Natural Computing manuscript No.
(will be inserted by the editor)

Counting pure states

A quantum variant of automatic complexity

Bjørn Kjos-Hanssen

Received: date / Accepted: date

Abstract The automatic complexity of a finite word was defined by Shallit and Wang in 2001. Here, a quantum analogue of automatic complexity is studied. The semi-classical quantum automatic complexity $Q_s(x)$ of the word x over a finite alphabet Σ is the infimum (in lexicographic order) of those pairs of nonnegative integers (n,q) such that there is a subgroup G of the projective unitary group $\mathrm{PU}(n)$ with $|G| \leq q$ and with $U_b \in G$, $b \in \Sigma$ such that, in terms of a standard basis $\{e_k\}$ and with $U_z = \prod_k U_{z(k)}$, we have $U_x e_1 = e_2$ and $U_y e_1 \neq e_2$ for all $y \neq x$ with $|y| = |x|$. We show that Q_s is unbounded and not constant for strings of a given length. In particular,

$$Q_s(0^2 1^2) \leq (2,12) < (3,1) \leq Q_s(0^{60} 1^{60})$$

and $Q_s(0^{120}) \leq (2,121)$.

Keywords Automatic complexity · Quantum computation

1 Introduction

Quantum locks. Imagine a lock with two states, *locked* and *unlocked*, which may be manipulated using two operations, called `0` and `1`. Moreover, the only way to (with certainty) unlock using four operations is to do them in the sequence 0011, i.e., $0^n 1^n$ where $n = 2$. In this scenario one might think that the lock needs to be in certain further states after each operation, so that there is some memory of what has been done so far. Here we show that this memory can be entirely encoded in superpositions of the two basic states *locked* and *unlocked*. As dictated by quantum mechanics, the

This work was partially supported by a grant from the Simons Foundation (#704836 to Bjørn Kjos-Hanssen).

B. Kjos-Hanssen
University of Hawai‘i at Mānoa
Tel.: +1 808-956-8595
E-mail: bjoernkh@hawaii.edu

operations are given by unitary matrices. Moreover, we show using Jordan's theorem on finite linear groups that a similar lock is not possible for $n = 60$.

Quantum security. A problem with traditional padlocks is that a clever lock-breaker can seek to detect what internal state the lock is in part-way through the entering of the lock code. This problem disappears when the internal states are just superpositions of *locked* and *unlocked*. Of course, there may be a positive probability that the system when observed part-way through the entering of the lack code is observed in the *unlocked* state. To remedy this, one could use a sequence of many locks, say $\mathtt{10}^k$ for a suitable positive integer k. Add a third *permanently locked* state which, once reached, cannot be left. Then observing the lock will likely eventually result in landing in the permanently locked state. (In the case of the code $\mathtt{0}^n\mathtt{1}^n$, the permanently locked state might be implemented as having a probability related to the difference in the number of 0s and 1s entered in the code so far.) Note that in theory this is a purely quantum phenomenon: any simulation of the quantum device using classical hardware will be subject to the original problem that a lock-breaker may try to discern the internal states of the classical hardware.

Quantum automata. One of the fascinating aspects of quantum mechanics is how our understanding of states is enriched, with observable states, pure states, and mixed states. The notion of *state* of a finite automaton begs for a generalization to the quantum realm. Indeed, quantum finite automata have been studied already [12].

For the notion of automatic complexity due to Shallit and Wang [17], *arbitrary repetititions* of the form a^k, where a is a word and k an integer, play a main role. For the quantum analogue we shall see that subtle aspects of *group structure* take over this role.

We shall consider complexity with respect to an arbitrary semigroup before considering the quantum case of the projective unitary group $\mathrm{PU}(n)$.

Definition 1 *Let Σ be a finite alphabet. Let T_X denote the set of all transformations of the set X; $T_X = \{f \mid f : X \to X\}$. The complexity of a string $x \in \{0,1\}^n$, $n \geq 0$, is the class of all semigroup actions $\varphi : G \to T_X$ for semigroups G and sets X, with*

- *$\delta_b \in G$, for each $b \in \Sigma$, inducing $\delta_y = \prod_{k=|y|}^{1} \delta_{y_k}$ for each $y \in \Sigma^n$, $y = y_1 \dots y_{|y|}$;*
- *an initial state $\alpha \in X$; and*
- *a final state $\omega \in X$,*

such that x is the only $y \in \Sigma^n$ for which $\delta_y \alpha := \varphi(\delta_y)\alpha = \omega$.

In this case we say that x has complexity at most φ*, or, if φ is understood,* complexity at most G.

1.1 Quantum automatic complexity

Let $e_j^{(n)}$, $1 \leq j \leq n$ be the standard basis for $\mathbb{C}^n$. Let $\mathrm{U}(n)$ be the group of unitary complex $n \times n$ matrices and let $\mathrm{PU}(n)$ be the projective unitary group.

For $n \times n$ matrices U_b, $b \in \Sigma$ and a string x, we define

$$U_x = \prod_{k=1}^{|x|} U_{x(k)}.$$

Definition 2 *A* quantum deterministic finite automaton *(quantum DFA) M with q states consists of an* initial state $\alpha \in \mathbb{CP}^q$, *a* final state ω, *and* $\delta_b \in \mathrm{PU}(q), b \in \Sigma$. *We say that M* accepts *a word $x \in \Sigma^n$, $n \geq 0$ if*

$$\delta_x \alpha = \omega.$$

Let $x \in \Sigma^n$, $n \geq 0$. The quantum automatic complexity of *x, $Q(x)$, is the least q such that there exists a quantum DFA M with q states such that for all $y \in \Sigma^n$, M accepts y iff $y = x$.*

- *If we additionally require that δ_0, δ_1 generate a finite subgroup of* $\mathrm{PU}(q)$, *we obtain the* finite quantum automatic complexity $Q_f(x)$.
- *If we require $\alpha = e_1$ and $\beta = e_2$ then we obtain* semi-classical quantum automatic complexity Q_s.
- *If we require both of the extra requirements for Q_s and Q_f, we get Q_{sf}.*

We can write $Q_s(x) \leq (n, \infty)$ if $Q_s(x) \leq n$, and $Q_s(x) \leq (n, f)$ if $Q_{sf}(x) \leq n$ as witnessed by a finite group of order f. This way we see A_{perm}, the automatic complexity [17] with the added restriction that the transition functions be permutations, as an upper bound for n and a lower bound for f. Ordering these pairs lexicographically, we shall show that

$$(3, 121) \leq Q_s(0^{60}1^{60}) \leq (121, \infty)$$

using Theorem 5.

