RANDOMNESS AND COMPLEXITY IN MATRIX GROUPS

VLADIMIR SHPILRAIN

ABSTRACT. We reflect on how to define complexity of a matrix and how to sample a random invertible matrix. We also discuss a related issue of complexity of algorithms in matrix groups.

In memory of Alfred Lvovich Shmelkin, a man of honor and integrity

1. INTRODUCTION

To define complexity of an algorithm, one has to define complexity of its inputs first. In group theory, it is standard to define complexity of an element g of a group G as the *word length*, i.e., the length of a word in the generators of G that represents the element g. For this complexity to be well-defined, it actually has to be the length of a *shortest* word in the generators of G representing g (i.e., the *geodesic length*), which can be inconvenient because, for instance, there are rather natural groups where computing the geodesic length is NP-hard [13]. This is why, in most natural algorithmic problems in group theory, inputs are words rather than elements. For example, the correct (from the complexity theory point of view) formulation of the conjugacy problem would be: given two words, u and v in the generators of a group G, find out whether or not there is another word t such that the words u and $t^{-1}vt$ represent the same element of G. An input of an algorithm that deals with this problem is therefore a pair of words $\{u, v\}$, and complexity of such an input is the sum of the word lengths of u and v. That said, we also point out that elements of some groups have a unique *normal form*, in which case complexity (whatever it means) of this normal form can be considered complexity of an input. An example of a normal form is a matrix from a group, say, $SL_n(\mathbb{Z})$. Being an element of a group, such a matrix has a presentation as a word in given generators of the ambient group. This presentation is not unique (because the group $SL_n(\mathbb{Z})$ is not free). On the other hand, the same element has a unique presentation (a normal form) as a matrix over $\mathbb{Z}$, i.e., an array of integers. Another point that can be made specifically in reference to the conjugacy problem is that since the word problem is a special case of the conjugacy problem (i.e., conjugacy to the identity element), it would not be a mistake to say that inputs of an algorithm that deals with the conjugacy problem are two *elements* of G rather than words in the generators (see e.g. [12]). Indeed, the conjugacy (decision) problem in a group G may only be

Research of the author was partially supported by the NSF grant CNS-1117675 and by the ONR (Office of Naval Research) grant N000141512164.

solvable if the the word problem is, and if the the word problem is solvable, then one can identify input words with elements of G.

Now we get to the question of how to define complexity of a matrix from, say, $SL_n(\mathbb{Z})$ or $SL_n(\mathbb{Q})$, for a fixed n. Given a matrix $M = (m_{ij})$, there are 3 principal ways to define complexity of M:

(1) This is what can be called the "norm" of a matrix: $||M|| = \sum |m_{ij}|$. Alternatively, it can be $\sqrt{\sum m_{ij}^2}$, etc.

(2) The word length of M with respect to a particular generating set of the ambient group. We denote this by $|M|$ when there is no ambiguity concerning the generating set.

(3) The Kolmogorov complexity. Informally speaking, this is the size of a shortest description of a given matrix M. This is the most adequate definition of complexity from the complexity theory point of view, but the trickiest one to use in concrete algebraic problems, which is why it is very rarely (if ever) used in assessing complexity of algorithms for standard problems in group theory, such as the word problem, the conjugacy problem, etc. We denote the Kolmogorov complexity of a matrix M by $|M|_{Kol}$.

In the following Section 2, we are trying to establish relations between different definitions of complexity.

2. Relations between definitions of complexity

In this section, we discuss relations between three different definitions of complexity mentioned in the Introduction, and also a related question of sampling a random matrix from one or another pool of matrices.

2.1. **The word length vs. the norm.** The norm $||M||$ of a matrix has some nice algebraic properties, in particular it satisfies the triangle inequality for both addition and multiplication of matrices. It was used in [4] to study complexity of some algorithms applied to matrices from $SL_2(\mathbb{Z})$. While this kind of complexity is intuitively adequate for matrices over $\mathbb{Z}$, it appears to be inappropriate for matrices over $\mathbb{Q}$. The reason is simple: for a rational number like $\frac{1}{m}$ with a large m, the absolute value can be very small, but intuitively, complexity of this number should be at least as large as complexity of m. Thus, complexity of a rational number should take into account the absolute values of both the numerator and denominator, and this should be reflected in the complexity of the whole matrix, which is something that standard definitions of a norm cannot provide.

