THE ART OF DISCRETE AND APPLIED MATHEMATICS

ISSN 2590-9770
The Art of Discrete and Applied Mathematics
https://doi.org/10.26493/2590-9770.1405.4ce
(Also available at http://adam-journal.eu)

The string C-group representations of the Suzuki, Rudvalis and O'Nan sporadic groups

Dimitri Leemans*, Jessica Mulpas

Département de Mathématique, Université libre de Bruxelles, C.P.216, Boulevard du Triomphe, 1050 Brussels, Belgium

Received 5 March 2021, accepted 2 June 2021

Abstract

We present new algorithms to classify all string C-group representations of a given group G. We use these algorithms to classify all string C-group representations of the Suzuki, Rudvalis and O'Nan sporadic groups. The new rank three algorithms also permit us to get all string C-group representations of rank three for the Conway group Co_2 and the Fischer group Fi_{22}.

Keywords: Abstract regular polytopes, string C-group representations, sporadic simple groups, algorithms.

Math. Subj. Class.: 52B11, 20D08, 20F05

1 Introduction

Marston Conder has been one of the pioneers in generating all geometric or combinatorial objects of a certain kind, as for instance all regular (hyper)maps of small genus, all abstract regular or chiral polytopes with a sufficiently small number of flags, etc. Conder's website[1] contains an impressive collection of data that greatly helped researchers over the years to get a better insight on the related topics. Most of the data available on Conder's website are in terms of groups and generators, the groups being automorphism groups of the objects considered. A natural way of gaining knowledge on the structure of a group is indeed to search for geometric and combinatorial objects on which it can act. Among those objects, abstract regular polytopes are of great interest as they are highly symmetric combinatorial

*This work was partially supported by an ARC Advanced grant from the Fédération Wallonie-Bruxelles, Belgium. The author would like to thank Martin Macaj for interesting discussions on finding a faster algorithm in the rank three case.

E-mail addresses: Leemans.Dimitri@ulb.be (Dimitri Leemans), jessica.mulpas@outlook.com (Jessica Mulpas)

[1] https://www.math.auckland.ac.nz/~conder/

structures with many singular algebraic and geometric properties. Moreover, due to the well-known one-to-one correspondence between abstract regular polytopes and string C-groups, finding an abstract regular polytope with a given automorphism group G means obtaining a sequence of involutions generating G, satisfying a certain intersection property and such that two nonconsecutive involutions commute. This means that G is a string C-group. Finally, string C-groups are smooth quotients of Coxeter groups, hence any Coxeter diagram found in this process permits to conclude that the group appears as a quotient of the corresponding (infinite) Coxeter group.

In 2006, Michael Hartley published "An atlas of small regular polytopes" [7], where he classified all regular polytopes with automorphism groups of order at most 2000 (not including orders 1024 and 1536). His work has been complemented later on by Conder who computed all regular polytopes with up to 2000 flags, up to isomorphism and duality, excluding those of rank 2 (regular polygons) and the degenerate examples that have a '2' in their Schläfli symbol, using the LowIndexNormalSubgroups function of MAGMA [1]. The complete list is available on Conder's website.

Also in 2006, Dimitri Leemans and Laurence Vauthier published "An atlas of polytopes for almost simple groups" [11], in which they gave classifications of all abstract regular polytopes with automorphism group an almost simple group with socle of order up to 900,000 elements. More recently in [8], Hartley also classified, together with Alexander Hulpke, all polytopes for the sporadic groups as large as the Held group (of order 4,030,387,200) and Leemans and Mark Mixer classified, among others, all polytopes for the third Conway group Co_3 (of order 495,766,656,000) [10]. Leemans, Mixer and Thomas Connor then computed all regular polytopes of rank at least four for the O'Nan-Sims group $\mathsf{O'N}$ [3].

All these computational data lead to many theoretical results. For more details we refer to the recent survey [9].

The recorded computational times and memory usages for the use of previous algorithms on groups among which $\mathsf{O'N}$ and Co_3 provide a motivation to improve the algorithms in order to obtain computational data for groups currently too large.

The challenges involve dealing with groups that have a relatively large smallest permutation representation degree as well as groups with large Sylow 2-subgroups. The largest sporadic group for which a classification of all string C-group representations exists was Co_3 group whose smallest permutation representation degree is on 276 points and whose Sylow 2-subgroups are of order 2^{10}.

We use the technique described in [3] and develop it further to obtain an algorithm that can be used on any permutation group as well as any matrix group to find polytopes of rank at least four.

We also present a new algorithm to find polytopes of rank three and two new algorithms to find polytopes of rank at least four that outperform the previous known algorithms.

