\psi\rangle} R^{\dagger}_{|\psi\rangle} = R^{\dagger}_{|\psi\rangle} R_{|\psi\rangle} = R^2_{|\psi\rangle} = I$.
- If $|\psi\rangle = |v\rangle / 2^k$, $R_{|\psi\rangle} \in \mathcal{O}_n$.

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

Gate Complexity of the Reflection Operator

Proposition

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

Proof Sketch.

$$|\psi\rangle \longrightarrow \boxed{\text{AGR}} \xrightarrow{G \in \mathcal{G}_n}$$

Gate Complexity of the Reflection Operator

Proposition

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

Proof Sketch.

$$|\psi\rangle \longrightarrow \boxed{\text{AGR}} \xrightarrow{G \in \mathcal{G}_n} |0\rangle = G|\psi\rangle \equiv G^\dagger|0\rangle = |\psi\rangle$$

Gate Complexity of the Reflection Operator

Proposition

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

Proof Sketch.

$$|\psi\rangle \longrightarrow \boxed{\text{AGR}} \xrightarrow{G \in \mathcal{G}_n} |0\rangle = G|\psi\rangle \equiv \boxed{G^\dagger|0\rangle = |\psi\rangle}$$

$$G^\dagger R_{|0\rangle} G = G^\dagger (I - 2|0\rangle\langle 0|) G = I - 2|\psi\rangle\langle\psi| =: R_{|\psi\rangle}$$

Gate Complexity of the Reflection Operator

Proposition

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

Proof Sketch.

$$|\psi\rangle \longrightarrow \boxed{\text{AGR}} \xrightarrow{G \in \mathcal{G}_n} |0\rangle = G|\psi\rangle \equiv \boxed{G^\dagger|0\rangle = |\psi\rangle}$$

$$G^\dagger R_{|0\rangle} G = G^\dagger (I - 2|0\rangle\langle 0|) G = I - 2|\psi\rangle\langle\psi| =: R_{|\psi\rangle}$$

$$R_{|\psi\rangle} = G^\dagger R_{|0\rangle} G \longrightarrow \boxed{\text{AGR}}$$

Gate Complexity of the Reflection Operator

Proposition

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

Proof Sketch.

$$|\psi\rangle \longrightarrow \boxed{\text{AGR}} \xrightarrow{G \in \mathcal{G}_n} |0\rangle = G|\psi\rangle \equiv \boxed{G^\dagger|0\rangle = |\psi\rangle}$$

$$G^\dagger R_{|0\rangle} G = G^\dagger (I - 2|0\rangle\langle 0|) G = I - 2|\psi\rangle\langle\psi| =: R_{|\psi\rangle}$$

$$R_{|\psi\rangle} = G^\dagger R_{|0\rangle} G \longrightarrow \boxed{\text{AGR}} \longrightarrow CC(G^\dagger) = CC(G) = O(nk)$$

Gate Complexity of the Reflection Operator

Proposition

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

Proof Sketch.

$$|\psi\rangle \longrightarrow \boxed{\text{AGR}} \xrightarrow{G \in \mathcal{G}_n} |0\rangle = G|\psi\rangle \equiv \boxed{G^\dagger|0\rangle = |\psi\rangle}$$

$$G^\dagger R_{|0\rangle} G = G^\dagger (I - 2|0\rangle\langle 0|) G = I - 2|\psi\rangle\langle\psi| =: R_{|\psi\rangle}$$

$$CC(G^\dagger) = CC(G) = O(nk)$$

$$R_{|\psi\rangle} = G^\dagger R_{|0\rangle} G \longrightarrow \boxed{\text{AGR}}$$

$$R_{|0\rangle} = (-1)_{[0]}$$

$$CC(R_{|0\rangle}) = O(1)$$

Gate Complexity of the Reflection Operator

Proposition

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

Proof Sketch.