Definition 3 *Let Σ be a finite alphabet and $n \geq 0$. Let $x \in \Sigma^n$. The permutation complexity $A^{\Sigma}_{\mathrm{perm}}(x)$ is the minimum number of states $q = |Q|$ of a DFA M with alphabet Σ for which $\delta_b, b \in \Sigma$ are bijections of Q, and such that the only word of length $|x|$ accepted by M is x.*

Theorems 4 and 5 show that, in fact, $A^{\Sigma}(x) = |x| + 1$ as long as $|\Sigma| \geq 2$.

Theorem 4 *For any Σ, $A^{\Sigma}_{\mathrm{perm}}(x) \leq |x| + 1$ for all $x \in \Sigma^n$, $n \geq 0$.*

Proof We construct a DFA M. After reading t symbols of x we are in state q_t, $0 \leq t \leq n$, where q_n is the accept state. Suppose $x = ub^s v$ where u does not end in b and v does not start with b. We let δ_b have a cycle of length s on the states $q_{|u|}, \ldots, q_{n-|v|}$. For all b and states q_i for which this leaves $\delta_b(q_i)$ unspecified we let $\delta_b(q_i) = q_i$.

Theorem 5 (Anthony Quas [6]) *If $|\Sigma| \geq 2$ then $A^{\Sigma}_{\mathrm{perm}}(x) \geq |x| + 1$ for all $x \in \Sigma^n$, $n \geq 0$.*

Proof We must show the following:

> Suppose $\delta_b : Q \to Q$, $b \in \Sigma$ are invertible functions, where $Q = \{0, \ldots, q-1\}$ is a set of q elements. For a word $w = w_1 \ldots w_n \in \Sigma^n$ we define $\delta_w = \delta_{w_n} \circ \delta_{w_{n-1}} \circ \cdots \circ \delta_{w_1}$. Suppose c, d and n are such that there is *exactly one* word $w \in \Sigma^n$ with $\delta_w(c) = d$. Then $q \geq n+1$.

To prove it, fix $c \in Q$ and $m \geq 0$. Let $\mathscr{R}_m(c) = \{\delta_w(c) : |w| = m\}$ and $r_m(c) = |\mathscr{R}_m(c)|$. In particular, $\mathscr{R}_0(c) = \{c\}$.

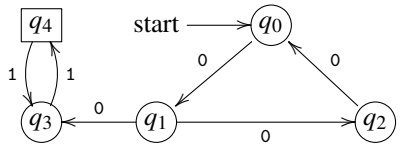
Claim 6 *Let $m \geq 0$. If $r_m(c) = r_{m+1}(c)$, then for all $M > m$ and for all $d \in \mathscr{R}_M(c)$, there are two words w in Σ^M with $\delta_w(c) = d$.*

Proof Let $S = \mathscr{R}_m(c)$. Then $\mathscr{R}_{m+1}(c) = \bigcup_{b\in\Sigma} \delta_b(S)$. By invertibility of the δ_b, each $\delta_b(S)$ has cardinality $r_m(c) = r_{m+1}(c)$, and their union also has cardinality $r_{m+1}(c)$. It follows that $\delta_b(S) = \delta_{b'}(S)$ for all $b, b' \in \Sigma$. Now let $w \in \Sigma^M$ and let $\delta_w(c) = d$. Let $w = uv$ where $|u| = m+1$ and $|v| = M - (m+1) \geq 0$. Let b be the last symbol in u and let $b' \neq b$, $b' \in \Sigma$.

Since $\delta_b(S) = \delta_{b'}(S)$, there exists a $u' \in \delta_{b'}(S)$, $|u'| = m+1$ such that $\delta_{u'}(c) = \delta_u(c)$. Now $\delta_v \circ \delta_{u'}(c) = \delta_v \circ \delta_u(c)$ and the Claim is proved.

It follows that if c and d are such that there is a unique w of length n with $\delta_w(c) = d$, then $r_{j+1}(c) > r_j(c)$ for each $j < n$, so that $q \geq r_n(c) \geq n + r_0(c) = n+1$.

The proof of Theorem 5 somewhat resembles that of the Morse–Hedlund theorem [13]. The significance of Theorem 5 is that, unlike ordinary automatic complexity, quantum automatic complexity will not have any cycles in the witnessing path. From a classical perspective, though, we may view the nonrepeating sequence of pure states as having some cycles among the basis states. It is analogous to the situation with protein folding in the hydrophobic–polar model [2] in which the protein folds, without self-intersections, but forming slipknots (Figures 1 and 3). For contrast, here is a nondeterministic automatic complexity [9] witness (non-quantum) for the word $0^5 1^5$, which does feature self-intersections:



Quantum automatic complexity may be a very rudimentary measure of complexity but we note that, for instance, $0^{60}1^{60}$ and 0^{120} have the same length but distinct complexity. The latter has complexity at most 2 whereas the former does not (Theorem 17).

Theorem 7 *Let x be a word over a finite alphabet Σ.*

1. *If $Q_s(x) > (n, \infty)$ then $Q_s(x) \geq (n+1, A_{\text{perm}}(x))$.*
2. *$Q_s(x) \leq (A_{\text{perm}}(x), A_{\text{perm}}(x))$.*

Proof For (1) we note that the quantum states can be considered as states, so that the Cayley graph of any group witnessing Q_{sf} can be thought of as a witness for A_{perm}. For (2) we note that we can restrict attention to only the states $e_j^{(q)}$, $1 \leq j \leq q$, refusing to use superposition.

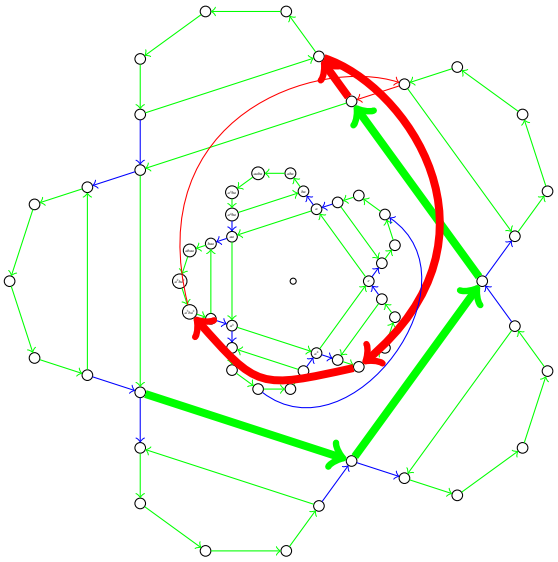


Fig. 1: Cayley graph of A_5 with generators a (green) and b (other colors) subject to the relations $a^5 = b^5 = (ab)^3 = (aabb)^2 = 1$, with $a = (12345)$ and $b = (14325)$. All vertices are shown. All edges are 5-fold symmetric (including symmetry between the inner and outer sets of 6 green pentagons) copies of edges shown. Solid "slipknot" path witnesses quantum automatic complexity $Q_{sf}(000111)$.

