Jahresbericht der Deutschen Mathematiker-Vereinigung (2022) 124:129–135
https://doi.org/10.1365/s13291-022-00248-6

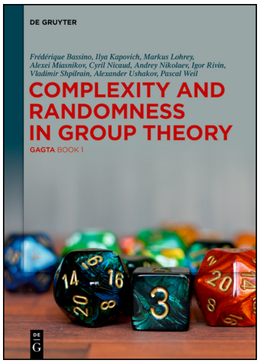
BOOK REVIEW

F. Bassino et al.: "Complexity and Randomness in Group Theory"

de Gruyter, 2020, xii+374 pp.

Laurent Bartholdi[1]

Accepted: 28 February 2022 / Published online: 10 March 2022


DE GRUYTER
Frédérique Bassino, Ilya Kapovich, Markus Lohrey, Alexei Miasnikov, Cyril Nicaud, Andrey Nikolaev, Igor Rivin, Vladimir Shpilrain, Alexander Ushakov, Pascal Weil
COMPLEXITY AND RANDOMNESS IN GROUP THEORY
GAGTA BOOK 1

Since its beginnings, group theory has been an eminently *applied* field: the monodromy group of a differential equation (namely, the transformations of the space of solutions as one encircles an algebraic singularity; see [6] for a captivating account), or the discrete plane symmetry groups (with the 17 cases richly illustrated in the Alhambra and in Escher's work).

More recently, groups have proven to be of immeasurable value as containers of mathematical problems, as well as providing tools for their solution. Let me begin with three instances, chosen to be sufficiently far apart as to show the diversity of the involved group theory.

1 Applications of Group Theory

1.1 Knots

The *knot problem* asks, given a knot (an imbedded circle $K \subset \mathbb{R}^3$) by its *diagram*, namely a generic projection of the knot to a plane, with the specification at each crossing of which one goes above and which one goes below, whether or not the knot is actually knotted (that is, whether a smooth motion of $\mathbb{R}^3$ can bring the knot to a round circle).

An approach to the knot problem is via fundamental groups: choose a basepoint $* \in \mathbb{R}^3 \setminus K$, and consider the homotopy classes of loops based at $*$ in $\mathbb{R}^3 \setminus K$; these form a group $G_K = \pi_1(\mathbb{R}^3 \setminus K, *)$ under concatenation of paths. By Dehn's lemma,

✉ L. Bartholdi
laurent.bartholdi@gmail.com

1 Universität des Saarlandes, Saarbrücken, Germany

G_K is a *complete invariant*: K is unknotted if and only if $G_K \cong \mathbb{Z}$, a generator being any loop that encircles K exactly once; see [13, §4B] for details.

Now it is easy to read off a presentation of G_K from any diagram of K; it has one generator per visible strand of the diagram, and one relation per crossing. In this manner we "reduce" the knot problem to the following question: given a finitely presented group, is it isomorphic to $\mathbb{Z}$?; or, more generally, given two finitely presented groups, are they isomorphic?

Alas, this last question is undecidable, as are most questions about finitely presented groups: by a cornerstone result of Adyan [2], simplified by Rabin [11] and building on Markov Jr's [9], if $\mathcal{P}$ is an abstract property of finitely presentable groups which holds for some group G_+ and fails for all groups containing some group G_-, then $\mathcal{P}$ is undecidable.

Nevertheless, the problem of deciding whether $G_K \cong \mathbb{Z}$ *is* decidable: G_K belongs to a subclass of finitely presented groups (for example, it is *residually finite* — every inequality in G_K can be witnessed in some finite quotient of G_K) for which the problem is decidable.

An outstanding open question is whether the knot problem is in the class **P** of problems decidable in polynomial time; that is, whether there is an algorithm $\mathcal{A}$ and a polynomial P such that, given a diagram with n strands, $\mathcal{A}$ determines whether it represents the unknot in time at most $P(n)$. At present, the complexity of the knot problem is known to be in the class **NP** $\cap$ **co-NP**, that is, for every diagram there exists either a polynomial-length proof that it is the unknot, or a polynomial-length proof that it is not the unknot.