For matrices in $SL_n(\mathbb{Z})$, however, this problem does not arise since larger (by the absolute value) integers have larger complexity. Now an interesting question is about a relation between $||M||$ and $|M|$, the word length with respect to a natural set of generators of $SL_n(\mathbb{Z})$. In the further discussion, we are going to focus on the group $SL_2(\mathbb{Z})$ to keep notation simpler. The latter group has standard generators that we denote by $A(1) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, $B(1) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$.

The group $SL_2(\mathbb{Z})$ is not free, so there are relations between $A(1)$ and $B(1)$, which means that the word length of a matrix $M \in SL_2(\mathbb{Z})$ is not necessarily equal to its geodesic length. On the other hand, the subgroups of $SL_2(\mathbb{Z})$ generated by pairs of matrices $A(k) = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}$, $B(k) = \begin{pmatrix} 1 & 0 \\ k & 1 \end{pmatrix}$ are free if $k \geq 2$, so for those groups comparison between the word length and the norm might be "more convincing".

First, consider a matrix $L = A(k)^m = \begin{pmatrix} 1 & mk \\ 0 & 1 \end{pmatrix}$. We have $|L| = m$, $||M|| = mk + 2$, so for this matrix, $|M|$ and $||M||$ are equivalent up to multiplicative and additive constants.

By contrast, consider a matrix $C = (A(k)B(k))^{\frac{m}{2}}$ (assuming that m is even). Then $|C| = m$, but $||C||$ is exponential in m, according to [3]. Thus, we see that $||M||$ can be equivalent to $|M|$, but it can also be exponential in $|M|$. In any case, $|M| = O(||M||)$ for matrices from that subgroup. A natural question now is:

Problem 1. *Denote by H_k the subgroup of $SL_2(\mathbb{Z})$ generated by the matrices $A(k)$ and $B(k)$, $k \geq 2$. What is the norm $||M||$ of a random matrix $M \in H_k$, as a function of the word length $|M|$ with respect to the generators $A(k)$ and $B(k)$. Is it exponential in $|M|$?*

Of course, the definition of a "random" matrix has to be somehow formalized. This can be done in several different ways; a good survey on the subject is [14]. In the context of Problem 1, it is probably reasonable to use randomness stemming from *asymptotic density*, in the spirit of [7]. That is, one considers a ball B_N of radius N in the Cayley graph of the group H_k generated by $A(k)$ and $B(k)$ and picks a matrix from B_N uniformly at random. Then, assuming that the answer to one or another question (e.g. Problem 1) is given by a function $f(N)$, one computes the (upper) limit of $f(N)$ as N goes to ∞, and this (upper) limit (if it exists) is accepted as an answer to the relevant question.

As far as Problem 1 is concerned, our conjecture is that $||M||$ is exponential in $|M|$ for random matrices M.

2.2. **The norm vs. the Kolmogorov complexity.** We start by pointing out that the Kolmogorov complexity of an integer n is $O(\log n)$ because to describe n (in binary, or decimal, or any other form), it is sufficient to use $O(\log n)$ digits. Therefore, for a matrix M over $\mathbb{Z}$, we have $|M|_{Kol} = O(\log ||M||)$.

For a rational number $\frac{p}{q}$, the Kolmogorov complexity is $O(\log p + \log q) = O(\log pq)$. Therefore, there is no explicit formula relating $|M|_{Kol}$ to $||M||$ for matrices M over $\mathbb{Q}$, so we will leave it at that.

2.3. **The word length vs. the Kolmogorov complexity.** Combining observations in Sections 2.1 and 2.2, we see that for matrices $M \in SL_2(\mathbb{Z})$, $|M|$ can be exponential in $|M|_{Kol}$ (cf. matrix L in Section 2.1), and it can be linear in $|M|_{Kol}$ (cf. matrix C in Section 2.1). Thus, a natural question on "intermediate growth" is:

Problem 2. *Is there an infinite series of matrices in a subgroup H_k, $k \geq 2$, of $SL_2(\mathbb{Z})$ such that for matrices M in this series, one has $|M|$ superlinear but subexponential in $|M|_{Kol}$? Here the word length $|M|$ is considered, as usual, with respect to the standard generators $A(k)$ and $B(k)$.*

3. Random matrices

Random matrices have been studied by many authors, from various points of view, see e.g. recent monographs [1], [17]. In the context of the present paper, we are interested in *sampling* random matrices, i.e., in procedures for generating matrices according to some intuitively reasonable probability distribution(s).