Matrices M of dimension $n \times n$ whose entries are 0 and 1, with exactly one 1 per column, act on $X = Q = \{1,\ldots,n\}$ by matrix multiplication in the following way:

$$\varphi(M)(j) = k, \qquad \text{where } Me_j^{(n)} = e_k^{(n)}.$$

If M is additionally invertible then it thus induces an element of the symmetric group S_n and belongs to $\mathrm{O}(n)$, the group of orthogonal matrices M (satisfying $M^{-1} = M^T$).

Theorem 8 *For each string x, $Q_f(x)$ is finite.*

Proof By the embedding of S_n into $\mathrm{O}(n)$ above, and then inclusion of $\mathrm{O}(n)$ into $\mathrm{U}(n)$ (simply because $U^{-1} = U^T$ for a real matrix U implies $U^{-1} = U^\dagger$), we have $Q_f(x) \leq (A_{\mathrm{perm}}(x), A_{\mathrm{perm}}(x)) \leq (x+1, x+1)$.

We also have $Q \leq Q_f \leq Q_{sf}$ and $Q \leq Q_s \leq Q_{sf}$. For our quantum lock analogy we want distinct initial and final states, whereas for automatic complexity $A(x)$ or $A_{\mathrm{perm}}(x)$ it is natural to not require that.

2 Bounds on Q_s

We now show how any binary string can be encoded, in a sense, by two 2×2 matrices.

Theorem 9 (Sanov [15]) *Let*

$$a = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$$

and let b be the transpose of a. Then a and b generate a free subgroup of $\mathrm{SL}(2,\mathbb{Z})$.

We shall not use Theorem 9, also known as the ping-pong lemma, directly. Instead we use some of the ideas of the proof to obtain Theorem 10.

Theorem 10 *For each binary string x there exist* $U_0, U_1 \in \mathrm{GL}_2(\mathbb{Q})$ *such that* $U_x e_1 = e_2$ *and for any* $y \neq x$ *with* $|y| = |x|$, $U_y e_1 \neq e_2$.

Proof Let x be given and let $y \neq x, |y| = |x|$. Let $A_0 = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$ and $A_1 = A_0^T$. Note that $A_i = I + B_i$ where B_i has nonnegative entries. Thus, if D is any matrix with nonnegative entries which is not diagonal then so is $A_i D = D + B_i D$. Consequently, A_x is a matrix with nonnegative entries which is not diagonal.

So there is some i for which e_i is not an eigenvector of A_x. Let $v = e_i$. Let $C = [v|A_x v]$. That is, $Ce_1 = v$ and $Ce_2 = A_x v$. Then $A_x v \notin \mathrm{span}(v)$ and so C is invertible. Let $U_b = C^{-1} A_b C$ for $b \in \{0,1\}$. We have $U_x e_1 = C^{-1} A_x C e_1 = C^{-1} A_x v = C^{-1} C e_2 = e_2$. Suppose y is a word with $C^{-1} A_y C e_1 = e_2$. Let u be the longest common prefix of x and y and write $x = ubx^+$, $y = u\bar{b}y^+$, where $|b| = 1$, $\bar{b} = 1 - b$. Then

$$A_b(A_{x^+} C e_1) = A_u^{-1} C e_2 = A_{\bar{b}}(A_{y^+} C e_1)$$

which is a contradiction to the fact that $\{A_0, A_1\} = \{A_b, A_{\bar{b}}\}$ is a set of two operators with disjoint ranges on the set $P = \{(x,y) : 0 \leq x, 0 \leq y, 0 < x+y\}$. Indeed, consider

$$S = \{(x,y) : 0 \leq x < y\} \subset P.$$

Then

$$A_0 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x+2y \\ y \end{bmatrix}, \qquad A_1 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ 2x+y \end{bmatrix}$$

so $A_1[P] \subseteq S$ and $A_0[P] \subseteq P \setminus S$. So $A_0[P]$ and $A_1[P]$ are disjoint subsets of P.

Theorem 11 $Q(x) \leq 2$ *for all strings x.*

Proof It suffices to show that there is a free group generated by two unitary matrices. It is well-known [20] that a generic pair of unitaries in $\mathrm{U}(2)$ generates a free group. Indeed, the existence of free subgroups of $\mathrm{SO}(3)$ (and hence its double cover $\mathrm{SU}(2)$) was already known to F. Hausdorff [5]; see also [18] and explicit examples in [19].

Unfortunately, perhaps, free groups are incompatible with the *semi-classical* $e_1 \mapsto e_2$ property in the following way:

Theorem 12 *There is no word x of length* $n > 0$ *and pair of unitary matrices* U_0, U_1 *such that* U_0 *and* U_1 *generate a free group and* $U_x e_1 = e_2$ *in projective space* $\mathbb{P}(\mathbb{C}^n)$.

Proof If U is unitary and $Ue_1 = e_2$ in projective space then we may write it as

$$U = \begin{bmatrix} 0 & -b^* \\ be^{i\theta} & 0 \end{bmatrix}.$$

But then, in $\mathbb{P}(\mathbb{C}^n)$,

$$U^2 = \begin{bmatrix} 0 & -b^* \\ be^{i\theta} & 0 \end{bmatrix} \begin{bmatrix} 0 & -b^* \\ be^{i\theta} & 0 \end{bmatrix} = \begin{bmatrix} -|b|^2 e^{i\theta} & 0 \\ 0 & -|b|^2 e^{i\theta} \end{bmatrix} = I_n.$$

Thus, any group in which U is one of the generators is not free.

3 Unboundedness of Q_f

As usual we denote by $H \trianglelefteq G$ that H is a normal subgroup G, and by $[G:H]$ the index of H in G. Let $M_n(\mathbb{C})$ denote the group of $n \times n$ complex matrices.

Theorem 13 (Camille Jordan [10]) *There is a function $f(n)$ such that given a finite group G that is a subgroup of $M_n(\mathbb{C})$, there is an abelian subgroup $H \trianglelefteq G$ such that $[G:H] \leq f(n)$.*

Corollary 14 *For each n there exists an m such that for any $u, v \in \mathrm{U}(n)$ which generate a finite group, we have $[u^m, v^m] = 1$, i.e., $u^m v^m = v^m u^m$.*

Proof (Proof of Corollary from Theorem.) If G is a finite group generated by u and v, and H a normal abelian subgroup of index $[G:H] = m$, then $u^m H = H$ and $v^m H = H$ (since any group element raised to the order of the group is the identity) and so u^m and v^m belong to H, hence, H being abelian, they commute.