$$|\psi\rangle \longrightarrow \boxed{\text{AGR}} \xrightarrow{G \in \mathcal{G}_n} |0\rangle = G|\psi\rangle \equiv \boxed{G^\dagger|0\rangle = |\psi\rangle}$$

$$G^\dagger R_{|0\rangle} G = G^\dagger(I - 2|0\rangle\langle 0|)G = I - 2|\psi\rangle\langle\psi| =: R_{|\psi\rangle}$$

$$R_{|\psi\rangle} = G^\dagger R_{|0\rangle} G \longrightarrow \boxed{\text{AGR}}$$

$$CC(G^\dagger) = CC(G) = O(nk)$$

$$+ \qquad = O(nk)$$

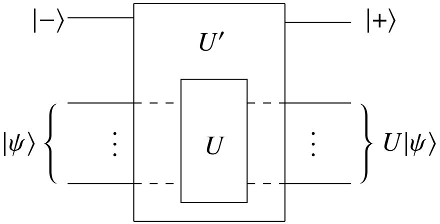
$$R_{|0\rangle} = (-1)_{[0]}$$

$$CC(R_{|0\rangle}) = O(1)$$

Unitary Simulation

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

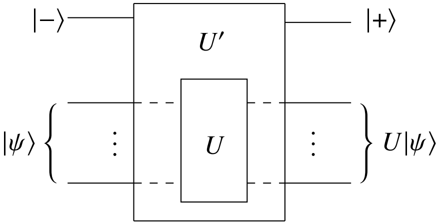
$$U' = |+\rangle \langle -| \otimes U + |-\rangle \langle +| \otimes U^\dagger.$$



Unitary Simulation

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

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

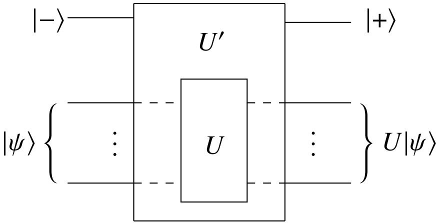


- $U' \in \mathcal{O}_{2n}$ and U' is unitary.

Unitary Simulation

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

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



- $U' \in \mathcal{O}_{2n}$ and U' is unitary.
- U' is Hermitian and thus normal.

Unitary Factorization

Let $|u_j\rangle$ be the j-th column vector in U and $|j\rangle$ be the j-th computational basis vector. U' can be factored into n reflections in $\mathcal{O}_{2n}$.

$$U' = \prod_{j=0}^{n-1} R_{|\omega_j^-\rangle}, \quad |\omega_j^-\rangle = \frac{\big(|-\rangle\,|j\rangle - |+\rangle\,|u_j\rangle\big)}{\sqrt{2}}$$

Unitary Factorization

Let $|u_j\rangle$ be the j-th column vector in U and $|j\rangle$ be the j-th computational basis vector. U' can be factored into n reflections in $\mathcal{O}_{2n}$.

$$U' = \prod_{j=0}^{n-1} R_{|\omega_j^-\rangle}, \quad |\omega_j^-\rangle = \frac{\big(|-\rangle\,|j\rangle - |+\rangle\,|u_j\rangle\big)}{\sqrt{2}}$$

- $\left\{|\omega_j^{\pm}\rangle\,;\ 0 \le j \le n-1\right\}$ forms an orthonomal basis.

Unitary Factorization

Let $|u_j\rangle$ be the j-th column vector in U and $|j\rangle$ be the j-th computational basis vector. U' can be factored into n reflections in $\mathcal{O}_{2n}$.

$$U' = \prod_{j=0}^{n-1} R_{|\omega_j^-\rangle}, \quad |\omega_j^-\rangle = \frac{\left(|-\rangle\,|j\rangle - |+\rangle\,|u_j\rangle\right)}{\sqrt{2}}$$

- $\left\{|\omega_j^\pm\rangle\,;\ 0 \le j \le n-1\right\}$ forms an orthonomal basis.

$\Rightarrow\ I = \sum_{j=0}^{n-1}\left(|\omega_j^+\rangle\,\langle\omega_j^+| + |\omega_j^-\rangle\,\langle\omega_j^-|\right)$ The completeness relation

Unitary Factorization

Let $|u_j\rangle$ be the j-th column vector in U and $|j\rangle$ be the j-th computational basis vector. U' can be factored into n reflections in $\mathcal{O}_{2n}$.

$$U' = \prod_{j=0}^{n-1} R_{|\omega_j^-\rangle}, \quad |\omega_j^-\rangle = \frac{\left(|-\rangle\,|j\rangle - |+\rangle\,|u_j\rangle\right)}{\sqrt{2}}$$

- $\left\{|\omega_j^\pm\rangle\,;\ 0 \le j \le n-1\right\}$ forms an orthonomal basis.