1.2 Rubik's Cube and Similar Puzzles

There is a large number of puzzles with the following structure: there are objects $o_1, \dots, o_n$ at distinct positions out of $\{p_1, \dots, p_n\}$, and a specification of some "elementary" moves which permute the objects at certain positions. The puzzle is solved when each object o_i is brought to the correct position p_i. A typical example is the "Rubik's cube", with objects the $n = 54$ facets of the cube; there are nine elementary moves, consisting of rotations of a slice of the cube. We refer to [8] for an entertaining list of such puzzles, along with the associated group theory.

Expressed mathematically, the following is what it means to solve the cube, or any such puzzle. The current position of the objects is expressed as a permutation σ on n letters, with $\sigma(i) = j$ if the object at position p_i is o_j. The elementary moves are expressed as a set S of permutations, and the goal is to express σ as a product of elements of S — or prove that there are no such expressions, as in the case of Sam Loyd's "15 puzzle".

There is a standard strategy to solve this problem algorithmically, which is at the heart of essentially all computations with permutation groups: the *stabilizer chain*. Let $G \le S_n$ be the group generated by S, and let G_i be the subgroup of G that fixes $1, 2, \dots, i$; so $G = G_0 \ge G_1 \ge \dots \ge G_n = 1$. For all i let T_i be a set of left coset representatives of G_{i-1} in G_i, expressed both as permutations and as words over S. These may be found by exhaustive search, as part of a pre-processing step. The total size of the T_i is at most n^2, while the size of G could be $n!$.

Given then a permutation σ, we first find an element of $t_1 \in T_1$ with $t_1^{-1}\sigma \in G_1$; then $t_2 \in T_2$ with $t_2^{-1}t_1^{-1}\sigma \in G_2$; etc., and in the end $t_n^{-1} \cdots t_1^{-1}\sigma \in G_n = 1$, so $\sigma = t_1 \cdots t_n$. If at any moment we could not find a $t_i \in T_i$ such that $t_i^{-1} \cdots t_1^{-1}\sigma$ fixes i, then σ does not belong to G. It is not hard to see that the word obtained in this manner has length at most quadratic in n. (This gives an upper bound of a few thousands to the minimal number of moves needed to unscramble the Rubik's cube, admittedly far from the exact answer 20 [12].)

1.3 An Infinite Puzzle

The classical "Towers of Hanoi" puzzle is presented as follows: there are three rods, and $n = 64$ disks, sorted in decreasing size, on the first rod. The goal is to move the disks one at a time, never putting a disk on a smaller one, so that they are eventually all on the second rod.

Here, the puzzle's state is expressed as an element s of $\{0, 1, 2\}^n$, with s_k indicating the rod on which disk #k is. There are three elementary moves a_0, a_1, a_2, uniquely specified as follows: for $\{i, j, k\} = \{0, 1, 2\}$, the operation a_i moves the top disk from rod j to rod k or from rod k to rod j, whichever is smaller. The action of a_i on a state is $a_i(i \dots ij \dots) = i \dots i(3 - i - j) \dots$ if $j \neq i$ and $a_i(i \dots i) = i \dots i$. The goal is to find a word in the a_i's mapping 0^n to 1^n, and it is a classical exercise to show that the minimum length of such a word is $2^n - 1$.

Now there is no need to restrict to finite n, and a very interesting group H, generated by $\{a_0, a_1, a_2\}$ and acting on $\Omega := \{0, 1, 2\}^\infty$, arises [7]. Note that this is a group of a quite different nature than the groups considered in Section 1.1: both are finitely generated, but the group H is given by generators acting in an explicit manner while the groups G_K are given as abstract groups (that is, without any concrete representation by permutations). A variety of questions may be asked of H: given an almost-everywhere-0 sequence $s \in \Omega$, how many generators are needed to bring it to the everywhere-0 sequence (bringing back all disks to rod 0)? which word would do this?; given a word over $\{a_0, a_1, a_2\}$, does it preserve the everywhere-0 configuration? if not, what is the largest / smallest disk moved? etc. In all cases, if these problems are solvable then one would like to know the complexity of their solution.