Theorem 15 *For each n there is a binary string z with $Q_f(z) > n$.*

Proof Let n be given and let $m = m_n$ be as in Corollary 14. Let $z = 0^m 1^m$. Given $\delta_0, \delta_1 \in \mathrm{PU}(n) = \mathrm{U}(n)/\mathrm{U}(1)$, choose x and y in $\mathrm{U}(n)$ such that $\delta_0 = x\mathrm{U}(1)$ and $\delta_1 = y\mathrm{U}(1)$. Then

$$\delta_z = (x\mathrm{U}(1))^m (y\mathrm{U}(1))^m = x^m y^m \mathrm{U}(1) = y^m x^m \mathrm{U}(1) = \delta_{1^m 0^m}$$

and $z \neq 1^m 0^m$, showing that δ_0, δ_1 do not witness that $Q_f(z) \leq n$.

The extent to which 2×2 matrices suffice for quantum automatic complexity is indicated in Table 1.

Theorem 16 (Collins [1, Theorem A]) *Let G be a finite group.*

- *If G is a primitive subgroup of $\mathrm{GL}(2,\mathbb{C})$, then $[G:Z(G)] \leq 60$.*
- *If G is a primitive subgroup of $\mathrm{GL}(3,\mathbb{C})$, then $[G:Z(G)] \leq 360$.*

Theorem 17 *$Q_f(0^{60}1^{60}) > 2$. and hence $Q_{sf}(0^{60}1^{60}) > 2$.*

	$e_1 \mapsto e_2$ required	not required
finite group required	∞	∞
not required	unknown	2

Table 1: Supremum of quantum automatic complexity over all strings. In the case where $e_1 \mapsto e_2$ is required (semi-classical quantum automatic complexity Q_s) but finiteness (Q_f) is not, we at least know that free groups cannot answer the question, by Theorem 12.

Proof Let $Z(G)$ denote the center of a group G. Note that we may assume our finite subgroups are primitive as there is no point in having a separate automaton disconnected from the witnessing one. By Theorem 16, if G is a finite primitive subgroup of $\mathrm{GL}(2,\mathbb{C})$, then $[G : Z(G)] \leq 60$. Thus $Z(G)$ turns out be an abelian subgroup as desired. Thus if $z = z_n$ is defined as in Theorem 15 for $m = m_n$ with $n = 2$, then $m_2 = 60$ and $z = 0^{60}1^{60}$, giving $Q_f(0^6 01^6 0) > 2$.

On the other hand, we show below in Theorem 23 that $Q_{sf}(0^2 1^2) = 2$, leaving a gap $(2,60)$ for the least n such that $Q_{sf}(0^n 1^n) > 2$. The state of our knowledge of finiteness of quantum automatic complexity is given in Table 1.

4 A concrete example: the word `0011`

In this section we show that $Q_s(0011) \leq (2,12)$. The group $\mathrm{SU}(2)$ is the group of unit quaternions with the matrix representation

$$\mathbf{1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{i} = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}, \qquad \mathbf{j} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \qquad \mathbf{k} = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}$$

where i is the imaginary unit. We shall consider its order 24 subgroup the binary tetrahedral group

$$\left\{ \pm\mathbf{1}, \pm\mathbf{i}, \pm\mathbf{j}, \pm\mathbf{k}, \frac{1}{2}(\pm\mathbf{1} \pm \mathbf{i} \pm \mathbf{j} \pm \mathbf{k}) \right\},$$

also known by isomorphism as $\mathrm{SL}(2,3)$. Moreover we shall consider the order 12 quotient $\mathrm{PSL}(2,3)$ which is isomorphic to the alternating group $\mathrm{Alt}(4)$.

Theorem 18 *There exist* $\mathbf{a}, \mathbf{b} \in \mathrm{SU}(2)$ *such that*

$$\mathbf{aabb} \notin \{\mathbf{a}, \mathbf{b}\}^4 \setminus \{\mathbf{aabb}\}.$$

Proof It turns out we can use the binary tetrahedral group to realize 0011 within $\mathrm{SU}(2)$. Namely, let

$$\mathbf{a} = \delta_0 = (\mathbf{1} + \mathbf{i} + \mathbf{j} - \mathbf{k})/2, \qquad \mathbf{b} = \delta_1 = (\mathbf{1} + \mathbf{i} + \mathbf{j} + \mathbf{k})/2$$

in the quaternion representation,

$$\mathbf{a} = \frac{1}{2} \begin{bmatrix} 1+i & 1-i \\ -i-1 & 1-i \end{bmatrix}, \qquad \mathbf{b} = \frac{1}{2} \begin{bmatrix} 1+i & 1+i \\ i-1 & 1-i \end{bmatrix}.$$

We can check that $\mathbf{aabb} = -\mathbf{j}$ is unique among 4-letter words in $\mathbf{a}, \mathbf{b}$.

Theorem 19 *For $x = 0011$, there exist $\delta_0, \delta_1 \in \mathrm{SO}(3)$ such that for all $y \in \{0,1\}^4$, $\delta_y = \delta_x$ iff $y = x$.*

Proof Another way to express **a** and **b** in Theorem 18 is as

$$e^{i\varphi} \begin{bmatrix} e^{i\Psi} & 0 \\ 0 & e^{-i\Psi} \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} e^{i\Delta} & 0 \\ 0 & e^{-i\Delta} \end{bmatrix}$$

where $\varphi = 0$, $\theta = \pi/4$, and **a** has $(\Psi, \Delta) = (0, \pi/4)$ and **b** has $(\Psi, \Delta) = (\pi/4, 0)$. Thus

$$\mathbf{a} = \frac{1-i}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} i & 0 \\ 0 & 1 \end{bmatrix} = \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \right) \left(\frac{1-i}{\sqrt{2}} \begin{bmatrix} i & 0 \\ 0 & 1 \end{bmatrix} \right)$$

is a product of two matrices **rs** in $\mathrm{SU}(2)$. The first one, **r**, corresponds [4,3] to the $\mathrm{SO}(3)$ rotation

$$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

which is a 90-degree rotation in the xz-plane, and the second one, **s**, to a 90-degree rotation

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

in the xy-plane in $\mathrm{SO}(3)$. We have

$$\mathbf{b} = \frac{1-i}{2} \begin{bmatrix} i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} = \mathbf{sr}.$$

One remaining wrinkle, taken care of in Theorem 23, is to make sure the other words are not only distinct from **aabb**, but map the start state to distinct vectors from what **aabb** does.