With our new algorithms we manage to classify all string C-group representations of the Suzuki sporadic group Suz and the Rudvalis sporadic group Ru. Although these groups have order smaller than the one of Co_3, their smallest permutation representation degrees are respectively 1782 and 4060, and their Sylow 2-subgroups are of respective orders 2^{13} and 2^{14}. We also finish the classification of the string C-group representations for $\mathsf{O'N}$ started in [3] by computing all of its rank three representations. Finally, we classify all string C-group representations of rank three for the Conway group Co_2 and the Fischer group Fi_{22}.

The classifications for Suz, Ru and O′N are made available in the online version of the Atlas [11].

The paper is organised as follows. In Section 2, we give the background needed to understand this paper. In Section 3, we describe the new algorithm to compute rank three string C-group representations of a group. In Section 4, we describe three new algorithms to compute rank three string C-group representations of a group and compare the efficiency of the algorithms. In Section 5, we describe the new results we manage to obtain with the algorithms described in the previous two sections.

2 Abstract regular polytopes and string C-groups

An *abstract n-polytope* $\mathcal{P}$ is a ranked partially ordered set $(\mathcal{P}, \leq)$ with four defining properties (P1), (P2), (P3) and (P4) as defined below. Here, n is the *rank* of $\mathcal{P}$. We shall name the elements of $\mathcal{P}$ *faces* and differentiate the elements by rank calling an element of rank $i \in \{-1, 0, \cdots, n-1, n\}$ an *i-element*. For any two faces F and G of $\mathcal{P}$ such that $F \leq G$, we also define the *section* G/F to be the collection of all faces H of $\mathcal{P}$ such that $F \leq H \leq G$. Note that any section of a polytope is itself a polytope.

(P1) $\mathcal{P}$ has two *improper* faces : a least face F_{-1} of rank -1 and a greatest face F_n of rank n.

(P2) Each flag (i.e. maximal totally ordered subset) of $\mathcal{P}$ contains $n+2$ faces (including the two improper faces).

(P3) $\mathcal{P}$ is *strongly connected*, that is, each section of $\mathcal{P}$ (including $\mathcal{P}$ itself) is connected (in the sense given below).

(P4) $\mathcal{P}$ satisfies the *diamond condition*, that is, for any two faces F and G of $\mathcal{P}$ such that $F < G$ and $\text{rank}(G) = \text{rank}(F) + 2$, there are exactly two faces H such that $F < H < G$.

A ranked partially ordered set of rank d with properties $(P1)$ and $(P2)$ is said to be connected if either $d \leq 1$ or $d \geq 2$ and for any two *proper* faces F and G of $\mathcal{P}$ (i.e. any faces other than F_{-1} and F_d) there is a sequence of proper faces $F = H_0, H_1, \cdots, H_{k-1}, H_k = G$ such that $H_i \leq H_{i+1}$ for $i \in \{0, 1, \cdots, k-1\}$.

Two flags of an n-polytope are said to be *adjacent* if they differ by only one face. In particular, two flags are said to be *i-adjacent* if they differ only by their i-face. As a consequence of the diamond condition, any flag Φ of a polytope has exactly one i-adjacent (for any $i \in \{0, 1, ..., n-1\}$) flag that we shall denote by Φ^i. Note that $(\Phi^i)^i = \Phi$ for all i and that $(\Phi^j)^i = (\Phi^i)^j$ for all i and j such that $\mid i - j \mid \geq 2$

The automorphism group of an n-polytope $\mathcal{P}$ is denoted by $\Gamma(\mathcal{P})$. An n-polytope is said to be *regular* if its automorphism group $\Gamma(\mathcal{P})$ has exactly one orbit on the flags of $\mathcal{P}$, or equivalently, if for any flag $\Phi = \{F_{-1}, F_0, F_1, \cdots, F_n\}$ and each $i \in \{0, 1, ..., n-1\}$, there is a unique (involutory) automorphism ρ_i of $\mathcal{P}$ such that $\rho_i(\Phi) = \Phi^i$. In this case, one usually chooses a flag as a reference flag and calls it *base* flag.

Finally, an n-polytope is called *equivelar* is for any $i \in \{1, 2, ..., n-1\}$, there is an integer p_i such that any section G/F of $\mathcal{P}$ defined by an $(i-2)$-face F and an $(i+1)$-face G is a p_i-gon. If this is the case then we say that $\mathcal{P}$ has *Schläfli type* $\{p_1, p_2, ..., p_{n-1}\}$.

Let G be a group and $S := \{\rho_0, \rho_1, ..., \rho_{n-1}\}$ be a set of elements of G such that $G = \langle S \rangle$. We define the following two properties.