$\Rightarrow$ $I = \sum_{j=0}^{n-1} \left(|\omega_j^+\rangle\,\langle\omega_j^+| + |\omega_j^-\rangle\,\langle\omega_j^-|\right)$ The completeness relation

- $|\omega_j^+\rangle$ and $|\omega_j^-\rangle$ are the $+1$ and -1 eigenstates of U'.

Unitary Factorization

Let $|u_j\rangle$ be the j-th column vector in U and $|j\rangle$ be the j-th computational basis vector. U' can be factored into n reflections in $\mathcal{O}_{2n}$.

$$U' = \prod_{j=0}^{n-1} R_{|\omega_j^-\rangle}, \quad |\omega_j^-\rangle = \frac{(|-\rangle\,|j\rangle - |+\rangle\,|u_j\rangle)}{\sqrt{2}}$$

- $\left\{|\omega_j^{\pm}\rangle\,;\ 0 \le j \le n-1\right\}$ forms an orthonomal basis.

$\Rightarrow$ $I = \sum_{j=0}^{n-1}\left(|\omega_j^+\rangle\,\langle\omega_j^+| + |\omega_j^-\rangle\,\langle\omega_j^-|\right)$ The completeness relation

- $|\omega_j^+\rangle$ and $|\omega_j^-\rangle$ are the $+1$ and -1 eigenstates of U'.

$\Rightarrow$ $U' = \sum_{j=0}^{n-1}\left(|\omega_j^+\rangle\,\langle\omega_j^+| - |\omega_j^-\rangle\,\langle\omega_j^-|\right)$ The spectral theorem

Unitary Factorization

Let $|u_j\rangle$ be the j-th column vector in U and $|j\rangle$ be the j-th computational basis vector. U' can be factored into n reflections in $\mathcal{O}_{2n}$.

$$U' = \prod_{j=0}^{n-1} R_{|\omega_j^-\rangle}, \quad |\omega_j^-\rangle = \frac{\left(|-\rangle\,|j\rangle - |+\rangle\,|u_j\rangle\right)}{\sqrt{2}}$$

- $\left\{|\omega_j^\pm\rangle\,;\ 0 \le j \le n-1\right\}$ forms an orthonomal basis.

$\Rightarrow$ $I = \sum_{j=0}^{n-1}\left(|\omega_j^+\rangle\langle\omega_j^+| + |\omega_j^-\rangle\langle\omega_j^-|\right)$ The completeness relation

- $|\omega_j^+\rangle$ and $|\omega_j^-\rangle$ are the $+1$ and -1 eigenstates of U'.

$\Rightarrow$ $U' = \sum_{j=0}^{n-1}\left(|\omega_j^+\rangle\langle\omega_j^+| - |\omega_j^-\rangle\langle\omega_j^-|\right)$ The spectral theorem

$$I - U' = 2\sum_{j=0}^{n-1}|\omega_j^-\rangle\langle\omega_j^-| \Rightarrow U' = I - 2\sum_{j=0}^{n-1}|\omega_j^-\rangle\langle\omega_j^-| = \prod_{j=0}^{n-1}R_{|\omega_j^-\rangle}.$$

Gate Complexity of the Householder Algorithm

Theorem

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

Proof Sketch.

- U can be simulated by U' where

$$U' = |+\rangle\langle-| \otimes U + |-\rangle\langle+| \otimes U^\dagger = \prod_{j=0}^{n-1} R_{|\omega_j^-\rangle}.$$

Gate Complexity of the Householder Algorithm

Theorem

Let $U \in \mathcal{O}_n$ with $\operatorname{lde}(U) = k$. Then U can be represented by $O(n^2 k)$ generators from $\mathcal{G}_n$ using the Householder algorithm.