Theorem 20 (well known) *The order 24 group $\mathrm{SL}(2,3)$ is given by $a^3 = b^3 = c^2 = abc$, or equivalently $a^3 = b^3 = abab$.*

Theorem 21 ([14]) $\mathrm{SL}(2,3)$ *is isomorphic to the binary tetrahedral group, a subgroup of* $\mathrm{U}(2)$.

The group $\mathrm{Alt}(4)$ does serve as complexity bound for 0011. It is not a subgroup of $\mathrm{U}(2)$ [14], but:

Theorem 22 *There is a faithful, irreducible representation of* $\mathrm{Alt}(4) \cong \mathrm{PSL}(2,3)$ *as a subgroup of* $\mathrm{SL}(2,3)$ *of index 2 and as a subgroup of* $\mathrm{PU}(2)$.

Proof Let a be as in Theorem 20. We define an equivalence relation $\equiv$ by $u \equiv v \iff u \in \{v, a^3 v\}$. It is required to show that each element of our $\mathrm{SL}(2,3)$ is equivalent to an element of $\mathrm{Alt}(4)$. This is done in detail in Figure 2.

The representation from Theorem 22 is used in the proof of Theorem 23.

Theorem 23 $Q_{sf}(\texttt{0011}) = 2$.

$$
\begin{aligned}
&1, a, && a^4 = ab^3 = b^3a = a^2bab = ababa \equiv a,\\
&b, && a^3b = ba^3 = b^4 = abab^2 \equiv b,\\
&a^2 = bab, && a^2b^2 = a^3ba \equiv ba,\\
&ab, && ab^2a = a^2ba^2,\\
&ba, && ba^2b \equiv abba,\\
&b^2 = aba, && b^2a^2 = a^4b = aba^3 = ab^4 \equiv ab,\\
&a^3 = b^3 = baba = abab \equiv 1, && a^5 = a^2b^3 = ab^3a \equiv a^2,\\
&a^2b = bab^2, && a^3b^2 \equiv b^2,\\
&ab^2 = a^2ba, && a^2b^2a \equiv baa,\\
&ba^2 = b^2ab = aba^2b, && ab^2a^2 \equiv aab,\\
&b^2a = aba^2 = ab^2ab, && ba^2b^2 \equiv bba,\\
& && b^2a^2b \equiv abb.
\end{aligned}
$$

Fig. 2: The 24 elements of $\mathrm{SL}(2,3)$. All strings of length at most 2 are unique of their length. By symmetry, words of length 5 starting with b are not written down.

Proof Let $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ with $v_1, v_2 \in \mathbb{R}$. Let

$$E_0 = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ -1-i & 1-i \end{bmatrix},\ E_1 = \tfrac{1}{2}\begin{bmatrix} 1+i & 1+i \\ -1+i & 1-i \end{bmatrix},$$
$$D = \begin{bmatrix} v_1 & -v_2 \\ v_2 & v_1 \end{bmatrix} = [v \mid E_{0011}v], \qquad C = \tfrac{1}{\sqrt{\det D}}D,$$
$$U_j = C^{-1}E_jC, \qquad j \in \{0,1\}.$$

Then it follows that

$$CU_{0011}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = E_{0011}C\begin{bmatrix} 1 \\ 0 \end{bmatrix} = E_{0011}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = C\begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Hence

$$U_{0011}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Since fortunately our E_0 and E_1 satisfy $E_{0011} = -\mathbf{j}$, C is orthogonal, and in particular C is unitary. If we now choose $v = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, then v is sufficiently generic that $U_y e_1 \neq e_2$ as elements of $\mathbb{CP}^1$ for all $y \in \{0,1\}^4 \setminus \{x\}$. We have verified as much with an Octave computation (see Figure 3 and Figure 4). We have

$$C = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix}, \qquad C^{-1} = \tfrac{1}{\sqrt{5}}\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix},$$
$$U_0 = \frac{1}{10}\begin{bmatrix} 5+i & 5+7i \\ -5+7i & 5-i \end{bmatrix},\ U_1 = \tfrac{1}{10}\begin{bmatrix} 5-7i & 5+i \\ -5+i & 5+7i \end{bmatrix}.$$

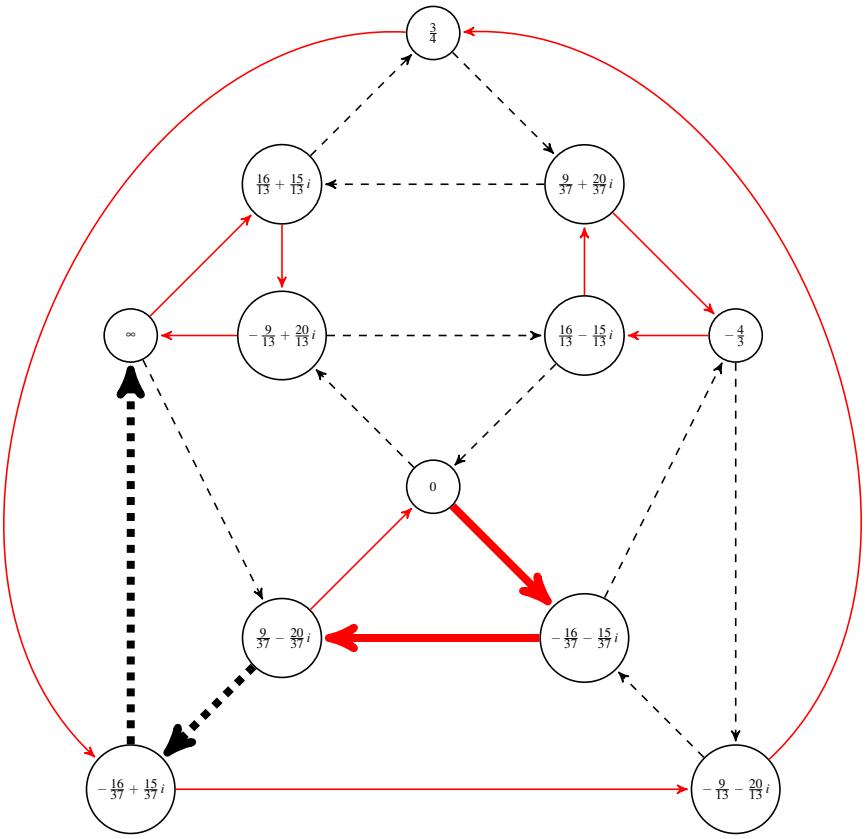


Fig. 3: Quantum complexity witness having the shape of a cuboctahedron. The label α represents the projective point $[1:\alpha]$. The initial state is $[1:0]$ denoted by 0 and the accept state is $[0:1]$ denoted by ∞. Dashed lines indicate multiplication by U_0. Solid lines indicate multiplication by U_1.