(SP) the *string property*, that is $(\rho_i\rho_j)^2 = 1_G$ for all $i, j \in \{0, 1, ..., n-1\}$ with $\mid i-j \mid \geq 2$;

(IP) the *intersection property*, that is $\langle \rho_i \mid i \in I\rangle \cap \langle \rho_j \mid j \in J\rangle = \langle \rho_k \mid k \in I \cap J\rangle$ for any $I, J \subseteq \{0, 1, ..., n-1\}$.

A pair (G, S) as above that satisfies property (SP) is called a *string group generated by involutions* (or sggi in short). A *string C-group* is an sggi that satisfies property (IP). The *rank* of (G, S) is the size of S. For any subset $I \subseteq \{0, \dots, n-1\}$, we denote $\langle \rho_j : j \in \{0, \dots, n-1\} \setminus I\rangle$ by G_I. If $I = \{i\}$, we denote $G_{\{i\}}$ by G_i. Similarly, if $I = \{i, j\}$, we denote $G_{\{i,j\}}$ by G_{ij}.

As shown in [12], the automorphism group $\Gamma(\mathcal{P})$ of an abstract regular polytope $\mathcal{P}$, together with the involutions that map a base flag to its adjacent flags is a string C-group representation (see [12, Propositions 2B8, 2B10 and 2B11]). More precisely, if one fixes a base flag Φ of $\mathcal{P}$, $\Gamma(\mathcal{P}) = \langle \rho_0, \rho_1, ..., \rho_{n-1}\rangle$ with ρ_i the unique involution such that $\rho_i(\Phi) = \Phi^i$ then the pair $(\Gamma(\mathcal{P}), \{\rho_0, \rho_1, ..., \rho_{n-1}\})$ is a string C-group representation. The *Schläfli type* of a string C-group representation $(G, \{\rho_0, \dots, \rho_{n-1}\})$ is the ordered set $\{p_1, p_2, ..., p_{n-1}\}$ where p_i is the order of the element $\rho_{i-1}\rho_i$ for $i = 1, \dots, n-1$.

Conversely, as shown in [12, Theorem 2E11], a regular n-polytope can be constructed uniquely from a string C-group representation (G, S) with $S := \{\rho_0, ..., \rho_{n-1}\}$. Let $G_i := \langle \rho_j \mid j \neq i\rangle$ for any $i \in \{0, 1, ..., n-1\}$. We also set $G_{-1} = G_n := G$. For $i \in \{-1, 0, 1, ..., n-1, n\}$, we take the set of i-faces of $\mathcal{P}$ to be the set of all right cosets $G_i\varphi$ of G_i in G. We define a partial order on $\mathcal{P}$ as follows : $G_i\varphi \leq G_j\psi$ if and only if $-1 \leq i \leq j \leq n$ and $G_i\varphi \cap G_j\psi \neq \emptyset$.

We say that two string C-group representations (G, S) and (G, S') of G are *isomorphic* if there exists an automorphism of G that maps S onto S'. The *dual* of a string C-group representation $(G, \{\rho_0, \rho_1, ..., \rho_{n-1}\})$ is the string C-group representation $(G, \{\rho_{n-1}, \rho_{n-2}, ..., \rho_0\})$.

Due to the one-to-one correspondence between string C-group representations and automorphism groups of abstract regular polytopes, finding all abstract regular polytopes with a fixed automorphism group amounts to considering all ways of presenting a particular group as a string C-group and verifying whether they yield non-isomorphic abstract regular polytopes. Our algorithms classify string C-group representations of a group G up to isomorphism and duality.

3 The rank three case

Every rank three string C-group representation of a group G is a pair $(G, \{\rho_0, \rho_1, \rho_2\})$ where $\langle \rho_0, \rho_1\rangle$ is a dihedral group and ρ_2 commutes with ρ_0.

The dihedral subgroups of G can be generated from the conjugacy classes of elements of G using the following simple observation.

Lemma 3.1. *Let G be a group. The group G has a dihedral subgroup D if and only if there exists an element τ of G and an involution $\rho \neq \tau$ of G such that $D := \langle \tau, \rho\rangle$ and $\tau^\rho = \tau^{-1}$.*

Proof. If there exists a dihedral subgroup D in G, take two involutions that generate D, say ρ and ρ'. Then $\tau = \rho\rho'$ is such that $D := \langle \tau, \rho\rangle$ and $\tau^\rho = \tau^{-1}$.