Proof Sketch.

- U can be simulated by U' where

$$U' = |+\rangle\langle-| \otimes U + |-\rangle\langle+| \otimes U^\dagger = \prod_{j=0}^{n-1} R_{|\omega_j^-\rangle}.$$

- $R_{|\omega_j^-\rangle}$ can be exactly represented by $O(nk)$ generators from $\mathcal{G}_n$.

Gate Complexity of the Householder Algorithm

Theorem

Let $U \in \mathcal{O}_n$ with $\operatorname{lde}(U) = k$. Then U can be represented by $O(n^2 k)$ generators from $\mathcal{G}_n$ using the Householder algorithm.

Proof Sketch.

- U can be simulated by U' where

$$U' = |+\rangle\langle-| \otimes U + |-\rangle\langle+| \otimes U^\dagger = \prod_{j=0}^{n-1} R_{|\omega_j^-\rangle}.$$

- $R_{|\omega_j^-\rangle}$ can be exactly represented by $O(nk)$ generators from $\mathcal{G}_n$.
- To represent U, we need $n \cdot O(nk) = O(n^2 k)$ generators from $\mathcal{G}_n$.

$\square$

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.

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

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.

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

- To **reduce the gate complexity**, we take a **global** view of each matrix.

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.

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

- To **reduce the gate complexity**, we take a **global** view of each matrix.
- Define a global synthesis method for $U \in \mathcal{L}_8$, then **leverage** this to find a global synthesis method for $U \in \mathcal{O}_8$.

Orthogonal Scaled Dyadic Matrices

Definition

$\mathcal{L}_n$ 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

Definition

$\mathcal{L}_n$ 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 (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.

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.

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

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

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.

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

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

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

The Circuit-Matrix Correspondence II

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

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

The Circuit-Matrix Correspondence II

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

Theorem

Let U be an $n \times n$ matrix. $U \in \mathcal{L}_n$ if and only if

- *U can be written as a product of elements of $\mathcal{F}_n$.*
    - *The gate complexity is $O(2^n k)$.*

The Circuit-Matrix Correspondence II

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

Theorem

Let U be an $n \times n$ matrix. $U \in \mathcal{L}_n$ if and only if

- *U can be written as a product of elements of $\mathcal{F}_n$.*
  - *The gate complexity is $O(2^n k)$.*
- *U can be exactly represented by a circuit over $\{X, CX, CCX, H\}$.*
  - *The gate complexity is $O(2^n \log(n) k)$.*

Intuitions

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

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

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

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

Binary Patterns of $\mathcal{L}_8$

Proposition

Let $U \in \mathcal{L}_8$ with $\mathrm{lde}_{\sqrt{2}}(U) \geq 2$. Then up to row permutation, column permutation, and taking the transpose, $\overline{U}$ is one of the 14 binary patterns.

Proof Sketch. Case distinction using the Weight and Collision Lemmas.

Definition

Let n be even and $B \in \mathbb{Z}_2^{n \times n}$. B is ***row-paired*** if the rows of B can be partitioned into identical pairs.

Example: Row-paired

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

Example: NOT row-paired

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

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.

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

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.

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

Number-Theoretic Properties

Weight Lemma

Let $U \in \mathcal{L}_n$ with $\mathrm{lde}_{\sqrt{2}}(U) = k$. If $k > 1$, the number of 1's in any column of $\overline{U}$ is doubly-even.

Number-Theoretic Properties

Weight Lemma

Let $U \in \mathcal{L}_n$ with $\mathrm{lde}_{\sqrt{2}}(U) = k$. If $k > 1$, the number of 1's in any column of $\overline{U}$ is doubly-even.

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

Collision Lemma

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

Number-Theoretic Properties

Weight Lemma

Let $U \in \mathcal{L}_n$ with $\mathrm{lde}_{\sqrt{2}}(U) = k$. If $k > 1$, the number of 1's in any column of $\overline{U}$ is doubly-even.

