Generators and Relations for $U_n(\mathbb{Z}[\frac{1}{2},i])$

Xiaoning Bian and Peter Selinger
Dalhousie University

Consider the universal gate set for quantum computing consisting of the gates X, CX, CCX, $\omega^\dagger H$, and S. All of these gates have matrix entries in the ring $\mathbb{Z}[\frac{1}{2},i]$, the smallest subring of the complex numbers containing $\frac{1}{2}$ and i. Amy, Glaudell, and Ross proved the converse, i.e., any unitary matrix with entries in $\mathbb{Z}[\frac{1}{2},i]$ can be realized by a quantum circuit over the above gate set using at most one ancilla. In this paper, we give a finite presentation by generators and relations of $U_n(\mathbb{Z}[\frac{1}{2},i])$, the group of unitary $n \times n$-matrices with entries in $\mathbb{Z}[\frac{1}{2},i]$.

1 Introduction

It is well-known that certain useful classes of quantum circuits correspond to particular subrings of the complex numbers. For example, the well-known Clifford+T circuits, which are generated by the gates

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \omega = e^{\frac{\pi}{4} i} = \frac{1+i}{\sqrt{2}}, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & \omega \end{bmatrix}, \quad CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

can represent exactly those unitary matrices whose matrix entries are in the ring $\mathbb{Z}[1/\sqrt{2},i]$, the smallest subring of the complex numbers containing $\frac{1}{\sqrt{2}}$ and i. The left-to-right implication is obvious, since all of the above generators are matrices with entries in this ring. The converse was proved in [5], where it was shown that every unitary matrix over $\mathbb{Z}[1/\sqrt{2},i]$ can be represented as a Clifford+T circuit using at most one ancilla.

This result was extended by [1] to several other subrings of the complex numbers. In this paper, we are concerned with the ring $\mathbb{Z}[\frac{1}{2},i]$, which corresponds to the class of quantum circuits generated by the gates X, CX, CCX, K, and S. Here, X and CX are as above, CCX is the doubly-controlled X-gate, also known as the Toffoli gate, and

$$K = \omega^\dagger H = \frac{1}{1+i} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}.$$

An important problem in quantum computing is the simplification of quantum circuits. Often, the goal is to minimize the resources required for quantum computing as much as possible, for example by reducing the number of uses of certain costly gates such as the T-gate. There are many successful strategies for simplifying quantum circuits [3, 4, 11, 9, 7, 2, 12]. Some of these strategies are based on *rewriting*, which means applying algebraic laws, such as $H^2 = I$, to replace subcircuits by equivalent ones. In designing such rewriting techniques, it can be useful to have a complete set of such algebraic laws, i.e., a set of equations by which any circuit can in principle be rewritten into any equivalent circuit. While a complete set of such equations does not in itself guarantee the existence of good rewriting strategies, it can be a useful tool for designing such strategies, a method that is called "semantically guided rewriting".

M. Backens, C. Heunen (Eds.): Quantum Physics and Logic (QPL) 2021
EPTCS 343, 2021, pp. 145–164, doi:10.4204/EPTCS.343.8

In this paper, we contribute to the algebraic theory of circuits by giving a presentation of the group $U_n(\mathbb{Z}[\frac{1}{2},i])$, the group of unitary $n\times n$-matrices with entries in $\mathbb{Z}[\frac{1}{2},i]$, in terms of generators and relations. This follows earlier work by Greylyn [6], who gave a presentation for $U_4(\mathbb{Z}[1/\sqrt{2},i)$, and Li et al. [10], who gave a presentation for $U_n(\mathbb{Z}[\frac{1}{2}])$. The question of giving a presentation for $U_n(\mathbb{Z}[1/\sqrt{2},i])$ is still open for $n\geq 5$.

2 Notation and background

The concepts and results of this section are similar to those used in [5, 6, 1, 10].

2.1 Rings

As usual, we write $\mathbb{N}=\{0,1,\ldots\}$ for the set of natural numbers. We also write $\mathbb{Z}$, $\mathbb{R}$, and $\mathbb{C}$ to denote the rings of integers, real numbers, and complex numbers, respectively. We write $z^\dagger$ for the complex conjugate of a complex number. Recall that in any ring, a *unit* is an invertible element.

Let $\mathbb{Z}[i]=\{a+bi \mid a,b\in\mathbb{Z}\}$ be the ring of Gaussian integers. It is a Euclidean domain, and therefore, division with remainder, greatest common divisors, divisibility, and the concept of a prime (i.e. a non-unit that cannot be written as the product of two non-units) all can be used in $\mathbb{Z}[i]$. Let $\gamma=1+i$. It is well-known that γ is a Gaussian prime [8]. Note that γ is a divisor of 2, and γ^2 and 2 divide each other. Using Euclidean division, it is further easy to check that

$$\mathbb{Z}[i]/(\gamma)=\{0,1\},\quad \mathbb{Z}[i]/(2)=\mathbb{Z}[i]/(\gamma^2)=\{0,1,i,1+i\},\quad \text{and}\quad \mathbb{Z}[i]/(\gamma^3)=\{\pm 1,\pm i,0,1+i,1-i,2\}.$$

Definition 2.1. For $x\in\mathbb{Z}[i]$, if $x\equiv 0 \pmod{\gamma}$, we say x is *even*, otherwise *odd*.

For example, $2+3i$ and $5+2i$ are odd, and $2+4i$ and $1+3i$ are even. In particular, in our context, by an even Gaussian integer we mean one that is divisible by γ, not necessarily by 2. However if $x\in\mathbb{Z}$, then x is even (odd) as an integer if and only if x is even (odd) as a Gaussian integer. Therefore, our definition of parity extends the usual one on the integers. The following lemmas are straightforward.

Lemma 2.2. *Let $\alpha=a+bi\in\mathbb{Z}[i]$. Then α is odd if and only if $\|\alpha\|^2=a^2+b^2$ is an odd integer.*

Lemma 2.3. *If $\alpha\in\mathbb{Z}[i]$ is odd, then $\alpha\equiv\pm 1,\pm i \pmod{\gamma^3}$. In other words, $\alpha\equiv i^e \pmod{\gamma^3}$ for some $e\in\{0,1,2,3\}$. Moreover, $\alpha\equiv i^e \pmod{\gamma^2}$ for some $e\in\{0,1\}$.*

Let $\mathbb{D}=\mathbb{Z}[\frac{1}{2}]=\{\frac{a}{2^k}\mid a\in\mathbb{Z},k\in\mathbb{N}\}$ be the ring of *dyadic fractions*, i.e., fractions whose denominator is a power of 2. Consider $\mathbb{D}[i]=\{r+si\mid r,s\in\mathbb{D}\}=\mathbb{Z}[\frac{1}{2},i]$. For every $t\in\mathbb{D}[i]$, there exists some natural number k such that $\gamma^k t\in\mathbb{Z}[i]$. This motivates the following definition.

Definition 2.4. Let $t\in\mathbb{D}[i]$. A natural number $k\in\mathbb{N}$ is called a *denominator exponent* for t if $\gamma^k t\in\mathbb{Z}[i]$. The least such k is called the *least denominator exponent* of t, denoted by $\mathrm{lde}_\gamma(t)$. Equivalently, the least denominator exponent of t is the smallest $k\in\mathbb{N}$ such that t can be written in the form $\frac{s}{\gamma^k}$, where $s\in\mathbb{Z}[i]$.

More generally, we say that k is a denominator exponent for a vector or matrix if it is a denominator exponent for all of its entries. The least denominator exponent for a vector or matrix is therefore the least k that is a denominator exponent for all of its entries.

Note that for $t\in\mathbb{D}[i]$, if $k=\mathrm{lde}_\gamma(t)>0$, then $\gamma^k t$ is odd.

2.2 One- and two-level matrices

Consider complex matrices of dimension $n \times n$. We number the rows and columns of matrices starting from zero, i.e., the entries of an $n \times n$-matrix are $a_{00}, \ldots, a_{n-1,n-1}$. We define a special class of matrices called one- and two-level matrices.

Definition 2.5. Given $z \in \mathbb{C}$ and $j \in \{0, \ldots, n-1\}$, the *one-level matrix* $z_{[j]}$ is

$$z_{[j]} = \begin{array}{c} \\ \vdots \\ j \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccc} \cdots & j & \cdots \end{array} \\ \begin{bmatrix} I & 0 & 0 \\ 0 & z & 0 \\ 0 & 0 & I \end{bmatrix} \end{array},$$

i.e., the matrix that is like the $n \times n$-identity matrix, except that the entry at position (j, j) is z. Similarly, given a 2×2-matrix $U = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and $j, k \in \{0, 1, ..., n-1\}$ with $j < k$, the *two-level matrix* $U_{[j,k]}$ is

$$U_{[j,k]} = \begin{array}{c} \\ \vdots \\ j \\ \vdots \\ k \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} \cdots & j & \cdots & k & \cdots \end{array} \\ \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & a & 0 & b & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & c & 0 & d & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \end{array}.$$

Note that if $|z| = 1$, then $z_{[j]}$ is unitary; similarly, if U is unitary, then so is $U_{[j,k]}$. We say that $U_{[j,k]}$ is a two-level matrix of *type* U, and similarly, we say that $z_{[j]}$ is a one-level matrix of *type* z.