On the other hand, suppose there is a subgroup D of G such that $D := \langle \tau, \rho\rangle$ and $\tau^\rho = \tau^{-1}$ for some element τ of G and some involution ρ of G distinct from τ. Then

$\rho' := \rho\tau$ is an involution (as $\rho'\rho' = \rho\tau\rho\tau = \rho^{-1}\tau\rho\tau = \tau^{-1}\tau = 1_G$ and $\rho' \neq 1_G$ since $\rho \neq \tau$) and $\rho\rho' = \rho\rho\tau = \tau$. Therefore D is a dihedral subgroup of G. □

So an easy way to construct all conjugacy classes of dihedral subgroups of a group G is the following:

1. Let $\mathcal{D} := \emptyset$.

2. Compute a set C containing one representative of each conjugacy class of elements of G.

3. For each $\tau \in C$, compute $N(\tau) := N_G(\langle\tau\rangle)$.

4. For each involution $\rho \in N(\tau)$, if $\tau^\rho = \tau^{-1}$ then add $\langle\tau, \rho\rangle$ to $\mathcal{D}$ provided it is not conjugate in G to an element already in $\mathcal{D}$.

At the end of the above process, $\mathcal{D}$ contains one representative of each conjugacy class of dihedral subgroups of G.

In order to generate all rank three string C-group representations of G, we now need to do the following.

1. Let $\mathcal{S} := \emptyset$.

2. For each $D \in \mathcal{D}$

3. For each pair ρ_0, ρ_1 of generating involutions of D, if it is not conjugate in G to a pair already in $\mathcal{S}$, add this pair to $\mathcal{S}$

At the end of the above process, $\mathcal{S}$ contains all the pairs ρ_0, ρ_1 that we try now to extend to a triple of involutions.

Here we can either try to extend the ordered pair $[\rho_0, \rho_1]$ to the right or to the left (which is equivalent to extending the reversed pair $[\rho_1, \rho_0]$ to the right). If we try to extend it to the right, that is by adding a ρ_2, we know that $\rho_2 \in C_G(\rho_0)$. So we can try every involution in $C_G(\rho_0)$, and if G is simple, we also know that ρ_2 does not commute with ρ_1, for otherwise the string C-group $(G, \{\rho_0, \rho_1, \rho_2\})$ is degenerate and G is therefore not simple (e.g. $\langle\rho_1, \rho_2\rangle$ being a non-trivial normal subgroup of G).

This algorithm turns out to be extremely efficient as we will show in Section 5. It permitted us to compute all rank three string C-group representations of Ru, Suz, O′N, Co_2 and Fi_{22}.

4 The ranks four and above

We describe three techniques to find string C-group representations of rank at least four.

4.1 Using centralizers of involutions

This algorithm uses the following observations.

Lemma 4.1. *Let G be a group. Let $(G, \{\rho_0, \ldots, \rho_{n-1}\})$ be a string C-group representation of G. The subgroup G_1 is a subgroup of the centraliser of ρ_0, in particular $G_1 = \langle\rho_0\rangle \times G_{01}$ where $G_{01} \leq C_G(\rho_0)$.*

Proof. This is a direct consequence of Proposition $2B12$ in [12] and of the definition of string C-groups. □

This lemma implies that we can take for ρ_0 a representative of a conjugacy class of involutions of G, compute its centraliser $C_G(\rho_0)$ and then the subgroup G_1 must be a subgroup of $C_G(\rho_0)$.

We then proceed to construct G_1. In order to do so, we produce all string C-group representations of subgroups H of $C_G(\rho_0)$ that have ρ_0 as first generator. As $C_G(\rho_0)$ is substantially smaller than G, one may now apply the former algorithms (from [10]) to $C_G(\rho_0)$ in order to classify all the string C-group representations of all its subgroups. With each string C-group representation $\langle\rho_0, \rho_2, ..., \rho_{n-1}\rangle$ of G_1, we proceed by adding an appropriate involution ρ_1 of G to it, giving a string C-group representation for G. In order to restrict the number of involutions ρ_1 to consider, we use the following observation.

Lemma 4.2. *If $(G, \{\rho_0, \ldots, \rho_{n-1}\})$ is a string C-group representation of the group G, then $\rho_1 \in C_G(\rho_3) \cap \ldots \cap C_G(\rho_{n-1})$.*

Proof. Since ρ_1 has to commute with every generator ρ_i for $i \geq 3$ by property (SP), we have that $\rho_1 \in C_G(\rho_3) \cap \ldots \cap C_G(\rho_{n-1})$. □

We then check that the resulting pair $(G, \{\rho_0, \rho_1, ..., \rho_{n-1}\})$ is a string C-group representation of G using the following proposition.