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

Collision Lemma

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

Example: Evenly many collisions

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

Example: Oddly many collisions

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

Number-Theoretic Properties

Weight Lemma

Let $U \in \mathcal{L}_n$ with $\mathrm{lde}_{\sqrt{2}}(U) = k$. If $k > 1$, the number of 1's in any column of $\overline{U}$ is doubly-even.

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

Collision Lemma

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

Example: Evenly many collisions

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

Example: Oddly many collisions

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

When the Binary Pattern is NICE

Lemma (Row-Paired Reduction)

Let n be even, $U \in \mathcal{L}_n$ with $\mathrm{lde}_{\sqrt{2}}(U) = k$. If $\overline{U}$ is row-paired, there exists $P \in S_n$ such that $\mathrm{lde}_{\sqrt{2}}\big(((I \otimes H)\, P)U\big) < \mathrm{lde}_{\sqrt{2}}(U)$.

Proof Sketch. Since $\overline{U}$ is row-paired, there exists $P \in S_n$ such that

$$PU = \frac{1}{\sqrt{2}^k}\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}, r_1 \equiv r_2(2),\ \ldots,\ r_{n-1} \equiv r_n(2).$$

When the Binary Pattern is NICE

Lemma (Row-Paired Reduction)

Let n be even, $U \in \mathcal{L}_n$ with $\operatorname{lde}_{\sqrt{2}}(U) = k$. If $\overline{U}$ is row-paired, there exists $P \in S_n$ such that $\operatorname{lde}_{\sqrt{2}}\big(((I \otimes H)\, P)U\big) < \operatorname{lde}_{\sqrt{2}}(U)$.

Proof Sketch. Since $\overline{U}$ is row-paired, there exists $P \in S_n$ such that

$$PU = \frac{1}{\sqrt{2}^k}\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}, r_1 \equiv r_2(2),\ \ldots,\ r_{n-1} \equiv r_n(2).\ H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } I \otimes H = \left[\begin{array}{c|c|c} H & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \ddots & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & H \end{array}\right]$$

implies that $(I \otimes H)\, PU = \frac{1}{\sqrt{2}^{k+1}}\begin{bmatrix} r_1 + r_2 \\ r_1 - r_2 \\ \vdots \\ r_{n-1} - r_n \end{bmatrix} = \frac{2}{\sqrt{2}^{k+1}}\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix} = \frac{1}{\sqrt{2}^{k-1}}\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix}, t_1, t_2,\ \ldots,\ t_n \in \mathbb{Z}.$ □

Global Synthesis for $\mathcal{L}_8$

Lemma (When the Binary Pattern is NOT NICE)

Let $U \in \mathcal{L}_8$ *with* $\mathrm{lde}_{\sqrt{2}}(U) = k$*. If* $\overline{U}$ *is neither row-paired nor column-paired,* $\overline{(I \otimes H)\, U\, (I \otimes H)}$ *is row-paired and* $\mathrm{lde}_{\sqrt{2}}\big((I \otimes H)\, U\, (I \otimes H)\big) \leq \mathrm{lde}_{\sqrt{2}}(U)$*.*

Global Synthesis for $\mathcal{L}_8$

Lemma (When the Binary Pattern is NOT NICE)

Let $U \in \mathcal{L}_8$ *with* $\mathrm{lde}_{\sqrt{2}}(U) = k$. *If* $\overline{U}$ *is neither row-paired nor column-paired,* $\overline{(I \otimes H)\, U\, (I \otimes H)}$ *is row-paired and* $\mathrm{lde}_{\sqrt{2}}\big((I \otimes H)\, U\, (I \otimes H)\big) \leq \mathrm{lde}_{\sqrt{2}}(U)$.