2.3 The exact synthesis algorithm

Let $U_n(\mathbb{D}[i])$ be the group of unitary matrices with entries in $\mathbb{D}[i]$. In this section, we will show that every element of $U_n(\mathbb{D}[i])$ can be written as a product of one- and two-level matrices of the form $i_{[j]}, X_{[j,k]}$, and $K_{[j,k]}$, where i is the imaginary unit and the matrices X and K are the ones that were defined in the introduction. In other words, we will show that these matrices are *generators* for the group $U_n(\mathbb{D}[i])$. We will do so by exhibiting a particular algorithm that inputs a matrix $U \in U_n(\mathbb{D}[i])$ and outputs a corresponding word in the generators. In quantum computing, such an algorithm is called an *exact synthesis algorithm* (as opposed to an approximate synthesis algorithm, which only approximates a unitary matrix up to some given ε). The algorithm in this section is adapted from [5] and [1].

Remark 2.6. A slight technical inconvenience arises because the matrix K is not self-inverse. Therefore, in the following presentation we sometimes use K as a generator and sometimes its inverse $K^\dagger$. We have carefully chosen which one to use in each instance, to make the proofs in the later parts of the paper as simple as possible. However, readers who wish to ignore the difference between K and $K^\dagger$ can safely do so, because in any case we have the relation $K^\dagger = iK$, so whatever is generated by K is also generated by $K^\dagger$ and vice versa.

We start with a number of lemmas. Since $\mathbb{Z}[i]$ and $\mathbb{D}[i]$ are subrings of $\mathbb{C}$, the usual properties of complex numbers, complex vectors, and complex matrices, such as complex conjugation, inner product, norm, and conjugate transposition are naturally inherited. For example, for a vector $v \in \mathbb{D}[i]^n$, the *norm* of v is $\|v\| = \sqrt{v^\dagger v}$, where $v^\dagger$ is the conjugate transpose of v. Note that $\|v\|^2 = v^\dagger v \in \mathbb{D}$, since $v^\dagger v \in \mathbb{D}[i] \cap \mathbb{R}$. Similarly if $v \in \mathbb{Z}[i]$, then $v^\dagger v \in \mathbb{Z}$. A *unit vector* is a vector of norm 1.

If v is a vector, the notation v_j refers to its jth entry; entries are numbered starting from $j = 0$.

Lemma 2.7. *Let v be a unit vector in $\mathbb{Z}[i]^n$. Then v has exactly one non-zero entry, and that entry is a power of i.*

Proof. First note that for $\alpha = a+bi \in \mathbb{Z}[i]$, $\|\alpha\|^2 = a^2+b^2$ is a non-negative integer, and $\|\alpha\| = 0$ if and only if $\alpha = 0$. Let $v = [v_0, v_1, ..., v_{n-1}]^T$. By assumption, $\sum_{j=0}^{n-1} \|v_j\|^2 = 1$. Since each $\|v_j\|^2$ is a non-negative integer, there is exact one non-zero v_j such that $\|v_j\| = 1$. Then it is easy to see that $v_j \in \{\pm 1, \pm i\}$. □

Corollary 2.8. *Let v be a unit vector in $\mathbb{Z}[i]^n$, and let $p \in \{0,\ldots,n-1\}$. Then there exists a matrix G that is a product of one- and two-level matrices of types i and X, such that $Gv = e_p$, where e_p is the pth standard basis vector.*

Proof. By Lemma 2.7, v has a unique non-zero entry $v_m \in \{\pm 1, \pm i\}$. Let $e \in \{0,1,2,3\}$ such that $i^e v_m = 1$. Define

$$G = \begin{cases} i^e_{[m]} & \text{if } m = p, \\ X_{[m,p]} i^e_{[m]} & \text{otherwise.} \end{cases}$$

Then $Gv = e_p$ as desired. □

Lemma 2.9. *Let v be a unit vector in $\mathbb{D}[i]^n$ with least denominator exponent $k > 0$, and let $w = \gamma^k v \in \mathbb{Z}[i]^n$. Then w has an even number of odd entries.*

Proof. We have

$$\sum_{j=0}^{n-1} \|w_j\|^2 = \|w\| = |\gamma|^{2k}\|v\| = (\gamma\gamma^\dagger)^k \equiv 0 \pmod{\gamma}.$$

Therefore, there are an even number of $j \in \{0,...,n-1\}$ such that $\|w_j\|^2$ is odd. It follows by Lemma 2.2 that an even number of w_j are odd. □

For brevity, we sometimes write $\alpha(\gamma^\ell)$ to denote any member of the congruence class of α modulo γ^ℓ. In other words, we write $\alpha(\gamma^\ell)$ for any element of $\mathbb{Z}[i]$ of the form $\alpha + \beta\gamma^\ell$, when the value of β is not important.

Lemma 2.10 (Row operation)**.** *Let $v \in \mathbb{D}[i]^n$ be a vector with $lde_\gamma(v_j) = lde_\gamma(v_\ell) = k > 0$. Then there exists an exponent $q \in \{0,1\}$ such that, if we let $v' = K^\dagger_{[j,\ell]} i^q_{[\ell]} v$, then $lde_\gamma(v'_j), lde_\gamma(v'_\ell) < k$. The remaining entries of v are unchanged.*

Proof. Let $w_j = \gamma^k v_j$ and $w_\ell = \gamma^k v_\ell$. Then both w_j and w_ℓ are odd. By Lemma 2.3, there exists $q \in \{0,1\}$ such that $w_j \equiv i^q w_\ell \pmod{\gamma^2}$. Then we have

$$K^\dagger i^q_{[1]} \begin{bmatrix} w_j \\ w_\ell \end{bmatrix} = K^\dagger \begin{bmatrix} w_j \\ i^q w_\ell \end{bmatrix} = K^\dagger \begin{bmatrix} w_j \\ w_j + \alpha\gamma^2 \end{bmatrix} = \frac{1}{\gamma^\dagger} \begin{bmatrix} 2w_j + \alpha\gamma^2 \\ -\alpha\gamma^2 \end{bmatrix} = \begin{bmatrix} (w_j + \alpha i)\gamma \\ -\alpha i \gamma \end{bmatrix} = \begin{bmatrix} 0(\gamma) \\ 0(\gamma) \end{bmatrix}.$$

This means $\text{lde}_\gamma(K^\dagger i^q_{[1]} \begin{bmatrix} v_j \\ v_\ell \end{bmatrix}) < k$. Clearly, performing $K^\dagger_{[j,\ell]} i^q_{[\ell]}$ on v has the same effect, and does not change any entries of v besides v_j and v_ℓ, proving the lemma. □

Lemma 2.11 (Column lemma)**.** *Let v be a unit vector in $\mathbb{D}[i]^n$, and $p \in \{0,1,...,n-1\}$. Then there exists a matrix G that is a product of one- and two-level matrices of types X, $K^\dagger$, and i, such that $Gv = e_p$.*

Proof. By induction on k, the least denominator exponent of v, and nested induction on the number of entries of v with least denominator exponent k. If $k = 0$, then the result follows from Corollary 2.8. Otherwise, by Lemma 2.9, v has at least two entries with least denominator exponent k. Pick a pair of such entries. Lemma 2.10 yields some one- and two level matrices that decrease this pair's least denominator exponent. The result then follows by the induction hypothesis. □

Since each column of a unitary matrix is a unit vector, the column lemma naturally gives a way to "fix" every column of a unitary matrix to the jth standard basis vector.

Lemma 2.12 (Matrix decomposition)**.** *Let U be a unitary $n \times n$-matrix with entries in $\mathbb{D}[i]$. Then there exists a matrix G that is a product of one- and two-level matrices of types X, $K^\dagger$, and i such that $GU = I$.*

Proof. This is an easy consequence of the column lemma. Specifically, first use the column lemma to find a suitable G_1 such that the rightmost column of G_1U is e_{n-1}. Because G_1U is unitary, it is of the form

$$\left[\begin{array}{c|c} U' & 0 \\ \hline 0 & 1 \end{array}\right].$$

Now recursively find one- and two-level matrices to reduce U' to the identity matrix. □

Corollary 2.13. *A matrix U with entries in $\mathbb{D}[i]$ is unitary if and only if it can be written as a product of one- and two-level matrices of types X, K, and i.*

Proof. The right-to-left implication is obvious, because the relevant one- and two-level matrices are themselves unitary. The left-to-right implication follows from Lemma 2.12. Specifically, by Lemma 2.12, we can find a product G of one- and two-level matrices of types X, $K^\dagger$, and i such that $GU = I$. Then $U = G^{-1}$. Since $X^{-1} = X$, $(K^\dagger)^{-1} = K$, and $i^{-1} = i^3$, the inverse G^{-1} can be expressed as a product of one- and two-level matrices of the required types. □

Since the proof of Lemma 2.12 is constructive, it yields an algorithm that inputs an element of $U_n(\mathbb{D}[i])$ and outputs a sequence G of one- and two-level matrices of types X, $K^\dagger$, and i such that $GU = I$. We call this an "exact synthesis algorithm". In principle, there are many different ways to achieve this; for example, one step in the proof of the column lemma requires us to "pick a pair" of entries, and the computed sequence of generators will depend on which pair we pick. For the rest of this paper, it is important to fix, once and for all, a particular deterministic method for making these choices. Therefore, we specify one such deterministic procedure in Algorithm 2.14.