Proposition 4.3 ([12, Proposition 2E16(a)]). *Let $(G, \{\rho_0, \rho_1, ..., \rho_{n-1}\})$ be an sggi. If G_0 and G_{n-1} are string C-groups and if $G_{n-1} \cap G_0 = G_{0,n-1}$ then $(G, \{\rho_0, \rho_1, ..., \rho_{n-1}\})$ is a string C-group.*

Finally, let us note that although the former algorithms only allowed to be implemented for groups of permutations, this new algorithm also works for matrix groups. This gives access to groups much larger in size, as long as the centralizers of involutions of G are small enough to be treated as permutation groups.

We give in Table 1 the pseudo-code of our new algorithm.

4.2 Higher ranks from the new rank three algorithm

If in the process of generating the rank three string C-group representations with the algorithm described in Section 3, we keep also the triples that do not generate the group G, we can then try to extend these triples to 4-tuples and so on, getting string C-group representations of all possible ranks.

There are two possible approaches here. We could either decide to keep triples of involutions ρ_0, ρ_1, ρ_2 and try to extend them to rank four and so on, or keep the triples of the shape ρ_0, ρ_1, ρ_3, namely, keeping the triples where the third involution commutes with the first two. We have designed algorithms to try both approaches. We call the first approach the linear approach and the second one the central approach.

4.3 The linear approach

We can easily give an algorithm based on the rank three method presented in Section 3 to produce all string C-group representations of all ranks for a given group. Indeed, if, while computing the rank three representations we keep track of all triples ρ_0, ρ_1, ρ_2 that satisfy

Input: G the group for which we want to compute all
       string C-group representations.
Output: L a sequence containing the pairwise non-isomorphic
        string C-group representations.

1. Compute the conjugacy classes $CC(G)$ of elements of G.
2. Initialise L.
3. For each conjugacy class c of elements of order 2 in $CC(G)$:
4.   Let r_0 be a representative of that class.
5.   Build the centralizer H of r_0 in G.
6.   If G is a matrix group,
7.   find a permutation representation $P(H)$ of H;
8.   reduce the degree of $P(H)$;
9.   If G is already a permutation group then
10.   only reduce its degree and call it $P(H)$.
11.   Using the existing procedures to do so (see [11]),
12.   compute the string C-group representations with generators $\{r_0, r_2, \ldots, r_{n-1}\}$ for subgroups of $C_G(r_0)$, forcing r_0 as first generator.
13.   For each such representation, try to complete it by inserting an involution r_1 between r_0 and r_2, using the fact that it has to be in the centralisers of the r_i's for $i = 3, \ldots, n-1$.
14.   Compute $G_0 = \langle r_1, ..., r_{n-1}\rangle$, $G_{n-1} = \langle r_0, ..., r_{n-2}\rangle$ and $G_{0,n-1} = \langle r_1, ..., r_{n-2}\rangle$.
15.   Check the intersection property for G_0 and G_{n-1} and
16.   Check that $G_0 \cap G_{n-1} = G_{0,n-1}$.
17.   Let $GP = \langle r_0, r_1, ..., r_{n-1}\rangle$ and let $\widetilde{GP} = \langle r_{n-1}, ..., r_1, r_0\rangle$ be its dual.
18.   If GP and $\widetilde{GP}$ are non-isomorphic to all the elements of L,
19.   add GP to L.
20. At the end, L contains one representative of each isomorphism class of string C-group representation of G.

Table 1: The pseudo-code of the new algorithm for ranks 4 and above

the intersection property but do not generate the full group G, we can then try to extend these triple to four-tuples and so on in the following way. Suppose $\mathcal{P}$ contains the triples ρ_0, ρ_1, ρ_2 that satisfy the intersection property but do not generate G.