Proposition

Let $U \in \mathcal{L}_8$ *with* $\mathrm{lde}_{\sqrt{2}}(U) = k$. U *can be represented by* $O(k)$ *generators in* $\mathcal{F}_8$ *using the global synthesis algorithm.*

Global Synthesis for $\mathcal{L}_8$

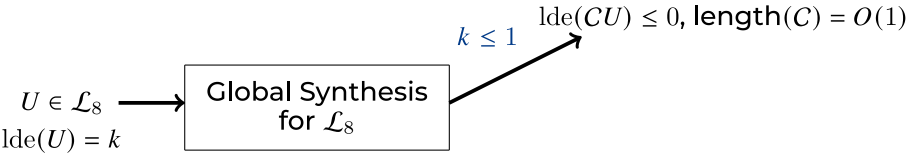
Lemma (When the Binary Pattern is NOT NICE)

Let $U \in \mathcal{L}_8$ *with* $\mathrm{lde}_{\sqrt{2}}(U) = k$. *If* $\overline{U}$ *is neither row-paired nor column-paired,* $\overline{(I \otimes H)\, U\, (I \otimes H)}$ *is row-paired and* $\mathrm{lde}_{\sqrt{2}}\big((I \otimes H)\, U\, (I \otimes H)\big) \leq \mathrm{lde}_{\sqrt{2}}(U)$.

Proposition

Let $U \in \mathcal{L}_8$ *with* $\mathrm{lde}_{\sqrt{2}}(U) = k$. U *can be represented by* $O(k)$ *generators in* $\mathcal{F}_8$ *using the global synthesis algorithm.*

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



Global Synthesis for $\mathcal{L}_8$

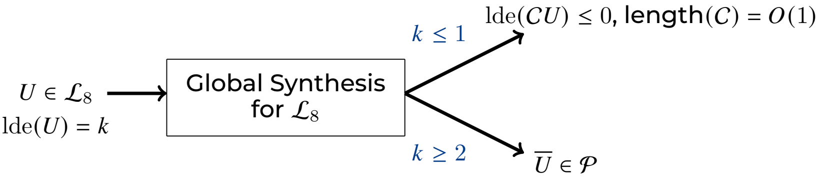
Lemma (When the Binary Pattern is NOT NICE)

Let $U \in \mathcal{L}_8$ with $\mathrm{lde}_{\sqrt{2}}(U) = k$. If $\overline{U}$ is neither row-paired nor column-paired, $\overline{(I \otimes H)\, U\, (I \otimes H)}$ is row-paired and $\mathrm{lde}_{\sqrt{2}}\big((I \otimes H)\, U\, (I \otimes H)\big) \leq \mathrm{lde}_{\sqrt{2}}(U)$.

Proposition

Let $U \in \mathcal{L}_8$ with $\mathrm{lde}_{\sqrt{2}}(U) = k$. U can be represented by $O(k)$ generators in $\mathcal{F}_8$ using the global synthesis algorithm.

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



Global Synthesis for $\mathcal{L}_8$

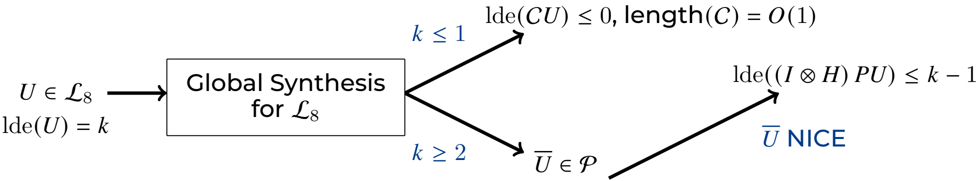
Lemma (When the Binary Pattern is NOT NICE)

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

Proposition

Let $U \in \mathcal{L}_8$ with $\mathrm{lde}_{\sqrt{2}}(U) = k$. U can be represented by $O(k)$ generators in $\mathcal{F}_8$ using the global synthesis algorithm.

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



Global Synthesis for $\mathcal{L}_8$

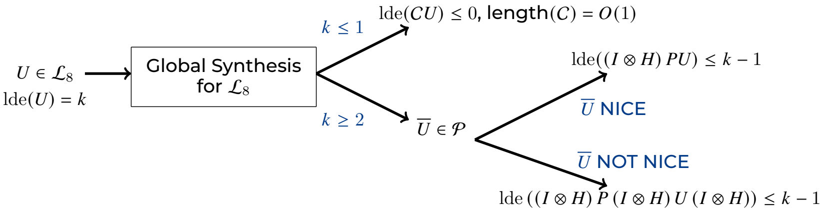
Lemma (When the Binary Pattern is NOT NICE)

Let $U \in \mathcal{L}_8$ *with* $\operatorname{lde}_{\sqrt{2}}(U) = k$. *If* $\overline{U}$ *is neither row-paired nor column-paired,* $\overline{(I \otimes H)\, U\, (I \otimes H)}$ *is row-paired and* $\operatorname{lde}_{\sqrt{2}}\big((I \otimes H)\, U\, (I \otimes H)\big) \leq \operatorname{lde}_{\sqrt{2}}(U)$.