2 Words, Words, Words

A number of group-theory problems can be expressed using free groups and words. For this purpose, recall that, for a set S, the *free group* on S has elements all reduced words (that is, without consecutive xx^{-1}) over the alphabet $S \sqcup S^{-1}$, with as group operation concatenation followed by maximal cancellation. The free group F_S admits the universal property that every map $S \to G$ extends uniquely to a group homomorphism $F_S \to G$. In particular, saying that $S \subseteq G$ is a generating set is equivalent to giving oneself a surjective homomorphism $F_S \twoheadrightarrow G$; and giving oneself a group presentation is akin to positing a homomorphism $F_S \twoheadrightarrow G$ and a set of normal generators for its kernel (the *relators*).

In this terminology, the *word problem* of a group $G = \langle S \rangle$ is really a *subset* of F_S: the kernel of the map $F_S \twoheadrightarrow G$. One may then ask whether there is an algorithm, or even a polynomial-time algorithm, that receives as input a word in F_S and determines whether it maps to 1 in G. Similarly, the *conjugacy problem* receives as input a pair of words x, y, and is asked to determine whether there exists z such that xz and zy map to the same element in G.

We note that there is a continuum of finitely generated groups [10, §14], but only countably many algorithms; so "most" finitely generated groups have unsolvable word problem. Fortunately, "most" groups are not interesting! Much effort has focused on finitely presented groups, for which (as we mentioned above) the word problem is undecidable. In important subclasses, such as "automatic groups" and in particular "word-hyperbolic groups" (see [5]), they are quite efficiently solvable; this covers for example all fundamental groups of 3-manifolds. Even more, the word problem is solvable in a *uniform* manner: there is an algorithm that takes as input a group presentation and a word in the generators, and returns, in case the presentation does define an automatic group G, whether the word evaluates to 1 in G. (The algorithm is allowed to do anything, including crash the computer, in case G is not automatic.) For hyperbolic groups, it is even possible to modify the presentation so that the following very simple algorithm ("Dehn's algorithm") succeeds: "scan the word to see if more than half a relator appears as a subword; if so, replace that subword by the inverse of the other half of the relator; freely reduce and repeat. The end result is trivial if and only if the original word evaluates to 1." Carefully coded, this algorithm can be made to run in real time (that is, on a multitape Turing machine that reads one input letter at each clock tick).

Just as most groups are not interesting, most words are not. For example, in a group generated by $\{x, y\}$, the word $(xyx^{-1}y^{-1})^{1000}x(yxy^{-1}x^{-1})^{1000}x^{-1}$ has length 8002 but is highly structured, and has much more chances of appearing as an input than any random sequence of x's and y's. When measuring complexity, should it count as an input of length 8002, or of length 53 (the length of its LaTeX code)? This topic is nicely formalized in *straight line programs*: by definition, in a group $G = \langle S \rangle$, a straight line program is a sequence of words $(w_1, \dots, w_n)$ such that each w_i is an element of $F_{S \sqcup \{w_1, \dots, w_{i-1}\}}$. (There are other formalisms, all equivalent.) The w_i may be evaluated in sequence to elements of G, and the value of a straight line program is the evaluation of its last term w_n. The length of a straight line program is the sum of the length of its words. Simple examples show that the length of a straight line program may be logarithmic in the length of its value.

It is quite remarkable that the word problem in free groups may be solved in polynomial time, with straight line programs as input (one then refers to the "compressed word problem"). This was independently proven by numerous authors, and we simply refer to Section 4.6 of the book under review, where this result is extended to a large classes of groups; in particular, "graph groups" (generators are vertices, which commute when they are connected by an edge) and the word-hyperbolic groups mentioned above. Furthermore, for these groups the compressed conjugacy problem is also solvable in polynomial time.

3 Worst-Case or Generic-Case?

The undecidability of the word problem ultimately relies on the ability of encoding a Turing machine into a word; so the instances of the word problem that trip up a purported algorithm are necessarily quite contrived. In a quite recent approach, developed in Chapter 1 of the book under review, one asks for *generic-case* complexity or solvability; in that the algorithm is required to succeed on an overwhelming proportion of the inputs — in a density approaching 1 as the input length tends to infinity.