In the following, by the *pivot column* of a matrix U, we mean the rightmost column where U differs from the identity matrix.

Algorithm 2.14 (Exact synthesis algorithm)**.**

INPUT: A unitary $n \times n$-matrix U with entries in $\mathbb{D}[i]$.

OUTPUT: A sequence of one- and two-level matrices $G_l, \ldots, G_1$ of types X, $K^\dagger$, and i such that $G_l \cdots G_1 U = I$.

STATE: Let M be a storage for a unitary $n \times n$-matrix, and let $\vec{G}$ be a storage for a sequence of one- and two-level matrices.

1. Set M to be U, and set $\vec{G}$ to be the empty sequence.
2. If $M = I$, stop, and output $\vec{G}$.
3. Let p be the index of the pivot column of M, and let k be the least denominator exponent of the pivot column of M.

(a) If $k = 0$, let $\vec{S}$ be obtained by applying Corollary 2.8 to the pivot column and p.

(b) If $k > 0$, Lemma 2.9 dictates there are an even number of entries that have least denominator exponent k. Let the indices of the first two odd entries be $j < \ell$. Let $\vec{S}$ be obtained by applying Lemma 2.10 to the pivot column and j, ℓ.

4. Set M to be $\vec{S}M$, and prepend $\vec{S}$ to $\vec{G}$. Go to step 2.

We refer to the $\vec{S}$ in step 3 as a *syllable*. We can also regard $\vec{G}$ as a sequence of syllables, so we can talk about the nth syllable in $\vec{G}$.

The fact that the algorithm is correct and terminating is just another way of stating Lemma 2.12. However, we will provide a more explicit proof of correctness and termination, as this will be useful later in this paper.

Definition 2.15. Let $M \in U_n(\mathbb{D}[i])$. The *level* of M, denoted by $\text{level}(M)$, is defined as follows: If $M = I$, then $\text{level}(M) = (0,0,0)$. Otherwise, $\text{level}(M)$ is a triple (p,k,m) where:

- p is the greatest index such that $Me_p \neq e_p$, i.e., the index of the pivot column.
- $k = \text{lde}(v)$, where v is the pivot column.
- m is the number of odd entries in $\gamma^k v$.

Note that p,k,m are natural numbers, so the set of all possible levels is a subset of $\mathbb{N}^3$. We use the lexicographic order on $\mathbb{N}^3$, defined by $(p,k,m) < (p',k',m')$ iff $p < p'$ or ($p = p'$ and $k < k'$) or ($p = p'$ and $k = k'$ and $m < m'$. This makes $\mathbb{N}^3$ into a well-ordered set.

Theorem 2.16. *Given a unitary matrix $U \in U_n(\mathbb{D}[i])$, Algorithm 2.14 outputs a finite sequence of one- and two-level matrices $G_l, \ldots, G_1$ such that $G_l \cdots G_1 U = I$.*

Proof. For correctness, it suffices to note that after the initialization in step 1, M and $\vec{G}$ are only updated in step 4, and at each point in the algorithm, we have $M = \vec{G}U$. The algorithm only stops when $M = I$, at which point we therefore have $I = \vec{G}U$. The only thing that remains to be shown is that the algorithm terminates.

We prove termination by well-ordered induction on the level of M. Specifically, we prove that after the algorithm reaches step 2, it always terminates. When $M = I$, this is trivially true. Otherwise, step 3 yields a syllable $\vec{S}$ such that $\text{level}(\vec{S}M) < \text{level}(M)$. Step 4 sets M to $\vec{S}M$, and loops back to step 2. At this point, since M now has a strictly smaller level, the algorithm terminates by the induction hypothesis. □

2.4 Monoid presentations

We recall some basic definitions and facts about generators and relations for monoids. A *monoid* is a set G together with an associative multiplication operation and a unit 1. A subset $H \subseteq G$ is called a *submonoid* of G if $1 \in H$ and H is closed under multiplication. Given a subset $X \subseteq G$, let $\langle X \rangle$ be the smallest submonoid of G containing X. We say that X is a set of *generators* for G if $G = \langle X \rangle$.

A *congruence* on G is an equivalence relation that is compatible with multiplication. In other words, a relation $\sim \subseteq G \times G$ is a congruence if it obeys the following rules:

$$\frac{}{w \sim w} \qquad \frac{w \sim v \quad v \sim u}{w \sim u} \qquad \frac{w \sim v}{v \sim w} \qquad \frac{w \sim w' \quad v \sim v'}{wv \sim w'v'}.$$

If $\sim$ is a congruence, then the quotient $G/\sim$ is a well-defined monoid. If $f : G \to H$ is a monoid homomorphism, then its kernel is the relation $\sim_f$ defined by $u \sim_f v$ if and only if $f(u) = f(v)$. The kernel

of every monoid homomorphism is a congruence, and conversely, every congruence $\sim$ is the kernel of some homomorphism, namely of the canonical quotient map $G \to G/\!\sim$.

Given a set X, let X^* be the set of finite sequences of elements of X, which we also call *words* over the alphabet X. Then X^* is a monoid, where multiplication is given by concatenation of words, and the unit is the empty word, which we also write as ε. By abuse of notation, we regard X as a subset of X^*, i.e., we identify each element $x \in X$ with the corresponding one-letter word $x \in X^*$. Given a set X, a *relation* is a pair (w,v) of words, where $w,v \in X^*$. Given a set $\mathscr{R}$ of relations for X, the set of consequences of $\mathscr{R}$, written $\sim_{\mathscr{R}}$, is the smallest congruence relation on X^* containing $\mathscr{R}$.

If X is a set of generators for some monoid G, and $w = x_1 \dots x_n \in X^*$ is a word, let us write $[\![w]\!] = x_1 \cdots x_n$ for the element of G obtained by multiplying $x_1, \dots, x_n$. In other words, $[\![-]\!] : X^* \to G$ is the unique monoid homomorphisms such that $[\![x]\!] = x$ for all $x \in X$. We say that a relation (w,v) is *true* in G if $[\![w]\!] = [\![v]\!]$. A set of relations $\mathscr{R}$ is *sound* for G if all of its consequences are true in G, i.e., if $w \sim_{\mathscr{R}} v \Rightarrow [\![w]\!] = [\![v]\!]$. A set of relations $\mathscr{R}$ is *complete* for G if all true relations are consequences of $\mathscr{R}$, i.e., if $[\![w]\!] = [\![v]\!] \Rightarrow w \sim_{\mathscr{R}} v$. We say that $(X,\mathscr{R})$ is a *presentation* of G (also called a presentation by generators and relations) if X is a set of generators for G and $\mathscr{R}$ is a sound and complete set of relations for G. In this case, we have $G \cong X^*/\!\sim_{\mathscr{R}}$, i.e., the monoid G is uniquely determined, up to isomorphism, by such a presentation.

Remark 2.17. To check that a set of relations $\mathscr{R}$ is sound for G, it suffices to check that $[\![w]\!] = [\![v]\!]$ holds for all $(w,v) \in \mathscr{R}$.

The monoids we consider in this paper are actually groups, i.e., all of their elements are invertible. However, when we speak of presentations, generators, relations, congruences, etc., we still mean this "in the monoid sense". In particular, we will include enough generators and relations to imply the invertibility of all elements. This differs from presentations "in the group sense", where inverses would exist automatically, and relations such as $g^{-1}g = \varepsilon$ would be unnecessary. The monoid approach is advantageous for our purposes, because it is much more straightforward to do calculations in free monoids (where the elements are words) than in free groups (where the elements are group-theoretic expressions modulo an equivalence relation).

3 A presentation of $U_n(\mathbb{D}[i])$

3.1 Statement of the main theorem

Let $\mathscr{G}$ be the set of all one- and two-level matrices of types X, K, and i. We refer to the elements of $\mathscr{G}$ as the *generators*. Specifically, they are:

- $X_{[j,\ell]}$, for $0 \le j < \ell < n$;
- $K_{[j,\ell]}$, for $0 \le j < \ell < n$; and
- $i_{[j]}$, for $0 \le j < n$.