- While $\mathcal{P}$ is not empty do the following.
- Let $\mathcal{Q}$ be an empty set.
- Let r be the number of involutions in a tuple of $\mathcal{P}$.
- For each tuple $\rho_0, \ldots, \rho_{r-1}$ of $\mathcal{P}$, try to extend it on the left by looking for involutions $\rho_{-1} \in C_G(\rho_1) \cap \ldots \cap C_G(\rho_{r-1})$ such that $o(\rho_{-1}\rho_0) > 2$ and such that $\langle \rho_{-1}, \ldots, \rho_{r-1} \rangle = G$ and $\{\rho_{-1}, \ldots, \rho_{r-1}\}$ satisfies the intersection property. All such ρ_{-1} give a string C-group representation of rank $r+1$. In this process, whenever a ρ_{-1} is found such that $\langle \rho_{-1}, \ldots, \rho_{r-1} \rangle < G$ and $\{\rho_{-1}, \ldots, \rho_{r-1}\}$ satisfies the intersection property, add it in $\mathcal{Q}$ provided it is not isomorphic to any element of $\mathcal{Q}$ yet.
- For each tuple $\rho_0, \ldots, \rho_{r-1}$ of $\mathcal{P}$, try to extend it on the right by looking for involutions $\rho_r \in C_G(\rho_0) \cap \ldots, \cap C_G(\rho_{r-2})$ such that $o(\rho_{r-1}\rho_r) > 2$ and such that $\langle \rho_0, \ldots, \rho_r \rangle = G$ and $\{\rho_0, \ldots, \rho_r\}$ satisfies the intersection property. All such ρ_r give a string C-group representation of rank $r+1$. In this process, whenever a ρ_r is found such that $\langle \rho_0, \ldots, \rho_r \rangle < G$ and $\{\rho_0, \ldots, \rho_r\}$ satisfies the intersection property, add it in $\mathcal{Q}$ provided it is not isomorphic to any element of $\mathcal{Q}$ yet.
- At the end of this for loop, we have found all string C-group representations of rank $r+1$ of G and in $\mathcal{Q}$ we have $r+1$ tuples that we can try to extend further.
- Let $\mathcal{P} := \mathcal{Q}$ and continue the while loop.

4.4 The central approach

We can also give an algorithm based on the rank three one to produce all string C-group representations of all ranks for a given group but using the observations of Section 4.1. Indeed, if, while computing the rank three representations we keep track of all ordered triples ρ_0, ρ_1, ρ_2 that satisfy the intersection property but do not generate the full group G, and such that ρ_2 commutes with both ρ_0 and ρ_1, we can then try to extend these triple to four-tuples and so on in the following way. Suppose $\mathcal{P}$ contains these triples ρ_0, ρ_1, ρ_2 described above.

- While $\mathcal{P}$ is not empty do the following.
- Let $\mathcal{Q}$ be an empty set.
- Let r be the number of involutions in a tuple of $\mathcal{P}$.
- For each tuple $\rho_0, \ldots, \rho_{r-1}$ of $\mathcal{P}$, try to extend it on the left by looking for involutions $\rho_{-1} \in C_G(\rho_1) \cap \ldots, \cap C_G(\rho_{r-2})$ such that $o(\rho_{-1}\rho_0) > 2$, $o(\rho_{-1}\rho_{r-1}) > 2$ and such that $\langle \rho_{-1}, \ldots, \rho_{r-1} \rangle = G$ and $\{\rho_{-1}, \ldots, \rho_{r-1}\}$ satisfies the intersection property. All such ρ_{-1} give a string C-group representation $(G, \{\rho_{r-1}, \rho_{-1}, \rho_0, \ldots, \rho_{r-2}\})$ of rank $r+1$. In this process, whenever a ρ_{-1} is found such that $\langle \rho_{-1}, \ldots, \rho_{r-1} \rangle < G$, $o(\rho_{-1}\rho_{r-1}) = 2$ and $\{\rho_{-1}, \ldots, \rho_{r-1}\}$ satisfies the intersection property, add the

ordered tuple $\{\rho_{-1}, \rho_0, \ldots, \rho_{r-2}, \rho_{r-1}\}$ in $\mathcal{Q}$, provided it is not isomorphic to any element of $\mathcal{Q}$ yet.

- For each tuple $\rho_0, \ldots, \rho_{r-1}$ of $\mathcal{P}$, try to extend it on the right by looking for involutions $\rho_r \in C_G(\rho_0) \cap \ldots, \cap C_G(\rho_{r-3})$ such that $o(\rho_{r-1}\rho_r) > 2$, $o(\rho_{r-2}\rho_r) > 2$ and such that $\langle \rho_0, \ldots, \rho_r \rangle = G$ and $\{\rho_0, \ldots, \rho_r\}$ satisfies the intersection property. All such ρ_r give a string C-group representation $(G, \{\rho_0, \ldots, \rho_{r-2}, \rho_r, \rho_{r-1}\})$ of rank $r+1$. In this process, whenever a ρ_r is found such that $\langle \rho_0, \ldots, \rho_r \rangle < G$ and $\{\rho_0, \ldots, \rho_r\}$ satisfies the intersection property, add the ordered tuple $\{\rho_0, \ldots, \rho_{r-2}, \rho_r, \rho_{r-1}\}$ in $\mathcal{Q}$, provided it is not isomorphic to any element of $\mathcal{Q}$ yet, $o(\rho_{r-1}\rho_r) = 2$ and $o(\rho_{r-2}\rho_r) > 2$.

- At the end of this for loop, we have found all string C-group representations of rank $r+1$ of G and in $\mathcal{Q}$ we have $r+1$ tuples that we can try to extend further.