5 Dihedral groups

Definition 24 *Suppose x is a word over the alphabet $\{0,1\}$. Let a, b be elements of the dihedral group Dih_k for some k.*

Let $\varphi = \varphi_{a,b,k}$ be the map from words over $\{0,1\}$ to elements of the dihedral group Dih_k (having $2k$ elements) such that $\varphi(0) = a$, $\varphi(1) = b$, and φ takes concatenation to multiplication: $\varphi(xy) = \varphi(x)\varphi(y)$.

We say that x is dihedrally simple *if there is some $\varphi_{a,b,k}$ such that $\varphi(x) \neq \varphi(y)$ for all $y \neq x$ of the same length as x.*

Definition 24 arises in our model of quantum security except we have relaxed and omitted the requirement that we map the locked state $|0\rangle$ to the unlocked state $|1\rangle$.

The relevance of dihedral groups is that Dih_k is representable as a subgroup of the projective unitary group $\mathrm{PU}(2,\mathbb{C})$. The interpretation then is that x is a secret

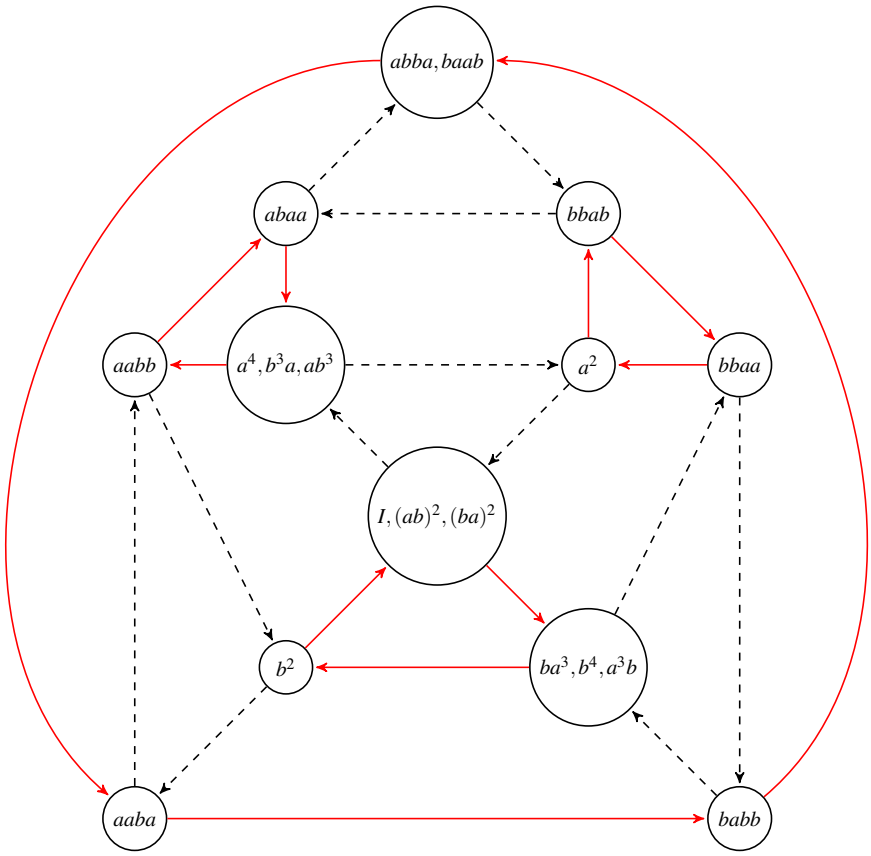


Fig. 4: Another view of the quantum complexity witness having the shape of a cuboctahedron of Figure 3.

code which should be punched into a quantum device with buttons labeled `0` and `1` and which trigger unitary operations U_A, U_B. Any code of the same length as x but different from x will not have the same effect (say, unlocking the device). Any attempt to inspect the state of the device during entering of the code, as one might do with a simple padlock, will constitute a measurement of the device and therefore reset the quantum superposition.

Lemma 25 *A word w over $\{0,1\}$ is dihedrally simple if and only if $\overline{w}$ is.*

Proof This is an immediate consequence of the following more precise observation. The word w is dihedrally simple as witnessed by the function $\varphi_{a,b,k}$ if and only if the word $\overline{w}$ is dihedrally simple as witnessed by the function $\varphi_{b,a,k}$.

Lemma 26 (Luc Guyot [8]) *Let $a \in Dih_k$ be either central or of order $k > 2$ and let $b \in Dih_k$ be a non-central element of order 2. Let w be a word over $\{0,1\}$ of length n. If w contains at least two `1`s (respectively, more than two `1`s), then $\varphi_{a,b,k}(w)$ is either*

of the form a^i or a^ib with $|i| \leq n-2$ (resp. $|i| < n-2$). If w is of the form 0^i10^j, then $\varphi_{a,b,k}(w) = a^{i-j}b$.

Proof Write $w = 0^{i_1}1^{j_1}\cdots 0^{i_s}1^{j_s}$ with $i_t, j_t \geq 0$. It follows from the identities $b^2 = 1$ and $bab = a^{-1}$ that $\varphi_{a,b,k}(w) = a^{\sum_{t=1}^s \pm i_t} b^\varepsilon$ with $\varepsilon \in \{0,1\}$ and $\varepsilon \equiv \sum_{t=1}^s j_t \mod 2$. If w contains at least two 1s (resp. more than two 1s), then $\sum_{t=1}^s i_t \leq n-2$ (resp. $\sum_{t=1}^s i_t < n-2$), which implies the first statement. Consider now the second statement with $w = 0^i10^j$. Then $\varphi_{a,b,k}(w) = a^iba^j = a^{i-j}b$.