We already saw in Corollary 2.13 that $\mathscr{G}$ indeed generates $U_n(\mathbb{D}[i])$. The purpose of this paper is to prove the following theorem:

Theorem 3.1 (Main theorem)**.** *Let $\mathscr{R}$ be the set of relations shown in Figure 1. Then $(\mathscr{G},\mathscr{R})$ is a presentation of $U_n(\mathbb{D}[i])$. In other words, the relations in Figure 1 are sound and complete for $U_n(\mathbb{D}[i])$.*

In Figure 1, relations (1)–(3) give the order of the generators. They also ensure that all of the generators (and therefore all words of generators) are invertible. Relations (4)–(9) state that generators with

$$i_{[j]}^4 \sim \varepsilon \quad (1)$$
$$X_{[j,k]}^2 \sim \varepsilon \quad (2)$$
$$K_{[j,k]}^8 \sim \varepsilon \quad (3)$$

$$i_{[j]}i_{[k]} \sim i_{[k]}i_{[j]} \quad (4)$$
$$i_{[j]}X_{[k,\ell]} \sim X_{[k,\ell]}i_{[j]} \quad (5)$$
$$i_{[j]}K_{[k,\ell]} \sim K_{[k,\ell]}i_{[j]} \quad (6)$$
$$X_{[j,k]}X_{[\ell,m]} \sim X_{[\ell,m]}X_{[j,k]} \quad (7)$$
$$X_{[j,k]}K_{[\ell,m]} \sim K_{[\ell,m]}X_{[j,k]} \quad (8)$$
$$K_{[j,k]}K_{[\ell,m]} \sim K_{[\ell,m]}K_{[j,k]} \quad (9)$$

$$i_{[k]}X_{[j,k]} \sim X_{[j,k]}i_{[j]} \quad (10)$$
$$X_{[k,\ell]}X_{[j,k]} \sim X_{[j,k]}X_{[j,\ell]} \quad (11)$$
$$X_{[j,\ell]}X_{[k,\ell]} \sim X_{[k,\ell]}X_{[j,k]} \quad (12)$$
$$K_{[k,\ell]}X_{[j,k]} \sim X_{[j,k]}K_{[j,\ell]} \quad (13)$$
$$K_{[j,\ell]}X_{[k,\ell]} \sim X_{[k,\ell]}K_{[j,k]} \quad (14)$$

$$K_{[j,k]}i_{[k]}^2 \sim X_{[j,k]}K_{[j,k]} \quad (15)$$
$$K_{[j,k]}i_{[k]}^3 \sim i_{[k]}K_{[j,k]}i_{[k]}K_{[j,k]} \quad (16)$$
$$K_{[j,k]}i_{[j]}i_{[k]} \sim i_{[j]}i_{[k]}K_{[j,k]} \quad (17)$$
$$K_{[j,k]}^2 i_{[j]}i_{[k]} \sim \varepsilon \quad (18)$$
$$K_{[j,k]}K_{[\ell,m]}K_{[j,\ell]}K_{[k,m]} \sim K_{[j,\ell]}K_{[k,m]}K_{[j,k]}K_{[\ell,m]} \quad (19)$$

Figure 1: A sound and complete set of relations for $U_n(\mathbb{D}[i])$. In each relation, the indices are assumed to be distinct; moreover, whenever a generator $X_{[a,b]}$ or $K_{[a,b]}$ is mentioned, we assume $a < b$.

disjoint indices commute. Relations (10)–(13) state that $X_{[j,k]}$ can be used to swap indices j and k in any other generator. Finally, relations (15)–(19) state additional properties of the generators. We note that the relations in Figure 1 are not minimal; for example, (1) and (18) imply (3). However, we think that they are a "nice" set of relations.

Remark 3.2. To prove the soundness part of Theorem 3.1, by Remark 2.17, it suffices to check that each relation in $\mathscr{R}$ is true in $U_n(\mathbb{D}[i])$. This can be verified by calculation. The remainder of this paper is devoted to proving completeness.

For convenience, we say that two words w,u are *relationally equivalent* if $w \sim_{\mathscr{R}} u$ and *semantically equivalent* if $[\![w]\!] = [\![u]\!]$. Completeness is thus the statement that all semantically equivalent words are relationally equivalent, and soundness is the converse.

3.2 The Cayley graph

In order to conceptualize the proof of Theorem 3.1, it is useful to define a particular kind of infinite directed graph, which we call the *Cayley graph* of $U_n(\mathbb{D}[i])$. The vertices of the Cayley graph are the elements of $U_n(\mathbb{D}[i])$, i.e., unitary matrices with entries in $\mathbb{D}[i]$. The identity matrix I plays a special role and we call it the *root* of the graph. We often use letters such as s,r,t,q to denote vertices of the Cayley graph. Our Cayley graph contains two different kinds of edges:

- *Simple edges.* Simple edges are labelled by generators $G \in \mathscr{G}$ and denoted by single arrows $\xrightarrow{G}$. There is an edge $s \xrightarrow{G} t$ if and only if $Gs = t$. For the latter equation to make sense, keep in mind that G, s, and t are all elements of $U_n(\mathbb{D}[i])$.
- *Normal edges.* Recall from Section 2.3 that each word $\vec{S}$ that is produced in step 3 of Algorithm 2.14 is called a *syllable*. From now on, we often write N to denote a single such syllable

(which may, however, be a word consisting of more than one generator). Normal edges are labelled by syllables, and are denoted by double arrows $\overset{N}{\Rightarrow}$. For every non-root vertex s of the Cayley graph, there is a unique normal edge originating at s, and it is given by $s \overset{N}{\Rightarrow} t$, where N is the syllable that Algorithm 2.14 produces when $M = s$, and $t = \vec{S}s$.

Note that as a consequence of these definitions, the Cayley graph has a tree structure with respect to the normal edges. Specifically, for every vertex s, there exists a unique path of normal edges starting at s. Also note that if $s \overset{N}{\Rightarrow} t$, then $\text{level}(t) < \text{level}(s)$, so the path of normal edges starting at any given vertex s is necessarily finite. Since the root I is the only vertex with no outgoing normal edge, every such path therefore necessarily ends at I, i.e., it is of the form $s = s_0 \overset{N_1}{\Rightarrow} s_1 \overset{N_2}{\Rightarrow} \dots \overset{N_m}{\Rightarrow} s_m = I$, for $m \geq 0$.

Definition 3.3. Given any matrix $U \in U_n(\mathbb{D}[i])$, its *normal word* is the word $w \in \mathscr{G}^*$ defined as follows: Let $U = s_0 \overset{N_1}{\Rightarrow} s_1 \overset{N_2}{\Rightarrow} \dots \overset{N_m}{\Rightarrow} s_m = I$ be the unique path of normal edges from U to I in the Cayley graph. Then we define w to be the concatenation $N_m \cdots N_1$, where each occurrence of $K^\dagger$ is replaced by K^7. Equivalently, w is the word output by Algorithm 2.14 on input U. Note that, by definition of the Cayley graph or equivalently by Theorem 2.16, we have $[\![w]\!]U = I$, or in other words $[\![w]\!] = U^{-1}$. We write $w = \text{normal}(U)$ when w is the normal word of U.

Moreover, given any word $u \in \mathscr{G}^*$, not necessarily normal, define its *normal form* to be $\text{normal}([\![u]\!]^{-1})$. Note that by definition, every word u has a unique normal form, and if w is the normal form of u, then $[\![w]\!] = [\![u]\!]$. Moreover, if $[\![u]\!] = [\![u']\!]$ then u and u' have the same normal form. We write $\text{NF}(u)$ for the normal form of a word u.

Given a path $\vec{G} = s_0 \xrightarrow{G_1} s_1 \xrightarrow{G_2} s_2 \dots s_{n-1} \xrightarrow{G_n} s_n$ of simple edges in the Cayley graph, we define the *level* of the path by $\text{level}(\vec{G}) = \max\{\text{level}(s_i) \mid i = 0, \dots, n\}$. Here, the level of a vertex is of course as defined in Definition 2.15.

3.3 Basic generators

To simplify the proof of completeness, it will sometimes be useful to consider a smaller set of generators for $U_n(\mathbb{D}[i])$, which we call the *basic generators*. They are the following:

- $X_{[j,j+1]}$, for $0 \leq j < n-1$;
- $K_{[0,1]}$; and
- $i_{[0]}$.

We also call the corresponding edges of the Cayley graph *basic edges*.

Lemma 3.4. *Each generator $G \in \mathscr{G}$ can be written as a product of basic generators by means of repeated applications of the following relations:*

$$\begin{aligned}
i_{[j]} &\sim_{\mathscr{R}} X_{[0,j]} i_{[0]} X_{[0,j]} && \text{for } j > 1,\\
K_{[j,\ell]} &\sim_{\mathscr{R}} X_{[0,j]} K_{[0,\ell]} X_{[0,j]} && \text{for } j > 0,\\
K_{[0,\ell]} &\sim_{\mathscr{R}} X_{[1,\ell]} K_{[0,1]} X_{[1,\ell]} && \text{for } \ell > 1,\\
X_{[j,\ell]} &\sim_{\mathscr{R}} X_{[j,j+1]} X_{[j+1,\ell]} X_{[j,j+1]} && \text{for } \ell > j+1.
\end{aligned}$$

Proof. By case distinction. For example, by repeated application of these relations, we have:

$$K_{[2,4]} \sim_{\mathscr{R}} X_{[0,1]}X_{[1,2]}X_{[0,1]}X_{[1,2]}X_{[2,3]}X_{[3,4]}X_{[2,3]}X_{[1,2]}K_{[0,1]}X_{[1,2]}X_{[2,3]}X_{[3,4]}X_{[2,3]}X_{[1,2]}X_{[0,1]}X_{[1,2]}X_{[0,1]}. \qquad \square$$

Moreover, we will use the following fact: if $s \xrightarrow{G} r$ is a simple edge of the Cayley graph and $s \xrightarrow{G_1} s_1 \xrightarrow{G_2} \dots \xrightarrow{G_m} r$ is the corresponding sequence of basic edges obtained from the above relations, then $\text{level}(s_1),\dots,\text{level}(s_{m-1}) \leq \max(\text{level}(s),\text{level}(r))$. In other words, the conversion to basic generators does not increase the level.