- Let $\mathcal{P} := \mathcal{Q}$ and continue the while loop.

4.5 Computational times

Table 2 gives, for a given group G, the Time[2] it takes (in seconds) to compute all string C-group representations with the central approach, the linear approach, the rank three algorithm and the algorithm described in Section 4.1 for the higher ranks, the old algorithm (when this time is small enough and the old algorithm was capable of getting the whole result), the number of string C-group representations of rank greater than 3 and the number of string C-group representations of rank three. The time in the "High+Rank3" column is given as a sum of two numbers. The first number is the time it takes to compute string C-group representations of rank at least four with the algorithm of Section 4.1. The second number is the time it takes to compute the rank three ones with the new algorithm described in Section 3. We leave a ??? when the computer was unable to finish the computation.

[2]The timings presented in this section were obtained using MAGMA [1] running on a computer with 8 cores running at 3.9Ghz and 512Gb of RAM at 2.9Ghz. Note that MAGMA does not do parallel computing, it is using one core at a time.

G	Central	Linear	High+Rank3	Old algo	# pol rk ¿ 3	# pol rk 3
M_{11}	0.11s	0.1s	0.04s + 0.05 = 0.09s	0.76s	0	0
M_{12}	5s	2.7s	2.8s + 0.2s = 3s	191s	14	23
J_1	4.2s	2.4s	0.9s + 1.3s = 2.2s	357s	2	148
M_{22}	1s	0.9s	0.22s + 0.04s = 0.26s	414s	0	0
J_2	11s	7.5s	8.5s + 2.1s = 10.6s		17	137
M_{23}	1.7s	1.8s	0.3s + 0.06s = 0.36s		0	0
HS	102s	74s	68s + 7.5s = 75.5s		59	252
J_3	207s	266s	41.2s + 121.7s = 162.9s		2	303
M_{24}	1622s	384s	1258s + 10s = 1268s = 21m8s		157	490
McL	15s	10s	2.09s+0.56s = 2.65s		0	0
He	20635s=5h43m	6216s=1h43m	16717s + 744s = 17461s		76	1188
Ru	127600s	91338s=25h22m	27739s+73518s = 101257s		227	21594
Suz	40322s=11h12m	43539s	41870s + 10439s = 52309s		270	7119
O′N	1360382s = 15.74 days	1536062s	573962s + 2619817s = 37 days		16	6536
Co_3	18531s=5h09m	27570s	28627s + 5522s = 9.5h		895	10586
Co_2			???s + 1305940s = ???		???	60370
Fi_{22}			???s + 2010594s = ???		???	25052

Table 2: Computing times for sporadic groups

5 New results

Two sporadic groups smaller than Co_3, namely Suz and Ru, had apparently never been investigated. The two main reasons were that these groups have a higher permutation representation degree than Co_3 and that they have larger Sylow 2-subgroups (of respective sizes 2^{13} and 2^{14} while the Sylow 2-subgroups of Co_3 have order 2^{10}). We analysed them with our new set of programs.

Our new algorithm also permitted to complete the classification of string C-group representations for the O'Nan group, and to obtain all string C-group representations of rank three for the Conway group Co_2 and the Fischer group Fi_{22}.

We now summarize our findings.

5.1 The Rudvalis group

The Rudvalis group was discovered by Arunas Rudvalis in 1973 [14]. It has order $145,926,144,000 = 2^{14} \cdot 3^3 \cdot 5^3 \cdot 7 \cdot 13 \cdot 29$ and smallest permutation representation degree 4060. It has two conjugacy classes of involutions. It has 21594 string C-group representations of rank three, 227 of rank four and none of higher rank.

5.2 The Suzuki sporadic group

The Suzuki sporadic group was discovered by Michio Suzuki in 1968 [15]. It has order $448,345,497,600 = 2^{13} \cdot 3^7 \cdot 5^2 \cdot 7 \cdot 11 \cdot 13$ and smallest permutation representation degree 1782. It has two conjugacy classes of involutions. It has 7119 string C-group representations of rank three, 257 of rank four, 13 of rank five and none of higher rank.

5.3 The O'Nan group

In 1973, Michael O'Nan provided in [13] strong evidence for the existence of a new sporadic group now called $\mathsf{O'N}$. Later in the seventies, Sims constructed this group with help of a computer (see [6] for a survey of the story of $\mathsf{O'N}$) but his work seems to be unpublished. The group has order $4,608,155,059,20 = 2^9 \cdot 3^4 \cdot 5 \cdot 7^3 \cdot 11 \cdot 19 \cdot 31$ and smallest permutation representation degree 122,760. This makes it a difficult group to deal with even though its 2-Sylows are not that large compared to Suz and Ru.