Proposition

Let $U \in \mathcal{L}_8$ *with* $\operatorname{lde}_{\sqrt{2}}(U) = k$. U *can be represented by* $O(k)$ *generators in* $\mathcal{F}_8$ *using the global synthesis algorithm.*

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



$\square$

Pushing Hadamard through $\mathcal{G}$ (PHG)[3]

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

[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*.

Pushing Hadamard through $\mathcal{G}$ (PHG)[3]

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

$$
\begin{aligned}
(I \otimes H)(I \otimes H) &= \epsilon \\
(I \otimes H)(-1)_{[a]} &= (-1)_{[a]} X_{[a,a+1]} (-1)_{[a]} (I \otimes H) \\
(I \otimes H)(-1)_{[a]} &= X_{[a-1,a]} (I \otimes H) \\
(I \otimes H) X_{[a,a+1]} &= (-1)_{[a+1]} (I \otimes H) \\
(I \otimes H) X_{[a,a+1]} &= K_{[a-1,a,a+1,a+2]} X_{[a,a+1]} (I \otimes H)
\end{aligned}
$$

[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*.

Pushing Hadamard through $\mathcal{G}$ (PHG)[3]

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

$$\begin{aligned}
(I \otimes H)(I \otimes H) &= \epsilon \\
(I \otimes H)(-1)_{[a]} &= (-1)_{[a]} X_{[a,a+1]} (-1)_{[a]} (I \otimes H) \\
(I \otimes H)(-1)_{[a]} &= X_{[a-1,a]} (I \otimes H) \\
(I \otimes H) X_{[a,a+1]} &= (-1)_{[a+1]} (I \otimes H) \\
(I \otimes H) X_{[a,a+1]} &= K_{[a-1,a,a+1,a+2]} X_{[a,a+1]} (I \otimes H)
\end{aligned}$$

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

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

Global Synthesis for $\mathcal{O}_8$

Theorem

Let $U \in \mathcal{O}_8$ with $\mathrm{lde}(U) = k$. U can be represented by $O(k)$ generators in $\mathcal{G}_8$ using the global synthesis algorithm.

Global Synthesis for $\mathcal{O}_8$

Theorem

Let $U \in \mathcal{O}_8$ with $\operatorname{lde}(U) = k$. U can be represented by $O(k)$ generators in $\mathcal{G}_8$ using the global synthesis algorithm.

Proof Sketch. Since $U \in \mathcal{L}_8$ with $\operatorname{lde}_{\sqrt{2}}(U) = 2k$, globally synthesizing U over $\mathcal{F}_8$ yields evenly many $I \otimes H$.

Global Synthesis for $\mathcal{O}_8$

Theorem

Let $U \in \mathcal{O}_8$ with $\mathrm{lde}(U) = k$. U can be represented by $O(k)$ generators in $\mathcal{G}_8$ using the global synthesis algorithm.

Proof Sketch. Since $U \in \mathcal{L}_8$ with $\mathrm{lde}_{\sqrt{2}}(U) = 2k$, globally synthesizing U over $\mathcal{F}_8$ yields evenly many $I \otimes H$.

$U \in \mathcal{O}_8$ ⟶ Global Synthesis for $\mathcal{L}_8$ ⟶ $U = C_1(I \otimes H)C_2(I \otimes H)C_3(I \otimes H)C_4(I \otimes H)C_5$

C_1, C_2, C_3, C_4, C_5 over $\mathcal{G}_8$

length(U) = $O(k)$

Global Synthesis for $\mathcal{O}_8$

Theorem

Let $U \in \mathcal{O}_8$ with $\mathrm{lde}(U) = k$. U can be represented by $O(k)$ generators in $\mathcal{G}_8$ using the global synthesis algorithm.