Lemma 27 (Luc Guyot [8]) *The elements of S are dihedrally simple.*

Proof If $w \in S$ is of length $n = 0$, then w is certainly dihedrally simple. Let us assume that $n > 0$ and let $k > 4n$. Let $a \in \mathrm{Dih}_k$ be an element of order k and let $b \in \mathrm{Dih}_k$ be a non-central element of order 2. Then $\varphi_{a,b,k}(0^n) = a^n$. Since $\mathrm{Dih}_k = \langle a\rangle \rtimes \langle b\rangle$, it follows from Lemma 1 that the image by $\varphi_{a,b,k}$ of any other word of length n is distinct from a^n. Thus 0^n is dihedrally simple. Since $\varphi_{a,b,k}(0^{n-1}1) = a^{n-1}b$, it also follows from Lemma 1 that $0^{n-1}1$ dihedrally simple. The case of 01^{n-1} is very similar but relies on $\overline{\varphi}_{a,b,k}$ instead; we will omit it. Let us now consider $\varphi_{a,b,k}(10^{n-2}1) = a^{-(n-2)}$. We infer from Lemma 1 that $10^{n-2}1$, and hence $01^{n-2}0$, is dihedrally simple. Eventually, let us consider $\varphi_{b,ba,k}((01)^{n/2})$. This is $a^{n/2}$ if n is even and $a^{(n-1)/2}b$ if n is odd. Since $b^2 = (ba)^2 = 1$, we have $\varphi_{b,ba,k}(00) = \varphi_{b,ba,k}(11) = 1$. Therefore the image of any word of length n by $\varphi_{b,ba,k}$ is the image of a word of the form $(01)^{i/2}$ or $(10)^{i/2}$ with $0 \leq i \leq n$, that is, a group element g of the form $a^{\lfloor i/2\rfloor}b^\varepsilon$ or respectively $a^{-\lfloor i/2\rfloor}(ba)^\varepsilon$ with $\varepsilon \in \{0,1\}$. It is easily checked that such an element g coincides with $\varphi_{b,ba,k}((01)^{n/2})$ if and only if $i = n$, which shows that $(01)^{n/2}$ is dihedrally simple.

Theorem 28 (Luc Guyot [8]) *The dihedrally simple words form a regular language, namely $S \cup T$ where*

$$S = \bigcup_{n=0}^{\infty} \{0^n, \quad 0^{n-1}1, \quad (01)^{n/2}, \quad 01^{n-1}, \quad 01^{n-2}0\},$$

$(01)^{t+\frac{1}{2}} = (01)^t0$*, and T is obtained from S by interchanging* 0 *and* 1.

Proof Given a word w over $\{0,1\}$, we denote by $\overline{w}$ the word obtained from w by interchanging 0 and 1. Let us show first that the elements of $S \cup T$, where $T = \overline{S}$, are dihedrally simple. This is immediate from Lemma 25 and Lemma 27.

For the other direction we must that if w is a dihedrally simple word, then $w \in S \cup T$.

Let w be dihedrally simple word of length n. Since $S \cup T$ contains all words of length at most three, we can assume that $n \geq 4$. By definition, we can find $k \geq 2$, $a, b \in \mathrm{Dih}_k$ such that $\varphi_{a,b,k}(w) \neq \varphi_{a,b,k}(v)$ for every word $v \neq w$ of length n. We can split our reasoning into three cases.

1. The group elements a and b commute. If w contains a subword of the form 01 or 10, then we can interchange these two subwords, producing a word $v \neq w$ such that $\varphi_{a,b,k}(v) = \varphi_{a,b,k}(w)$, a contradiction. Therefore w is of the form 0^n or 1^n, and hence lies in $S \cup T$.

2. The group elements a and b are non-commuting elements of order 2. We have then $\varphi_{a,b,k}(00) = \varphi_{a,b,k}(11) = 1$. Thus w cannot contain any of the subword 00 or 11, since interchanging them would yield a contradiction. Therefore w is of the form $(01)^{n/2}$ or $(10)^{n/2}$. Thus w lies in $S \cup T$.

3. The group elements a and b are non-commuting elements of order 2 and $k > 2$ respectively. We have then $\varphi_{a,b,k}(00) = 1$. If w contains at least one 1, i.e., $w = w'1w''$, then it cannot contain a subword of the form 00, since otherwise moving this subword from w' to w'', or vice versa, would yield a word $v \neq w$ with the same image. As we also have $\varphi_{a,b,k}(101) = \varphi_{a,b,k}(000)$, the word w cannot contain a subword of the form 101. As a result, w is of the form 0^n, or $1^{n-1}0$, or $01^{n-2}0$. Therefore w lies in $S \cup T$.

The very last case consists in interchanging the orders of a and b, but this is too similar to case (3). The theorem is proved.

We may note that the set $S \cup T$ above is closed under inverse morphisms [16]. For instance, if 001011001011 were in the list then by consider the morphism $0 \to 001$, $1 \to 011$, we better also have 0101 in the list. This is necessarily so, because we may change the generators a and b to get other presentations of the dihedral groups.

6 Other finite subgroups

Definition 29 *Let G be a group, possibly infinite. A word $w \in \Sigma^n$ is said to be G-simple if there exist $a_i \in G, i \in \Sigma$ such that the image of w under the monoid homomorphism $phi_a : \Sigma^n \to G$ induced by $i \to a_i$ is distinct from $\varphi_a(v)$ for every other $v \in \Sigma^n$.*

From Theorem 5, we have:

Theorem 30 *If G is finite then every G-simple word w satisfies $|w| \leq |G| - 1$.*

For the other finite subgroups of $\mathrm{PU}(2)$, *viz.*, A_5, S_4 and A_4, we do not have a simple characterization like the one we have proved for dihedral groups in Theorem 28, but by *brute force* we can find all G-simple words, see Appendix A. The A_5-simple words include the S_4-simple ones so we do not list those separately.

We did discover that the group A_5 provides a topologically non-trivial quantum automatic complexity witness for the word $0^3 1^3$. See Figures 5 (where no other word of length 6 leads to where a^3b^3 leads to, from a given starting state) and the discussion at [7].

Acknowledgments

I am grateful to Luc Guyot and Anthony Quas for their contributions to this paper, a preliminary version of which appeared at [11].

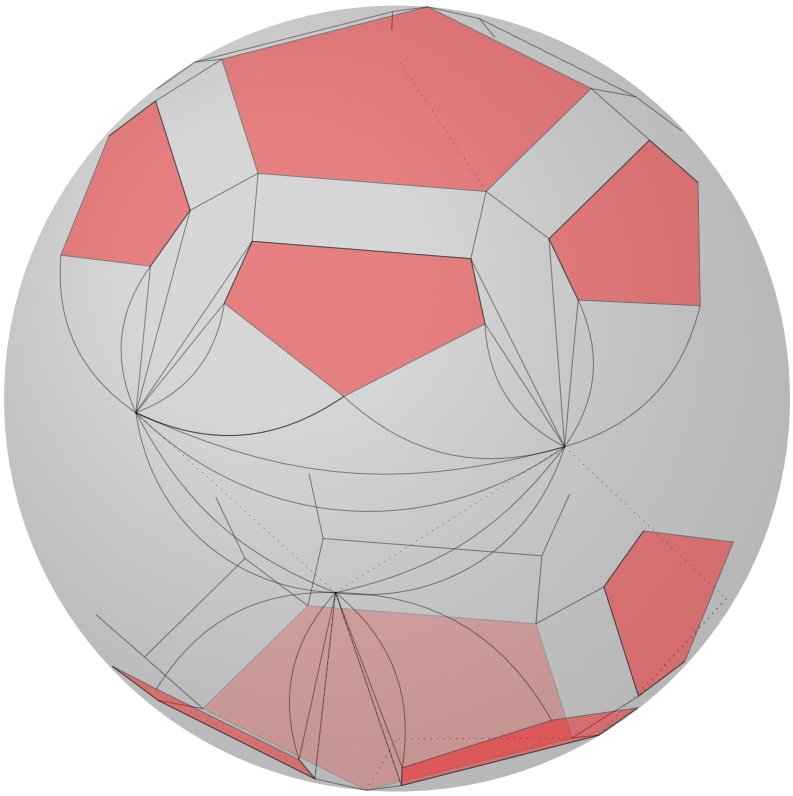


Fig. 5: Sketch of the Cayley graph of A_5 with two generators of order 5, drawn on a sphere with 10 crosscaps.