3.4 Reduction of completeness to the Main Lemma

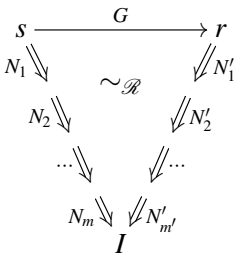
We now outline our strategy for proving completeness. For pedagogical reasons, the following lemmas are stated in the opposite order in which they are proved, i.e., each lemma implies the one before it. Completeness is a direct consequence of the following lemma.

Lemma 3.5. *Every word is relationally equivalent to its normal form.*

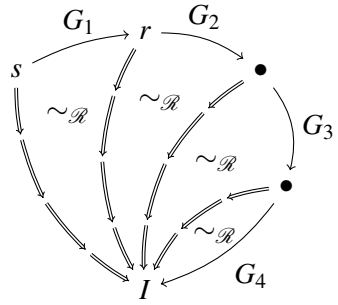
To see why the lemma implies completeness, let v,u be any two words such that $[\![v]\!] = [\![u]\!]$. Then v and u have the same normal form, say w. By the lemma, $v \sim_{\mathscr{R}} w \sim_{\mathscr{R}} u$, from which completeness follows by transitivity.

We will prove Lemma 3.5 by induction on the length of the word u. Since by Lemma 3.4, each generator is relationally equivalent to a word of basic generators, we can assume without loss of generality that u consists of basic generators. For the induction step, we use the following lemma:

Lemma 3.6. *Consider any basic edge $s \xrightarrow{G} r$ of the Cayley graph. Let $w = \text{normal}(s)$ and $u = \text{normal}(r)$. Then $w \sim_{\mathscr{R}} uG$. Or in pictures, the following diagram commutes relationally:*



To see why Lemma 3.6 implies Lemma 3.5, consider any word u composed of basic generators. If $u = \varepsilon$, then u is already normal so there is nothing to show. Otherwise, $u = u'G$ for some generator G. Let $s = [\![u]\!]^{-1}$ and $r = [\![u']\!]^{-1}$; then $s \xrightarrow{G} r$ is a basic edge of the Cayley graph. By Lemma 3.6, we have $\text{normal}(r)\,G \sim_{\mathscr{R}} \text{normal}(s)$. Also, by the induction hypothesis, since u' is a shorter word than u, we have $u' \sim_{\mathscr{R}} \text{NF}(u')$. Then the claim follows because $u = u'G \sim_{\mathscr{R}} \text{NF}(u')\,G = \text{normal}(r)\,G \sim_{\mathscr{R}} \text{normal}(s) = \text{NF}(u)$. The following diagram illustrates the proof in case $u = G_4G_3G_2G_1$ is a word of length 4.

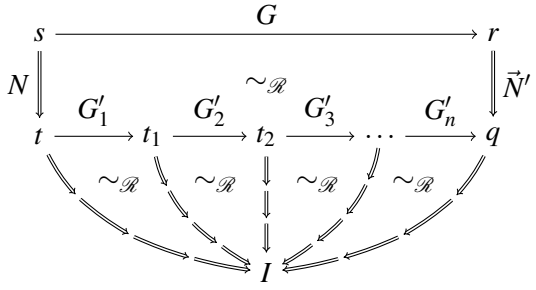


So now, what is left to do is to prove Lemma 3.6. We will prove this by induction on the level of s. The induction step uses the following lemma.

Lemma 3.7 (Main Lemma)**.** *Assume $s \xrightarrow{G} r$ is a basic edge of the Cayley graph, and $s \overset{N}{\Rightarrow} t$ is a normal edge. Then there exists a sequence of normal edges $r \overset{\vec{N}'}{\Longrightarrow} q$ and a sequence of basic edges $t \xrightarrow{\vec{G}'} q$ such that $\vec{N}'G \sim_{\mathscr{R}} \vec{G}'N$ and $level(\vec{G}') < level(s)$. Here is the relation $\vec{N}'G \sim_{\mathscr{R}} \vec{G}'N$ as a diagram:*

$$\begin{array}{ccc} s & \xrightarrow{G} & r \\ {\scriptstyle N}\Downarrow & \sim_{\mathscr{R}} & \Downarrow{\scriptstyle \vec{N}'} \\ t & \xrightarrow{\vec{G}'} & q. \end{array} \tag{20}$$

The proof that Lemma 3.7 implies Lemma 3.6 proceeds by induction on the level of s. The base case arises when $s = I$. In that case, an easy case distinction shows that the claim holds for all generators G. For the induction step, we have $s \neq I$, so there exists a unique normal edge $s \overset{N}{\Rightarrow} t$. By Lemma 3.7, there exists a sequence of normal edges $r \overset{\vec{N}'}{\Longrightarrow} q$ and a sequence of basic edges $t \xrightarrow{\vec{G}'} q$ such that (20) holds and $\mathrm{level}(\vec{G}') < \mathrm{level}(s)$. Since the level of each vertex occurring in the path $t \xrightarrow{\vec{G}'} q$ is strictly less than the level of s, by the induction hypothesis, the claim of Lemma 3.6 is already true for each edge in this path. Then the lemma follows by the following diagram; note that each part commutes relationally and therefore so does the whole diagram.



Given the above sequence of lemmas, to finish the completeness proof, all that is now left to do is to prove Lemma 3.7. Since the proof is a rather long and tedious case distinction, it can be found in the Appendix.

4 Conclusion and future work

We have given a presentation by generators and relations of the group of unitary $n \times n$-matrices with entries in the ring $\mathbb{D}[i] = \mathbb{Z}[\frac{1}{2}, i]$. This matrix group has some applications in quantum computing because it arises as the group of unitary operations that are exactly representable by a certain gate set that is a subset of the Clifford+T circuits; namely, it is generated by the X, controlled-X, Toffoli and S gates along with a modified Hadamard gate $K = \omega^{\dagger} H$. We have described this group in terms of generators that are 1- and 2-level matrices. We do not yet have a complete set of relations for the quantum circuits generated by the above gates, and doing so is a non-trivial problem because quantum circuits are described in terms of the tensor operation $\otimes$ and not the direct sum of vector spaces.

While other subgroups of the Clifford+T group have been described by generators and relations [10], this has not yet been done for the Clifford+T group itself, except in the case of matrices of size 4×4 [6]. Doing so would require extending our results from the ring $\mathbb{D}[i]$ to the ring $\mathbb{D}[\omega]$. This problem turns out to be harder than one would expect, because as the complexity of the ring increases, it becomes more and more difficult to complete all the cases of the Main Lemma. This is left for future work.

References

[1] Matthew Amy, Andrew N. Glaudell & Neil J. Ross (2020): *Number-theoretic characterizations of some restricted Clifford+T circuits*. Quantum 4, p. 252, doi:10.22331/q-2020-04-06-252. Also available from arXiv:1908.06076.

[2] Matthew Amy & Michele Mosca (2019): *T-count optimization and Reed-Muller codes*. IEEE Transactions on Information Theory 65(8), pp. 4771–4784, doi:10.1109/TIT.2019.2906374. Also available from arXiv:1601.07363.

[3] Niel de Beaudrap, Xiaoning Bian & Quanlong Wang (2020): *Fast and effective techniques for T-count reduction via spider nest identities*. In Steven T. Flammia, editor: 15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020), Leibniz International Proceedings in Informatics (LIPIcs) 158, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 11:1–11:23, doi:10.4230/LIPIcs.TQC.2020.11. Also available from arXiv:2004.05164.

[4] Niel de Beaudrap, Xiaoning Bian & Quanlong Wang (2020): *Techniques to reduce $\pi/4$-parity-phase circuits, motivated by the ZX calculus*. In: Proceedings of the 16th International Conference on Quantum Physics and Logic, QPL 2019, Electronic Proceedings in Theoretical Computer Science 318, p. 131–149, doi:10.4204/eptcs.318.9. Also available from arXiv:1911.09039.

[5] Brett Giles & Peter Selinger (2013): *Exact synthesis of multiqubit Clifford+T circuits*. Physical Review A 87, p. 032332 (7 pages), doi:10.1103/PhysRevA.87.032332. Also available from arXiv:1212.0506.

[6] Seth E. M. Greylyn (2014): *Generators and relations for the group $U_4(\mathbb{Z}[\frac{1}{\sqrt{2}}, i])$*. M.Sc. thesis, Dalhousie University. Available from arXiv:1408.6204.

[7] Luke E. Heyfron & Earl T. Campbell (2018): *An efficient quantum compiler that reduces T count*. Quantum Science and Technology 4(1), p. 015004, doi:10.1088/2058-9565/aad604. Also available from arXiv:1712.01557.

[8] Kenneth Ireland & Michael Rosen (1982): *A Classical Introduction to Modern Number Theory*. Graduate Texts in Mathematics 84, Springer, doi:10.1007/978-1-4757-2103-4.

[9] Aleks Kissinger & John van de Wetering (2020): *Reducing the number of non-Clifford gates in quantum circuits*. Phys. Rev. A 102, p. 022406, doi:10.1103/PhysRevA.102.022406.

[10] Sarah Meng Li, Neil J. Ross & Peter Selinger (2021): *Generators and relations for the group $O_n(\mathbb{Z}[1/2])$*. In: Proceedings of the 18th International Conference on Quantum Physics and Logic, QPL 2021, Electronic Proceedings in Theoretical Computer Science. Also available from arXiv:2106.01175.