The definition of a "random" matrix from a pool P can be based on one or another definition of complexity as follows. Suppose $|M|_c$ is a reasonably defined complexity of a matrix M. Then one considers a ball B_N of radius N in the pool of matrices we want to sample, with respect to this complexity, i.e., $B_N = \{M \in P : |M|_c \leq N\}$, and picks a matrix from B_N uniformly at random, provided that all B_N are finite. A disadvantage of this method is a bias toward matrices with larger complexity because the sphere of radius N usually comprises a large part of the ball of radius N.

If the pool P to be sampled includes *all* matrices (of a given size) over a ring R, then the problem of sampling matrices from R can be reduced to just sampling elements of R.

If, on the other hand, the pool P to be sampled is, say, the group $SL_n(\mathbb{Z})$, then picking a random matrix M from B_N has to take into consideration the fact that M should have determinant 1. This narrows down the choice of a complexity $|M|_c$ to be used in this context to just $|M|$, the word length of M with respect to a fixed generating set of $SL_n(\mathbb{Z})$ or a subgroup thereof, depending on the problem at hand.

Yet another idea for sampling matrices from $SL_n(\mathbb{Z})$ was discussed in [14]. One can consider a bijection $\beta : SL_n(\mathbb{Z}) \to S$, where S is a set (it does not have to have a group structure). If β is efficiently invertible, then one can sample a random element from S and then apply β^{-1} to get a random element of $SL_n(\mathbb{Z})$. The idea is that the bijection β may "distort" the metric on $SL_n(\mathbb{Z})$, and this will help avoid the "long element bias". A specific example of a bijection considered in [14] stems from a well-known action of $SL_n(\mathbb{R})$ on the upper halfplane equipped with a hyperbolic metric. See [14] for further details; here we just re-iterate the point that selecting an appropriate sampling procedure often depends on a particular problem at hand.

4. Complexity of the subgroup membership problem

Recall that we denote $A(k) = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}$, $B(k) = \begin{pmatrix} 1 & 0 \\ k & 1 \end{pmatrix}$. In an old paper [15], I. N. Sanov proved two simple yet remarkable theorems:

Theorem 1. *The subgroup of $SL_2(\mathbb{Z})$ generated by $A(2)$ and $B(2)$ is free.*

Theorem 2. *The subgroup of $SL_2(\mathbb{Z})$ generated by $A(2)$ and $B(2)$ consists of all matrices of the form $\begin{pmatrix} 1+4n_1 & 2n_2 \\ 2n_3 & 1+4n_4 \end{pmatrix}$ with determinant 1, where all n_i are arbitrary integers.*

These two theorems together yield yet another proof of the fact that the group $SL_2(\mathbb{Z})$ is virtually free because the group of all invertible matrices of the form $\begin{pmatrix} 1+4n_1 & 2n_2 \\ 2n_3 & 1+4n_4 \end{pmatrix}$ obviously has finite index in $SL_2(\mathbb{Z})$.

Our focus here is on Theorem 2 that has the following interesting corollary:

Corollary 1. *The membership problem in the subgroup of $SL_2(\mathbb{Z})$ generated by $A(2)$ and $B(2)$ is solvable in constant time.*

We note that this is, to the best of our knowledge, the only example of a natural (and nontrivial) algorithmic problem in group theory solvable in constant time. In fact, even problems solvable in sublinear time are very rare, see [16], and in those that are, one can typically get either "yes" or "no" answer in sublinear time, but not both. In light of Theorem 2, deciding whether or not a given matrix from $SL_2(\mathbb{Z})$ belongs to the subgroup generated by $A(2)$ and $B(2)$ boils down to looking at residues modulo 2 or 4 of the entries. The latter is decided by looking just at the last one or two digits of each entry (assuming that the entries are given in the binary or, say, decimal form). We also note that in this case, it does not matter what definition of complexity of an input one uses because a constant is a constant no matter what.

We emphasize though that solving this membership problem in constant time is only possible if an input matrix is known to belong to $SL_2(\mathbb{Z})$; otherwise one would have to check that the determinant of a given matrix is equal to 1, which cannot be done in constant time, although can still be done in sublinear time with respect to the norm $||M||$ of an input matrix M.

What would be a natural generalization of Sanov's Theorem 2 to $A(k)$ and $B(k)$, $k \in \mathbb{Z}_+$, is not valid for $k \geq 3$ and moreover, it was shown in [4] that the subgroup H_k generated by $A(k)$ and $B(k)$ has infinite index in $SL_2(\mathbb{Z})$ if $k \geq 3$.