For example, if G is a finitely generated group and $\overline{G}$ is an infinite quotient of G with solvable word problem, then the word problem is generically solvable in G. Indeed with overwhelming probability the (non)triviality of an element of G can be detected in the quotient $\overline{G}$. The same holds for a variety of other decision problems.

Group theory has been suggested as a convenient tool for cryptography, and in particular key exchange protocols. Recall the problem: Alice and Bob wish to share a secret while only using an unprotected communication channel on which Eve eavesdrops. For this, Alice and Bob each select a key, one part of which is secret and one part of which is public. What is desired is a function $f(x, y)$ such that $f(\text{Alice}_{\text{public}}, \text{Bob}_{\text{private}}) = f(\text{Bob}_{\text{public}}, \text{Alice}_{\text{private}})$, which will then be the shared secret; and such that $f(x, y)$ cannot be computed, or even guessed, from $\text{Alice}_{\text{public}}$ and $\text{Bob}_{\text{public}}$. (It is acceptable for f to be complicated, slow and with small range (say 64 bits); it will typically be used only to produce an encryption/decryption password for a faster and more efficient cipher.)

There are numerous methods of achieving this; perhaps the most famous is the "Diffie-Hellman key exchange protocol" [4]. It specifies some global data: a large finite field and a generator g of its group of units. Alice and Bob each respectively choose an integer a, b as their private key, and respectively publish g^a, g^b as their public key. The shared secret is $(g^a)^b = (g^b)^a$. The difficulty in recovering a private key from a public one comes from the (supposed) hardness of the discrete logarithm problem.

If quantum computing keeps progressing at the same pace, the discrete logarithm problem will soon become tractable [14]. On the other hand, more combinatorial problems, such as combinatorics of words, seem immune to quantum computing speedups [1]. A family of key exchange protocols have been suggested by Anshel, Anshel and Goldfeld [3], based on a group G and two finite subsets $A, B \subset G$ that are public data. For $w \in F_{A \sqcup B}$, let us write $\overline{w}$ for its evaluation in G. Alice and Bob respectively choose a word $x \in F_A, y \in F_B$ as their private key, and respectively publish $\{\overline{x^{-1}bx} : b \in B\}$ and $\{\overline{y^{-1}ay} : a \in A\}$ as their public key. Using Bob's public key, Alice may substitute $y^{-1}ay$ for each letter a in her key, obtaining $\overline{y^{-1}xy}$; and similarly for Bob. The shared secret is $\overline{x^{-1}(y^{-1}xy)} = \overline{(x^{-1}y^{-1}x)y}$.

For this to be a useful protocol, one should also specify how the group G and its generators are to be produced; represented (for the common secret to be of any use); and explain why Eve cannot easily guess x from the knowledge of B and their conjugates under x. It would make us comfortable if the conjugacy problem were unsolvable in G, though this is not exactly the problem that Eve must solve: she already knows that b and $x^{-1}bx$ are conjugate, but she must find an element $x \in \langle A \rangle$ that realizes this conjugation; and she must even find one that works for all $b \in B$.

It was at one moment thought that braid groups could serve for G; though they now seem irremediably compromised. I am not aware of candidate replacements for G that would offer security guarantees based on undecidability of a decision problem; see [15] for a discussion of this.

Another issue is the selection of the finite subsets A, B and the words x, y. Choosing words uniformly at random is probably a very bad idea, because of what we noted in the beginning of this section: it could well be that in G the "conjugacy search problem" is undecidable, but that generically it is easily solvable. The instances used for cryptographical purposes must then be as carefully selected as the words encoding a Turing machine.