All its string C-group representations of rank at least four were determined by Connor, Leemans and Mixer [3]. In [2], Connor and Leemans computed the number of regular maps having the O'Nan group as automorphism group, therefore bounding the number of string C-group representations of rank three. However, they were unable to compute all string C-group representations of rank three. Thanks to our new algorithm, we finally fill in this gap in the classification of string C-group representations of the O'Nan group. The O'Nan group has 6536 string C-group representations of rank three, 16 of rank four and none of higher rank.

5.4 The Conway group Co_2

The Conway group Co_2 was discovered by John Horton Conway in 1968 as a subgroup of a group he called .0, among with the other two simple sporadic groups Co_1 and Co_3 [4]. As pointed out by Conway in his paper, the simplicity of these groups was proven by John Thompson. The group has order $42,305,421,312,000 = 2^{18} \cdot 3^6 \cdot 5^3 \cdot 7 \cdot 11 \cdot 23$ and smallest permutation representation degree 2300. It has three conjugacy classes of involutions. It has 60370 string C-group representations of rank three.

5.5 The Fischer group Fi_{22}

The Fischer group Fi_{22} was discovered by Bernd Fischer in 1971 [5] while studying groups generated by 3-transpositions. The group has order $64, 561, 751, 654, 400 = 2^{17} \cdot 3^9 \cdot 5^2 \cdot 7 \cdot 11 \cdot 13$ and smallest permutation representation degree 3510. It has three conjugacy classes of involutions. It has 25052 string C-group representations of rank three.

References

[1] W. Bosma, J. Cannon and C. Playoust, The Magma Algebra System I: the user language, *J. Symbolic Comput.* **3-4** (1997), 235–265, doi:10.1006/jsco.1996.0125.

[2] T. Connor and D. Leemans, Algorithmic enumeration of regular maps, *Ars Math. Contemp.* **10** (2016), 211–222, doi:10.26493/1855-3974.544.1b4.

[3] T. Connor, D. Leemans and M. Mixer, Abstract regular polytopes for the O'Nan group, *Internat. J. Algebra Comput.* **24** (2014), 59–68, doi:10.1142/s0218196714500052.

[4] J. H. Conway, A perfect group of order 8,315,553,613,086,720,000 and the sporadic simple groups, *Proc. Nat. Acad. Sci. U.S.A.* **61** (1968), 398–400, doi:10.1073/pnas.61.2.398.

[5] B. Fischer, Finite groups generated by 3-transpositions. I, *Invent. Math.* **13** (1971), 232–246, doi:10.1007/bf01404633.

[6] D. Gorenstein, The classification of finite simple groups, I. Simple groups and local analysis, *J. Amer. Math. Soc.* **1** (1979), 43–199, doi:10.1090/s0273-0979-1979-14551-8.

[7] M. I. Hartley, An atlas of small regular polytopes, *Periodica Math. Hungar.* **53** (2006), 149–156, doi:10.1007/s10998-006-0028-x.

[8] M. I. Hartley and A. Hulpke, Polytopes derived from sporadic simple groups, *Contrib. Discrete Math.* **5** (2010), 106–118, doi:10.11575/cdm.v5i2.61945.

[9] D. Leemans, String C-group representations of almost simple groups: A survey, in: *Polytopes and discrete geometry*, Amer. Math. Soc., [Providence], RI, volume 764 of *Contemp. Math.*, pp. 157–178, 2021, doi:10.1090/conm/764/15335.

[10] D. Leemans and M. Mixer, Algorithms for classifying regular polytopes with a fixed automorphism group, *Contrib. Discrete Math.* **7** (2012), 105–118, doi:10.11575/cdm.v7i2.62152.

[11] D. Leemans and L. Vauthier, An atlas of abstract regular polytopes for small groups, *Aequationes Math.* **72** (2006), 313–320, doi:10.1007/s00010-006-2843-9.

[12] P. McMullen and E. Schulte, *Abstract regular polytopes*, volume 92, Cambridge University Press, 2002.

[13] M. O'Nan, Some evidence for the existence of a new simple group, *Proc. London Math. Soc. (3)* **32** (1976), 421–479, doi:10.1112/plms/s3-32.3.421.

[14] A. Rudvalis, A new simple group of order $2^{14} \cdot 3^3 \cdot 5^3 \cdot 7 \cdot 13 \cdot 29$, *Notices Amer. Math. Soc.* (1973), A–95.

[15] M. Suzuki, A simple group of order $448, 345, 497, 600$, in: *Theory of Finite Groups (Symposium, Harvard Univ., Cambridge, Mass., 1968)*, Benjamin, New York, 1969 pp. 113–119.