Proof Sketch. Since $U \in \mathcal{L}_8$ with $\mathrm{lde}_{\sqrt{2}}(U) = 2k$, globally synthesizing U over $\mathcal{F}_8$ yields evenly many $I \otimes H$.

Global Synthesis for $\mathcal{L}_8$

$$U = C_1 (I \otimes H) C_2 (I \otimes H) C_3 (I \otimes H) C_4 (I \otimes H) C_5$$

PHG

$$U = C_1 C_2' \underline{(I \otimes H)(I \otimes H)} C_3 C_4' \underline{(I \otimes H)(I \otimes H)} C_5$$

C_1, C_2, C_3, C_4, C_5 over $\mathcal{G}_8$

length(U) = $O(k)$

C_2', C_4' over $\mathcal{G}_8$

length(C_2') / length(C_2) = $O(1)$

length(C_4') / length(C_4) = $O(1)$

Global Synthesis for $\mathcal{O}_8$

Theorem

Let $U \in \mathcal{O}_8$ with $\mathrm{lde}(U) = k$. U can be represented by $O(k)$ generators in $\mathcal{G}_8$ using the global synthesis algorithm.

Proof Sketch. Since $U \in \mathcal{L}_8$ with $\mathrm{lde}_{\sqrt{2}}(U) = 2k$, globally synthesizing U over $\mathcal{F}_8$ yields evenly many $I \otimes H$.

Global Synthesis for $\mathcal{L}_8$

$$U = C_1(I \otimes H)C_2(I \otimes H)C_3(I \otimes H)C_4(I \otimes H)C_5$$

PHG

$$U = C_1C_2'\underline{(I \otimes H)(I \otimes H)}C_3C_4'\underline{(I \otimes H)(I \otimes H)}C_5$$

$(I \otimes H)(I \otimes H) = I$

$$U = C_1C_2'C_3C_4'C_5$$

C_1, C_2, C_3, C_4, C_5 over $\mathcal{G}_8$

length(U) = $O(k)$

C_2', C_4' over $\mathcal{G}_8$

length(C_2') / length(C_2) = $O(1)$

length(C_4') / length(C_4) = $O(1)$

$\square$

Future Work

- **Explore** the complexity-theoretic properties of Toffoli-Hadamard circuits through the lens of MQCSP[4].

[4] Nai-Hui Chia et al. (2021). "Quantum meets the minimum circuit size problem". In: *arXiv preprint arXiv:2108.03171*.

Future Work

- **Explore** the complexity-theoretic properties of Toffoli-Hadamard circuits through the lens of MQCSP[4].
- **Manifest** the advantage of our global synthesis algorithm by scaling it up.

[4]Nai-Hui Chia et al. (2021). "Quantum meets the minimum circuit size problem". In: *arXiv preprint arXiv:2108.03171*.

Future Work

- **Explore** the complexity-theoretic properties of Toffoli-Hadamard circuits through the lens of MQCSP[4].
- **Manifest** the advantage of our global synthesis algorithm by scaling it up.
- **Present** the global synthesis results of $\mathcal{O}_n$ and $\mathcal{L}_n$ using $\{X, CX, CCX, K\}$ and $\{X, CX, CCX, H\}$ directly.

[4]Nai-Hui Chia et al. (2021). "Quantum meets the minimum circuit size problem". In: *arXiv preprint arXiv:2108.03171*.

Future Work

- **Explore** the complexity-theoretic properties of Toffoli-Hadamard circuits through the lens of MQCSP[4].
- **Manifest** the advantage of our global synthesis algorithm by scaling it up.
- **Present** the global synthesis results of $\mathcal{O}_n$ and $\mathcal{L}_n$ using $\{X, CX, CCX, K\}$ and $\{X, CX, CCX, H\}$ directly.
- **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.

[4] Nai-Hui Chia et al. (2021). "Quantum meets the minimum circuit size problem". In: *arXiv preprint arXiv:2108.03171*.


Thank you!
sarah.li@uwaterloo.ca