It was also shown in [4] that there is a simple greedy algorithm for the membership problem in the subgroup H_k, $k \in \mathbb{Z},\ k \geq 2$. We note in passing that in general, the subgroup membership problem for $SL_2(\mathbb{Q})$ is open, while in $SL_2(\mathbb{Z})$ it is solvable since $SL_2(\mathbb{Z})$ is virtually free. The general solution, based on the automatic structure of $SL_2(\mathbb{Z})$ (see [5]), is not so transparent and has quadratic time complexity (with respect to the word length of an input). For a special case of subgroups H_k we have:

Proposition 1. [4] *Let $k \in \mathbb{Z},\ k \geq 2$, and let the complexity $||M||$ of a matrix $M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$ be the sum of all $|m_{ij}|$. There is a (greedy) algorithm that decides whether or not a given matrix $M \in SL_2(\mathbb{Z})$ is in the subgroup H_k of $SL_2(\mathbb{Z})$ generated by $A(k)$ and $B(k)$ (and if it does, finds a presentation of M as a group word in $A(k)$ and $B(k)$) in time $O(n \cdot \log n)$, where $n = ||M||$.*

We emphasize that the quadratic time complexity result of [5] holds with respect to the word length of an input matrix, while the $O(n \cdot \log n)$ time complexity result of Proposition 1 holds with respect to the norm $n = ||M||$ of an input matrix. Relation between the word length $|M|$ and the norm $||M||$ is discussed in our Section 2.1, but the bottom line is: $|M| = O(||M||)$ for $M \in H_k$. However, if an input is given as a matrix, then to apply the algorithm in [5], one would have to first present this matrix as a word in the generators, and this takes us back to the algorithm referred to in Proposition 1.

Finally, we note that statement similar to Proposition 1 holds also for the *monoid* generated by $A(k)$ and $B(k)$, for any $k \in \mathbb{Z},\ k > 0$.

4.1. **Generic-case complexity.** The $O(n \cdot \log n)$ is the worst-case complexity of the algorithm referred to in Proposition 1. It would be interesting to find out what the *generic-case complexity* (in the sense of [7]) of this algorithm is.

The basic theory of generic-case complexity was developed in [7] and [8]. It turns out that in many interesting cases the traditional group-theoretic decision problems, such as the word, conjugacy and subgroup membership problems, have provably very low generic-case complexity, even where the worst-case complexity is very high, or indeed, where the problem is undecidable. Moreover, it was proved in [8] that the true *average-case complexity* is frequently low as well. This is true even though solving the word problem for some specific words may take much more than linear time in the worst case. In fact there need not even be any algorithm at all which solves the word problem for all words. Similar sorts of results followed for other group-theoretic problems, involving individual group elements (e.g. [7], [10]) as well as subgroups ([2]). In each case a decision problem which might be quite difficult in the worst case was shown to be easy for "most" inputs, that is, for inputs in a so-called generic set. The precise definition of "generic" depends on the problem at hand, but is supposed to be natural in each particular situation. To study the generic complexity of algorithms in a specific group G, one has to somehow define a measure on G; then sets that asymptotically have the same measure as the whole of G will be generic. It turns out that to make such a definition natural, one has to take into account not only the nature of the problem at hand, but also the nature of the group G. For instance, in [9] it was shown that, if one takes a subset S of a free abelian group $\mathbf{Z}^k$ and its full preimage $\hat{S}$ in the free group F_k of the same rank, then the measure (more often called the "density") of S in $\mathbf{Z}^k$ in the "classical" sense (used in number theory for a long time now) is equal to the measure of $\hat{S}$ in F_k defined in a natural yet rather subtle way (the "annular density"). In a recent survey [6], other possible definitions of density were discussed.

In any case, generic-case complexity refers to the difficulty of solving a given problem for a generic set of inputs, or, intuitively, for "random" inputs.

Now we are going to explain the reason why we think the membership problem for the subgroup of $SL_2(\mathbb{Z})$ generated by $A(k)$ and $B(k)$ has low (probably sublinear time) generic-case complexity. The starting point is the following observation: the entries of matrices that are products of length n of positive powers of $A(k)$ and

$B(k)$ exhibit the fastest growth (as functions of n) if $A(k)$ and $B(k)$ alternate in the product: $A(k)B(k)A(k)B(k)\cdots$. More formally:

Proposition 2. [3] *Let $w_n(a,b)$ be an arbitrary positive word of even length n, and let $W_n = w_n(A(k),B(k))$, with $k \geq 2$. Let $C_n = (A(k)\cdot B(k))^{\frac{n}{2}}$. Then:* **(a)** *the sum of entries in any row of C_n is at least as large as the sum of entries in any row of W_n;* **(b)** *the largest entry of C_n is at least as large as the largest entry of W_n.*