4 The Book

These notes were but an overview of the book under review. Part of its attraction is that it has 10 authors and 6 chapters, so there is no (obvious) bijection between authors and chapters. Chapter 1 develops generic-case complexity in detail; I have summarized parts of it in the last section, combining my discussion with Chapter 6, which describes group-theoretical problems motivated by their applications to cryptography. Chapter 4 discusses compression techniques, and in particular the algorithmic complexity of straight line programs; I have compressed it to the second question. I have completely left out very interesting topics related to randomness: Chapter 2 answers the questions: what is a random finitely presented group, and what does it look like? what does a random subgroup look like? It could have been titled "generic presentations/subgroups", to emphasize the connection to Chapter 1. Chapter 3 is also concerned with generic properties, but more specifically in infinite linear groups; it explains what are "random elements", and how to produce them. Chapter 5 considers decision problems in groups beyond the word and conjugacy problems; for example, the group-theoretic analogue of the "knapsack problem": given $g_1, \dots, g_k, g \in G$, decide if there are non-negative integers $i_1, \dots, i_k$ with $g = g_1^{i_1} \cdots g_k^{i_k}$, or in words can you fill a knapsack of size g using only objects $g_1, \dots, g_k$ in that order?

The authors are well-known experts in the field, and they have collected an impressive amount of material in this volume. It is not intended as an introduction, nor as a definitive treatise; but rather a snapshot, taken in 2021, of a very active and lively domain of mathematics, with numerous connections within itself, to classical century-old problems, to applications, and to the future.

Funding Note Open Access funding enabled and organized by Projekt DEAL.

References

1. Aaronson, S., Ambainis, A.: The need for structure in quantum speedups. Theory Comput. **10**, 133–166 (2014). https://doi.org/10.4086/toc.2014.v010a006. MR3249097
2. Adyan, S.I.: Algorithmic unsolvability of problems of recognition of certain properties of groups. Dokl. Akad. Nauk SSSR (N.S.) **103**, 533–535 (1955) (Russian). MR0081851
3. Anshel, I., Anshel, M., Goldfeld, D.: An algebraic method for public-key cryptography. Math. Res. Lett. **6**(3–4), 287–291 (1999). MR1713130 (2000e:94034)
4. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theory **IT-22**(6), 644–654 (1976). https://doi.org/10.1109/tit.1976.1055638. MR437208
5. Epstein, D.B.A., Cannon, J.W., Holt, D.F., Levy, S.V.F., Paterson, M.S., Thurston, W.P.: Word Processing in Groups. Jones & Bartlett, Boston (1992)
6. Gray, J.J.: Linear Differential Equations and Group Theory from Riemann to Poincaré, 2nd edn. Birkhäuser Boston, Inc., Boston (2000). MR1751835
7. Grigorchuk, R.I., Šunić, Z.: Asymptotic aspects of Schreier graphs and Hanoi Towers groups. C. R. Math. Acad. Sci. Paris **342**(8), 545–550 (2006) (English, with English and French summaries). MR2217913 (2006k:20048)
8. Joyner, D.: Adventures in Group Theory: Rubik's Cube, Merlin's Machine, and Other Mathematical Toys, 2nd edn. Johns Hopkins University Press, Baltimore (2008). MR2599606
9. Markov, A.A.: The impossibility of certain algorithms in the theory of associative systems. Dokl. Akad. Nauk SSSR (N.S.) **77**, 19–20 (1951) (Russian). MR0040231
10. Neumann, B.H.: Some remarks on infinite groups. J. Lond. Math. Soc. **12**, 120–127 (1937). https://doi.org/10.1112/jlms/s1-12.46.120 (English)
11. Rabin, M.O.: Recursive unsolvability of group theoretic problems. Ann. Math. (2) **67**, 172–194 (1958). MR0110743 (22 #1611)
12. Rokicki, T., Kociemba, H., Davidson, M., Dethridge, J.: The diameter of the Rubik's cube group is twenty. SIAM J. Discrete Math. **27**(2), 1082–1105 (2013). https://doi.org/10.1137/120867366. MR3068558
13. Rolfsen, D.: Knots and Links. Mathematics Lecture Series, vol. 7. Publish or Perish, Inc., Houston (1990). Corrected reprint of the 1976 original. MR1277811
14. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997). https://doi.org/10.1137/S0097539795293172. MR1471990
15. Shpilrain, V., Zapata, G.: Combinatorial group theory and public key cryptography. Appl. Algebra Eng. Commun. Comput. **17**(3–4), 291–302 (2006). https://doi.org/10.1007/s00200-006-0006-9. MR2233788