References

1. Michael J. Collins. Bounds for finite primitive complex linear groups. *J. Algebra*, 319(2):759–776, 2008.
2. Aviezri S. Fraenkel. Complexity of protein folding. *Bulletin of Mathematical Biology*, 55(6):1199–1210, 1993.
3. I. M. Gel′fand, R. A. Minlos, and Z. Ja. ˇ Sapiro. *Predstavleniya gruppy vrashcheni i gruppy Lorentsa, ikh primeneniya*. Gosudarstv. Izdat. Fiz.-Mat. Lit., Moscow, 1958.
4. I. M. Gelfand, R. A. Minlos, and Z. Ya Shapiro. *Representations of the rotation and Lorentz groups and their applications*. Macmillan, New York, 1963.
5. F. Hausdorff. Bemerkung über den Inhalt von Punktmengen. *Math. Ann.*, 75(3):428–433, 1914.
6. Anthony Quas (https://mathoverflow.net/users/11054/anthony quas). Group action with unique word. MathOverflow. URL:https://mathoverflow.net/q/372714 (version: 2020-09-27).
7. Bjørn Kjos-Hanssen (https://mathoverflow.net/users/4600/). Cayley graph of A_5 with generators $(1,2,3,4,5),(1,4,3,2,5)$. MathOverflow. URL:https://mathoverflow.net/q/304138 (version: 2020-01-01).
8. Luc Guyot (https://mathoverflow.net/users/84349/luc guyot). Unique words in dihedral groups. MathOverflow. URL:https://mathoverflow.net/q/278923 (version: 2017-08-18).
9. Kayleigh Hyde and Bjørn Kjos-Hanssen. Nondeterministic automatic complexity of overlap-free and almost square-free words. *Electron. J. Combin.*, 22(3):Paper 3.22, 18 pp., 2015.
10. Camille Jordan. Mémoire sur les équations différentielles linéaires à intégrale algébrique. *J. Reine Angew. Math.*, 84:89–215, 1878.

11. Bjørn Kjos-Hanssen. Superposition as memory: unlocking quantum automatic complexity. In *Unconventional computation and natural computation*, volume 10240 of *Lecture Notes in Comput. Sci.*, pages 160–169. Springer, Cham, 2017.
12. Attila Kondacs and John Watrous. On the power of quantum finite state automata. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 66–75, 1997.
13. Marston Morse and Gustav A. Hedlund. Symbolic dynamics II. Sturmian trajectories. *Amer. J. Math.*, 62:1–42, 1940.
14. Krishna Mohan Parattu and Akın Wingerter. Tribimaximal mixing from small groups, 2011.
15. I. N. Sanov. A property of a representation of a free group. *Doklady Akad. Nauk SSSR (N. S.)*, 57:657–659, 1947.
16. Jeffrey Shallit. *A Second Course in Formal Languages and Automata Theory*. Cambridge University Press, New York, NY, USA, 1 edition, 2008.
17. Jeffrey Shallit and Ming-Wei Wang. Automatic complexity of strings. *J. Autom. Lang. Comb.*, 6(4):537–554, 2001. 2nd Workshop on Descriptional Complexity of Automata, Grammars and Related Structures (London, ON, 2000).
18. S. Świerczkowski. On a free group of rotations of the Euclidean space. *Nederl. Akad. Wetensch. Proc. Ser. A 61 = Indag. Math.*, 20:376–378, 1958.
19. S. Świerczkowski. A class of free rotation groups. *Indag. Math. (N.S.)*, 5(2):221–226, 1994.
20. Andreas Thom. Convergent sequences in discrete groups. *Canad. Math. Bull.*, 56(2):424–433, 2013.

A List of A_5-simple words starting with 0

For lengths $1 \leq n \leq 5$, all 2^{n-1} binary words starting with 0 are A_5-simple. At lengths $6 \leq n \leq 13$, the number of such words is 30,55,49,33,18,5,3,0, respectively.

```
Length 12: 001010010100 010100101001 011010110101
Length 11: 00101001010 01001010010 01010010100 01011010110 01101011010
Length 10: 0000000000 0001001000
0001010101 0001111000 0010010001 0010100101
0011001100 0100010001 0100101001 0100101101
0101001010 0101010111 0101101011 0101111010
0110101101 0110110110 0111011011 0111011101
Length 9
000000000 000100010 000100100 000101010 000111100 001000100 001001000 001001001
001010010 001010100 001010101 001100110 001111000 010000101 010001000 010001001
010010001 010010100 010010110 010100101 010101000 010101001 010101011 010110101
010111101 011001100 011010010 011010101 011010110 011011011 011011101 011101101
011101110
Length 8
00000000 00001010 00001100 00001111 00010001 00010010
00010101 00011000 00011110 00100010 00100100 00100101
00100110 00101001 00101010 00101101 00110000 00110001
00110011 00111100 01000010 01000100 01001000 01001001
01001010 01001011 01010000 01010010 01010100 01010101
01010110 01010111 01011010 01011011 01011110 01100001
01100100 01100110 01101001 01101010 01101011 01101101
01101110 01110011 01110110 01110111 01111000 01111001
01111010
Length 7
0000000 0000100 0000101 0000110 0000111 0001000 0001001 0001010
0001011 0001100 0001110 0001111 0010000 0010001 0010010 0010011
0010100 0010101 0010110 0010111 0011000 0011001 0011010 0011011
0011100 0011101 0011110 0100001 0100010 0100011 0100100 0100101
0100110 0101000 0101001 0101010 0101011 0101100 0101101 0101110
0101111 0110000 0110001 0110010 0110011 0110100 0110101 0110110
0110111 0111000 0111001 0111010 0111011 0111100 0111101
```

```
Length 6
000000 000010 000011 000100 000101 000110 000111 001000 001001 001010
001011 001100 001101 001110 001111 010000 010001 010010 010011 010100
010101 010110 010111 011000 011001 011010 011011 011100 011101 011110
```