The entries of C_n can be determined explicitly from a system of recurrence relations. These recurrence relations are linear, with constant coefficients, so their solutions are sums of exponential functions in n. This implies that complexity of the (greedy) algorithm mentioned in Proposition 1 is log of the complexity (= norm) of the input, when the algorithm is applied to a matrix C_n. On the other hand, if this algorithm is applied to a power of $A(k)$ or $B(k)$, then its complexity is linear in the size of the input. Now intuitively, a random (reduced) product of matrices $A(k)^{\pm 1}$ and $B(k)^{\pm 1}$ is "closer" to C_n for some n than it is to a power of $A(k)$ or $B(k)$ since the expected number of $A(k)^{\pm 1}$ factors in such a product is the same as the expected number of $B(k)^{\pm 1}$ factors. This is why we believe that the generic-case complexity of the algorithm referred to in Proposition 1 is sublinear (in fact, logarithmic) in the norm of an input matrix, which would be a really interesting result, so we ask:

Problem 3. *Is the generic-case complexity of the algorithm claimed in Proposition 1 sublinear in $||M||$?*

We note that, unlike the algorithms with low generic-case complexity considered in [7], this algorithm has a good chance to have low generic-case complexity giving both "yes" and "no" answers.

REFERENCES

[1] G. Akemann, J. Baik, P. Di Francesco, *The Oxford Handbook of Random Matrix Theory*, Oxford University Press. Reprint edition 2015.

[2] F. Bassino, C. Nicaud, P. Weil, *Generic properties of subgroups of free groups and finite presentations*, Contemporary Math., Amer. Math. Soc., to appear.

[3] L. Bromberg, V. Shpilrain, A. Vdovina, *Navigating in the Cayley graph of $SL_2(\mathbb{F}_p)$ and applications to hashing*, Semigroup Forum, to appear. http://arxiv.org/abs/1409.4478

[4] A. Chorna, K. Geller, V. Shpilrain, *On two-generator subgroups of $SL_2(\mathbb{Z})$, $SL_2(\mathbb{Q})$ and $SL_2(\mathbb{R})$*, J. Algebra, to appear.

[5] D. B. A. Epstein, J. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, W. P. Thurston, *Word processing in groups*. Jones and Bartlett Publishers, Boston, MA, 1992.

[6] I. Kapovich, *Musings on generic-case complexity*, preprint.

[7] I. Kapovich, A. G. Myasnikov, P. Schupp, V. Shpilrain, *Generic-case complexity, decision problems in group theory and random walks*, J. Algebra **264** (2003), 665–694.

[8] I. Kapovich, A. G. Myasnikov, P. Schupp, V. Shpilrain, *Average-case complexity and decision problems in group theory*, Advances in Math. **190** (2005), 343–359.

[9] I. Kapovich, I. Rivin, P. Schupp, V. Shpilrain, *Densities in free groups and $\mathbb{Z}^k$, visible points and test elements*, Math. Res. Lett. **14** (2007), 263–284.

[10] I. Kapovich, P. Schupp, and V. Shpilrain, *Generic properties of Whitehead's algorithm and isomorphism rigidity of random one-relator groups*, Pacific J. Math. **223** (2006), 113–140.

[11] R. C. Lyndon and P. E. Schupp, *Combinatorial Group Theory*, Ergebnisse der Mathematik, band 89, Springer 1977. Reprinted in the Springer Classics in Mathematics series, 2000.

[12] A. G. Myasnikov, V. Shpilrain, and A. Ushakov, *Non-commutative cryptography and complexity of group-theoretic problems*, Amer. Math. Soc. Surveys and Monographs, 2011.

[13] A. Myasnikov, V. Romankov, A. Ushakov, A. Vershik, *The word and geodesic problems in free solvable groups*, Trans. Amer. Math. Soc. **362** (2010), 4655–4682.

[14] I. Rivin, *How to pick a random integer matrix? (and other questions)*, Math. Comput. **85** (2016), 783–797.

[15] I. N. Sanov, *A property of a representation of a free group* (Russian), Doklady Akad. Nauk SSSR (N. S.) **57** (1947), 657–659.

[16] V. Shpilrain, *Sublinear time algorithms in the theory of groups and semigroups*, Illinois J. Math. **54** (2011), 187–197.

[17] T. Tao, *Topics in Random Matrix Theory*, Graduate Studies in Mathematics, American Mathematical Society, 2012.

DEPARTMENT OF MATHEMATICS, THE CITY COLLEGE OF NEW YORK, NEW YORK, NY 10031

E-mail address: shpil@groups.sci.ccny.cuny.edu