[11] Yunseong Nam, Neil J. Ross, Yuan Su, Andrew M. Childs & Dmitri Maslov (2018): *Automated optimization of large quantum circuits with continuous parameters*. Npj Quantum Information 4(1), doi:10.1038/s41534-018-0072-4. Also available from arXiv:1710.07345.

[12] Fang Zhang & Jianxin Chen (2019): *Optimizing T gates in Clifford+T circuit as $\pi/4$ rotations around Paulis*. Available from arXiv:1903.12456.

A Appendix: Proof of the Main Lemma

Before we prove Lemma 3.7, we collect a number of useful consequences of the relations from Figure 1. These are shown in Figure 2. The proofs of these relations are straightforward.

To keep the proof of Lemma 3.7 as readable as possible, we make the following simplification. Each time we complete the diagram (20), we will permit $\vec{G}'$ to be a sequence of simple edges, rather than basic edges as required by the lemma. This is justified because each such sequence of simple edges can be expanded into a (usually much longer) sequence of basic edges whose level is no higher than that of the original sequence.

With that in mind, we now proceed to prove Lemma 3.7 by case distinction. Assume $s \xrightarrow{G} r$ is a basic edge of the Cayley graph, and $s \overset{N}{\Rightarrow} t$ is a normal edge. Let $L = (p,k,m)$ be the level of s; specifically, p is the index of the pivot column, k is the least denominator exponent of the pivot column, m is the number of odd entries in $\gamma^k v$, where v is the pivot column. Also let $w = \gamma^k v$.

Case 1. $G = i_{[0]}$. Let j be the index of the first odd entry of w.

Case 1.1. $j > 0$. Note that $0 < j \leq p$. Then the normal edge N does not act on row 0, and N is still the normal edge for state r. We complete the diagram as follows.

$$\begin{array}{ccc} s & \xrightarrow{i_{[0]}} & r \\ {\scriptstyle N}\Big\Downarrow & & \Big\Downarrow{\scriptstyle N} \\ t & \xrightarrow{i_{[0]}} & q \end{array}$$

The diagram commutes relationally by (4)–(9), which ensure that disjoint generators commute. We will encounter additional cases in which the states s and r generate the same syllable N and the indices that N acts on are disjoint from those of G. We will refer to such cases as "disjoint" cases.

Case 1.2. $j = 0$, $k = 0$, and $p = 0$. Since the jth entry of v is odd, by Lemma 2.7, it must be of the form i^e for some $e \in \{0,\ldots,3\}$. Note that $e = 0$ is not possible, because then v would be e_p and would not be a pivot column. Therefore $e > 0$. In that case, the normal edge from s is $i_{[0]}^{4-e}$ and we complete the diagram as follows.

$$\begin{array}{ccc} s & \xrightarrow{i_{[0]}} & r \\ {\scriptstyle i_{[0]}^{4-e}}\Big\Downarrow & & \Big\Downarrow{\scriptstyle i_{[0]}^{3-e}} \\ t & \xrightarrow[\varepsilon]{} & t \end{array}$$

Note that here, $\xrightarrow{\varepsilon}$ denotes a path of length 0. Also, $\overset{i_{[0]}^{3-e}}{\Longrightarrow}$ denotes a path of length 0 if $e = 3$ and a path of length 1 otherwise. The diagram commutes relationally by reflexivity.

Case 1.3. $j = 0$, $k = 0$, and $p > 0$. In this case, the exact synthesis algorithm specifies that the normal edge from s is $X_{[0,p]} i_{[0]}^{-e}$, and the normal edge from r is $X_{[0,p]} i_{[0]}^{-e-1}$. Here and from now on, we will tacitly understand all exponents of $i_{[j]}$ to be taken modulo 4, which is justified by relation (1). Similarly, from now on we will also tacitly use relations (2) and (3) to invert the X and K generators when appropriate.

$$K^\dagger_{[j,\ell]} i_{[j]} \sim_{\mathscr{R}} i_{[j]} i_{[\ell]} X_{[j,\ell]} K^\dagger_{[j,\ell]} i_{[\ell]} \tag{21}$$

$$K_{[j,\ell]} \sim_{\mathscr{R}} i^3_{[j]} i^3_{[\ell]} K^\dagger_{[j,\ell]} \tag{22}$$

$$K^\dagger_{[j,\ell]} i_{[\ell]} K_{[j,\ell]} \sim_{\mathscr{R}} i^3_{[\ell]} X_{[j,\ell]} K^\dagger_{[j,\ell]} i_{[\ell]} \tag{23}$$

$$X_{[j,\ell]} i^q_{[j]} X_{[k,\ell]} \sim_{\mathscr{R}} X_{[j,k]} X_{[j,\ell]} i^q_{[j]} \tag{24}$$

$$X_{[k,\ell]} i^q_{[k]} X_{[j,k]} \sim_{\mathscr{R}} X_{[j,k]} X_{[j,\ell]} i^q_{[j]} \tag{25}$$

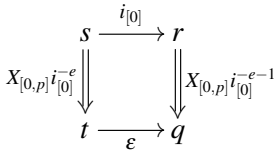
$$K_{[j,\ell]} i^q_{[\ell]} X_{[k,\ell]} \sim_{\mathscr{R}} X_{[k,\ell]} K_{[j,k]} i^q_{[k]} \tag{26}$$

$$K^\dagger_{[\ell,\ell']} K^\dagger_{[j,j']} K^\dagger_{[j',\ell']} K^\dagger_{[j,\ell]} X_{[\ell,j']} \sim_{\mathscr{R}} X_{[\ell,j']} K^\dagger_{[\ell,\ell']} K^\dagger_{[j,j']} K^\dagger_{[j',\ell']} K^\dagger_{[j,\ell]} \tag{27}$$

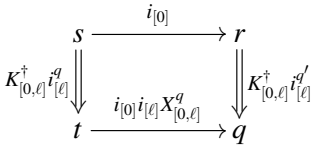
$$K^\dagger_{[j,\ell]} i_{[\ell]} X_{[j,\ell]} \sim_{\mathscr{R}} X_{[j,\ell]} i^3_{[j]} i_{[\ell]} K^\dagger_{[j,\ell]} i_{[\ell]} \tag{28}$$

Figure 2: Some useful relations in $\sim_{\mathscr{R}}$. All of these are consequences of the relations in Figure 1. As before, in each relation, the indices are assumed to be distinct, and when a generator $X_{[a,b]}$ or $K_{[a,b]}$ is mentioned, we assume $a < b$. $K^\dagger$ abbreviates K^7.

We complete the diagram as follows.



Case 1.4. $j = 0$ and $k > 0$. By Lemma 2.9, w has an even number of odd entries. Let ℓ be the index of the second odd entry of w. In this case, the exact synthesis algorithm specifies that the normal edge from s is $K^\dagger_{[0,\ell]} i^q_{[\ell]}$ and the normal edge from r is $K^\dagger_{[0,\ell]} i^{q'}_{[\ell]}$, for $q, q' \in \{0,1\}$ and $q' \neq q$. We complete the diagram as follows:



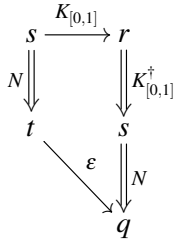
This diagram commutes relationally by (17) when $q = 0$ and by (21) when $q = 1$.

Case 2. $G = K_{[0,1]}$.

Case 2.1. $k = 0$. Let j be the index of the first odd entry of w.

Case 2.1.1. $j < 2$. In this case, the exact synthesis algorithm specifies that the normal edge from r is

$K^\dagger_{[0,1]}$. We complete the diagram as follows, and it commutes relationally by (3).

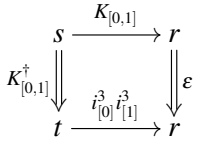


We will encounter additional cases in which the normal edge from r is relationally the inverse of G. In such cases, the diagram can always be completed in the same way. We refer to these cases as "retrograde".

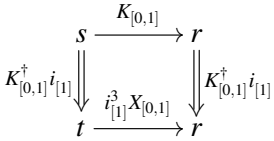
Case 2.1.2. $j \geq 2$. Note that $v_0 = v_1 = 0$. Here, and from now on, we write v_j for the jth component of a vector v. Since $j \leq p$, the normal edges from both s and r are $X_{[j,p]} i^{-e}_{[j]}$, which are disjoint from $K_{[0,1]}$. This is a disjoint case.

Case 2.2. $k > 0$. By Lemma 2.9, w has an even number of odd entries. Let j and ℓ be the indices of the first two odd entries of w.

Case 2.2.1. $j = 0$ and $\ell = 1$. In this case, the exact synthesis algorithm specifies that the normal edge from s is $K^\dagger_{[0,1]} i^q_{[1]}$. If $q = 0$, then $\mathrm{level}(r) < \mathrm{level}(s)$, and we complete the diagram as follows. It commutes relationally by (22).



If $q = 1$, then the exact synthesis algorithm specifies that the normal edge from r is also $K^\dagger_{[0,1]} i_{[1]}$. In this case, we complete the diagram as follows. It commutes relationally by (23).



Case 2.2.2. $j = 0$ and $\ell > 1$. Note that $j < \ell \leq p$, so the first two entries in each column after the pivot column are 0, hence $K_{[0,1]}$ does not change p. But it will increase the least denominator exponent of the pivot column from k to $k+1$. The exact synthesis algorithm then specifies that the normal edge from r is $K^\dagger_{[0,1]}$, so this case is retrograde.

Case 2.2.3. $j = 1$. This is similar to the previous case. Again, $K_{[0,1]}$ increases the denominator exponent of the pivot column, the normal edge is $K^\dagger_{[0,1]}$, and so the case is retrograde.

Case 2.2.4. $j > 1$. In this case, $\mathrm{lde}(v_0, v_1) < k$. Note that $j < \ell \leq p$, so the first two entries in each column after the pivot column are 0, hence $K_{[0,1]}$ does not change p. The exact synthesis algorithm specifies that the normal edge from s is $K^\dagger_{[j,\ell]} i^q_{[\ell]}$. Let v' be the pivot column of r.

If $\mathrm{lde}(v'_0,v'_1) < k$, then the normal edge from r is also $K^{\dagger}_{[j,\ell]}i^{q}_{[\ell]}$ and the case is disjoint. On the other hand, if $\mathrm{lde}(v'_0,v'_1) = k$. Let $w' = \gamma^k v'$. One can show that in this case, $w'_0 \equiv w'_1 \pmod{\gamma^2}$, and therefore the normal edge from r is $K^{\dagger}_{[0,1]}$. Then this case is retrograde.

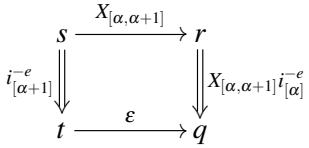
Case 3. $G = X_{[\alpha,\alpha+1]}$.

Case 3.1. $k = 0$. Let j be the index of the first odd entry of v. Note that $j \leq p$. Also, by Lemma 2.7, the jth entry of v is of the form i^e for some $e \in \{0,\ldots,3\}$.

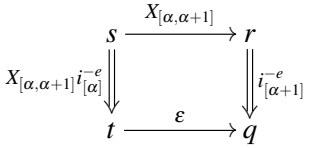
Case 3.1.1. $\alpha \geq p$. Applying $X_{[\alpha,\alpha+1]}$ increases p, and the exact synthesis algorithm specifies that the normal edge from r is $X_{[\alpha,\alpha+1]}$. Hence this case is retrograde using (2).

Case 3.1.2. $\alpha = p-1$.

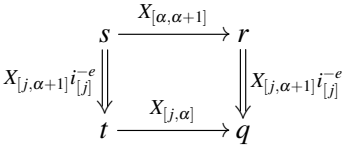
Case 3.1.2.1. $j = \alpha+1$. Note that $e = 0$ is not possible, because then v would be e_p and would not be a pivot column. Therefore $e > 0$. The exact synthesis algorithm specifies that the normal edge from s is $i^{-e}_{[\alpha+1]}$ and the normal edge from r is $X_{[\alpha,\alpha+1]}i^{-e}_{[\alpha]}$. We complete the diagram as follows, and it commutes relationally by (10) and (2).



Case 3.1.2.2. $j = \alpha$. The exact synthesis algorithm specifies $X_{[\alpha,\alpha+1]}i^{-e}_{[\alpha]}$ and $i^{-e}_{[\alpha+1]}$ as the normal edges from s and r, respectively. We complete the diagram as follows, and it commutes relationally by (10).



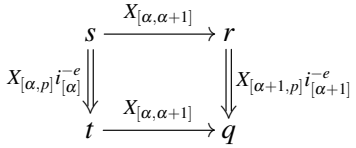
Case 3.1.2.3. $j \leq \alpha-1$. The exact synthesis algorithm specifies that the normal edge from both s and r is $X_{[j,\alpha+1]}i^{-e}_{[j]}$. We complete the diagram as follows. It commutes relationally by (24).



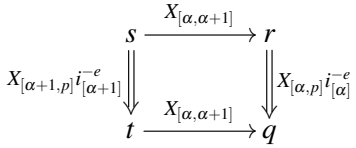
Case 3.1.3. $\alpha \leq p-2$.

Case 3.1.3.1. $j = \alpha$. The exact synthesis algorithm specifies that the normal edge from s is $X_{[\alpha,p]}i^{-e}_{[\alpha]}$ and the normal edge from r is $X_{[\alpha+1,p]}i^{-e}_{[\alpha+1]}$ We complete the diagram as follows. It commutes relationally

by (25).



Case 3.1.3.2. $j = \alpha + 1$. The exact synthesis algorithm specifies that the normal edge from s is $X_{[\alpha+1,p]}i^{-e}_{[\alpha+1]}$ and the normal edge from r is $X_{[\alpha,p]}i^{-e}_{[\alpha]}$ We complete the diagram as follows. It commutes relationally by (25) and (2).



Case 3.1.3.3. $j \neq \alpha$ and $j \neq \alpha + 1$. In this case, both normal edges are $X_{[j,p]}i^{-e}_{[j]}$. This case is disjoint.

Case 3.2. $k > 0$. By Lemma 2.9, w has an even number of odd entries. Let j and ℓ be the indices of the first two odd entries of w.

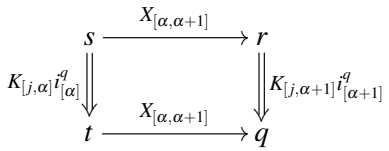
Case 3.2.1. $\alpha \geq p$. Applying $X_{[\alpha,\alpha+1]}$ increases p, and the normal edge from r is $X_{[\alpha,\alpha+1]}$. Therefore this case is retrograde.

Case 3.2.2. $\alpha < p$. We will do a case distinction on how $j < \ell$ overlaps with $\alpha < \alpha + 1$.

Case 3.2.2.1. $\ell < \alpha$. The normal edge from both s and r is $K^{\dagger}_{[j,\ell]}i^{q}_{[\ell]}$, so this case is disjoint.

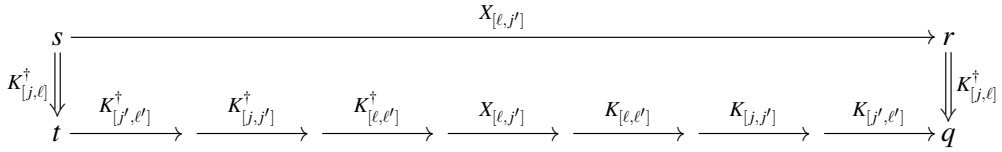
Case 3.2.2.2. $\ell = \alpha$. The exact synthesis algorithm specifies that the normal edge from s is $K^{\dagger}_{[j,\alpha]}i^{q}_{[\alpha]}$, for some $q \in \{0,1\}$.

If $w_{\alpha+1}$ is even, then the normal edge from r will be $K^{\dagger}_{[j,\alpha+1]}i^{q}_{[\alpha+1]}$. In this case, we can complete the diagram as follows. It commutes relationally by (26).



Now assume that $w_{\alpha+1}$ is odd. By Lemma 2.9, we have a fourth odd entry. Let $j' = \alpha + 1$ and ℓ' be the index of the fourth odd entry. By Lemma 2.3, we have $w_j = i^e + a\omega^3$, $w_\ell = i^f + b\omega^3$, $w_{j'} = i^g + c\omega^3$, and $w_{\ell'} = i^h + d\omega^3$, for some $e, f, g, h \in \{0, \ldots, 3\}$ and $a, b, c, d \in \mathbb{Z}[i]$.

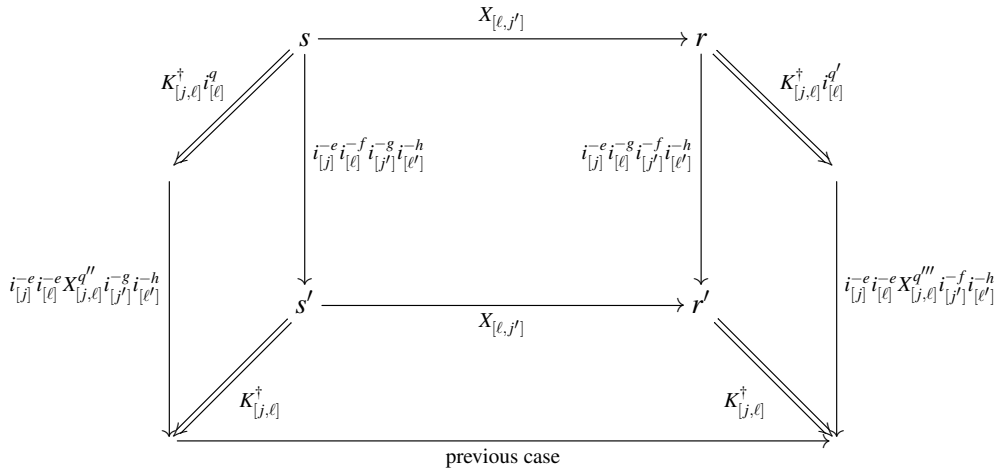
Case 3.2.2.2.1. $e = f = g = h = 0$. In this special case, the normal edges from s and r are both $K^{\dagger}_{[j,\ell]}$. We complete the diagram as follows.



The fact that this diagram commutes relationally follows from (27). We must verify that it satisfies the level condition. The following diagram shows only entries j,ℓ,j',ℓ' of γ^k times the pth column of each state. Since in all 8 states below the top row, at least two entries are even, the level of all of these states is strictly less than that of s.

$$\begin{array}{ccc}
\begin{pmatrix} 1+a\gamma^3 \\ 1+b\gamma^3 \\ 1+c\gamma^3 \\ 1+d\gamma^3 \end{pmatrix} & \xrightarrow{X_{[\ell,j']}} & \begin{pmatrix} 1+a\gamma^3 \\ 1+c\gamma^3 \\ 1+b\gamma^3 \\ 1+d\gamma^3 \end{pmatrix} \\
\Big\Downarrow K^\dagger_{[j,\ell]} & & \Big\Downarrow K^\dagger_{[j,\ell]} \\
\begin{pmatrix} (1+i)+i(a+b)\gamma^2 \\ i(a-b)\gamma^2 \\ 1+c\gamma^3 \\ 1+d\gamma^3 \end{pmatrix} & & \begin{pmatrix} (1+i)+i(a+c)\gamma^2 \\ i(a-c)\gamma^2 \\ 1+b\gamma^3 \\ 1+d\gamma^3 \end{pmatrix} \\
\Big\downarrow K^\dagger_{[j',\ell']} & & \Big\uparrow K_{[j',\ell']} \\
\begin{pmatrix} (1+i)+i(a+b)\gamma^2 \\ i(a-b)\gamma^2 \\ (1+i)+i(c+d)\gamma^2 \\ i(c-d)\gamma^2 \end{pmatrix} & & \begin{pmatrix} (1+i)+i(a+c)\gamma^2 \\ i(a-c)\gamma^2 \\ (1+i)+i(b+d)\gamma^2 \\ i(b-d)\gamma^2 \end{pmatrix} \\
\Big\downarrow K^\dagger_{[j,j']} & & \Big\uparrow K_{[j,j']} \\
\begin{pmatrix} 2i-(a+b+c+d)\gamma \\ i(a-b)\gamma^2 \\ -(a+b-c-d)\gamma \\ i(c-d)\gamma^2 \end{pmatrix} & & \begin{pmatrix} 2i-(a+c+b+d)\gamma \\ i(a-c)\gamma^2 \\ -(a+c-b-d)\gamma \\ i(b-d)\gamma^2 \end{pmatrix} \\
\Big\downarrow K^\dagger_{[\ell,\ell']} & & \Big\uparrow K_{[\ell,\ell']} \\
\begin{pmatrix} 2i-(a+b+c+d)\gamma \\ -(a-b+c-d)\gamma \\ -(a+b-c-d)\gamma \\ -(a-b-c+d)\gamma \end{pmatrix} & \xrightarrow{X_{[\ell,j']}} & \begin{pmatrix} 2i-(a+c+b+d)\gamma \\ -(a-c+b-d)\gamma \\ -(a+c-b-d)\gamma \\ -(a-c-b+d)\gamma \end{pmatrix}
\end{array}$$

Case 3.2.2.2.2. Otherwise. In the general case, the normal edge from s is $K^\dagger_{[j,\ell]}i^q_{[\ell]}$, where $q=0$ if $e-f$ is even and $q=1$ if $e-f$ is odd. Similarly, the normal edge from r is $K^\dagger_{[j,\ell]}i^{q'}_{[\ell]}$, where $q'=0$ if $e-g$ is even and $q'=1$ if $e-g$ is odd. Define $q''=0$ when $-q-f+e\equiv 0 \pmod 4$ and $q''=1$ when $e-f-q\equiv 2 \pmod 4$. Similarly, define $q'''=0$ when $e-g-q'\equiv 0 \pmod 4$ and $q'''=1$ when $-q-g+e\equiv 2 \pmod 4$. We complete the diagram using the outer perimeter of the following figure. Here, the bottom edge is given as in the previous case.



To see that the outer perimeter relationally commutes, it suffices to show that the four inner faces relationally commutes. The bottom face does so by the previous case. The middle face commutes by repeated applications of (5) and (10). To see why the left face commutes, we first note that

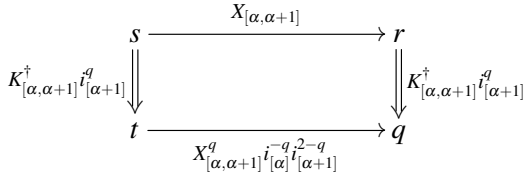
$$K^\dagger_{[j,\ell]}\, i^{e-f-q}_{[\ell]} \sim_{\mathscr{R}} X^{q''}_{[j,\ell]}\, K^\dagger_{[j,\ell]}. \tag{29}$$

Namely, this holds by (15) in case $e-f-q \equiv 2 \pmod 4$ and $q''=1$, and it holds trivially in case $e-f-q \equiv 0 \pmod 4$ and $q''=0$. Then the left face commutes because

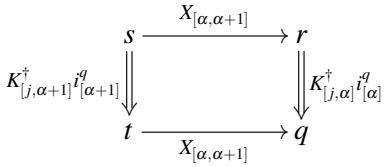
$$\begin{aligned}
K^\dagger_{[j,\ell]}\, i^{-e}_{[j]}\, i^{-f}_{[\ell]}\, i^{-g}_{[j']}\, i^{-h}_{[\ell']} &\sim_{\mathscr{R}} K^\dagger_{[j,\ell]}\, i^{-e}_{[j]}\, i^{-e}_{[\ell]}\, i^{e-f-q}_{[\ell]}\, i^{q}_{[\ell]}\, i^{-g}_{[j']}\, i^{-h}_{[\ell']} && \text{by (1)}\\
&\sim_{\mathscr{R}} i^{-e}_{[j]}\, i^{-e}_{[\ell]}\, K^\dagger_{[j,\ell]}\, i^{e-f-q}_{[\ell]}\, i^{q}_{[\ell]}\, i^{-g}_{[j']}\, i^{-h}_{[\ell']} && \text{by (17)}\\
&\sim_{\mathscr{R}} i^{-e}_{[j]}\, i^{-e}_{[\ell]}\, X^{q''}_{[j,\ell]}\, K^\dagger_{[j,\ell]}\, i^{q}_{[\ell]}\, i^{-g}_{[j']}\, i^{-h}_{[\ell']} && \text{by (29)}\\
&\sim_{\mathscr{R}} i^{-e}_{[j]}\, i^{-e}_{[\ell]}\, X^{q''}_{[j,\ell]}\, i^{-g}_{[j']}\, i^{-h}_{[\ell']}\, K^\dagger_{[j,\ell]}\, i^{q}_{[\ell]} && \text{by (4) and (6).}
\end{aligned}$$

The right face commutes for the same reason, just swapping the rules of f and g. Finally, we need to verify that the diagram satisfies the level condition. To this end, note that s, r, s', and r' all have the same level, because the operations $X_{[\ell,j']}$ and $i^{-e}_{[j]}\, i^{-f}_{[\ell]}\, i^{-g}_{[j']}\, i^{-h}_{[\ell']}$ neither change the pivot column, the denominator exponent, nor the number of odd entries. Together with the fact that normal edges are level decreasing and with what was shown in the previous case, this implies the level condition.

Case 3.2.2.3. $\ell = \alpha+1$ and $j=\alpha$. In this case, the exact synthesis algorithm prescribes that the normal edges from both s and r are $K^\dagger_{[\alpha,\alpha+1]} i^q_{[\alpha+1]}$, for $q \in \{0,1\}$. We complete the diagram as follows. It commutes relationally by the inverse of (15) when $q=0$ and by (28) when $q=1$.

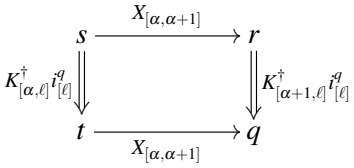


Case 3.2.2.4. $\ell = \alpha+1$ and $j \neq \alpha$. In this case, the exact synthesis algorithm specifies that the normal edge from s is $K^{\dagger}_{[j,\alpha+1]}i^{q}_{[\alpha+1]}$ and the normal edge from r is $K^{\dagger}_{[j,\alpha]}i^{q}_{[\alpha]}$. We complete the diagram as follows. It commutes relationally by (10) and (13).

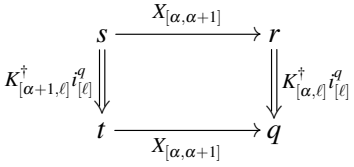


Case 3.2.2.5. $\ell > \alpha+1$ and $j < \alpha$. This case is disjoint.

Case 3.2.2.6. $\ell > \alpha+1$ and $j = \alpha$. In this case, the exact synthesis algorithm specifies that the normal edge from s is $K^{\dagger}_{[\alpha,\ell]}i^{q}_{[\ell]}$ and the normal edge from r is $K^{\dagger}_{[\alpha+1,\ell]}i^{q}_{[\ell]}$. We complete the diagram as follows. It commutes relationally by (5) and (13).



Case 3.2.2.7. $\ell > \alpha+1$ and $j = \alpha+1$. In this case, the exact synthesis algorithm specifies that the normal edge from s is $K^{\dagger}_{[\alpha+1,\ell]}i^{q}_{[\ell]}$ and the normal edge from r is $K^{\dagger}_{[\alpha,\ell]}i^{q}_{[\ell]}$. We complete the diagram as follows. It commutes relationally by (5) and (13).



Case 3.2.2.8. $\ell > \alpha+1$ and $j > \alpha+1$. This case is disjoint.

This finishes the proof of Lemma 3.7, and therefore of completeness.