

The Coxeter Group $[5, 3, 3, 3]$ Has No Torsion-Free Subgroup of Index 14400: A Computer-Assisted Result Related to Problem 4.124 of the K3 List

Alper Ferudun*

October 8, 2026

Abstract

Problem 4.124 of the K3 problem list asks whether a hyperbolic integer homology 4-sphere exists, whether an arithmetic one exists, and more generally whether there is a closed hyperbolic 4-manifold with Euler characteristic 2. By a theorem of Belolipetsky, an orientable arithmetic example would be the quotient of $\mathbb{H}^4$ by a torsion-free subgroup of index 28800 of the Coxeter group $W = [5, 3, 3, 3]$ that lies in the rotation subgroup of W . We prove, with computer assistance, that W has no torsion-free subgroup of index 14400. This answers a question that goes back to a remark of Davis from 1985, who thought such a subgroup quite possible. Equivalently: no facet pairing of a single compact regular hyperbolic 120-cell with dihedral angle $2\pi/3$ gives a closed hyperbolic 4-manifold; every torsion-free subgroup of finite index of W has index $14400m$ with $m \geq 2$ (Conder and Maclachlan found one with $m = 8$); and no closed manifold that covers the orbifold $\mathbb{H}^4/W$ has Euler characteristic 1. The proof reduces the question, with complete proofs, to the enumeration of certain involutions of the 14400 flags of the 120-cell. It then rests on two exhaustive enumerations, made by two separately written programs with different symmetry reductions and different pruning rules; both find nothing. No complete gluing was reached in either enumeration, so the result depends on every pruning rule being a necessary condition. We prove this for each of the 13 rules of the first program and each of the 14 rules of the second, and we report controls, mutation tests and partial re-enumerations with a third program. The theorem decides none of the three questions of Problem 4.124. It excludes only those subgroups of index 28800 that correspond to double covers of one-cell manifolds; the case of index 28800 itself is out of reach of our programs, and the problem stays open. This is an unrefereed note.

1 Introduction

1.1 The problem and what is known

Problem 4.124 of the list *K3: A New Problem List in Low-Dimensional Topology* [BKR26], proposed by A. Reid, begins: “Does there exist a hyperbolic integer homology four-sphere? What about an arithmetic one?”¹ Since a homology 4-sphere has Euler characteristic 2, the problem then asks, more generally, whether there is any closed hyperbolic 4-manifold with Euler characteristic 2. A hyperbolic 4-manifold M has volume $\frac{4}{3}\pi^2\chi(M)$, and a closed orientable one has even Euler characteristic [CM05,

*Mercury Software GmbH. Email: alper@mercurycodelab.com. GitHub: <https://github.com/AlperTheKing>. Unrefereed preprint. CC BY 4.0.

¹We quote the statement from record KP-4.124 of the UnsolvedMath dataset [ula26], which reproduces the problem list. The statement and the three remarks of the record agree with the authors’ preliminary version of [BKR26], where the problem is on pp. 292–293; the attribution “Proposed for K3 by: A. Reid” is taken from that version. We did not consult the published volume.

Section 1]. So an orientable example would be a closed orientable hyperbolic 4-manifold of the smallest possible volume. The smallest closed orientable example known has $\chi = 16$ [CM05].

The third remark to the problem in [BKR26] points to the arithmetic case. Let

$$W = [5, 3, 3, 3] = \langle r_1, \dots, r_5 \mid r_i^2, (r_1 r_2)^5, (r_2 r_3)^3, (r_3 r_4)^3, (r_4 r_5)^3, (r_i r_j)^2 \ (|i - j| \geq 2) \rangle$$

be the Coxeter group of the compact hyperbolic Coxeter 4-simplex with diagram 5–3–3–3, and let W^+ be its subgroup of index 2 consisting of the words of even length. Belolipetsky [Bel04, Bel07] proved the following.

Theorem 1.1 (Belolipetsky [Bel07, Theorem 5.5’]; see also [Bel04, Theorem 5.5]). *Let Γ_1 be the group of the smallest arithmetic hyperbolic 4-orbifold, and let M be a compact orientable arithmetic hyperbolic 4-manifold with $\chi(M) \leq 16$. Then M is defined over $\mathbb{Q}[\sqrt{5}]$, and M is isometric to $\Gamma_M \backslash \mathbb{H}^4$ for a torsion-free subgroup $\Gamma_M \leq \Gamma_1$ of index $7200 \chi(M)$.*

By [Bel04, Section 5.6], Γ_1 is the orientation-preserving subgroup of the Coxeter group with diagram 5–3–3–3; that is, $\Gamma_1 = W^+$ up to conjugation. The proof of the uniqueness of the smallest orbifold, on which this identification depends, was corrected in [Bel07]. Hence an orientable arithmetic closed hyperbolic 4-manifold with $\chi = 2$, in particular an arithmetic hyperbolic integer or rational homology 4-sphere, is $\mathbb{H}^4/G$ for a torsion-free subgroup G of index 28800 of W with $G \leq W^+$. This is the reduction stated in [BKR26], which speaks of an index-28800 cover of the reflection group with this diagram. Note the hypotheses: the theorem concerns compact, orientable, arithmetic manifolds. We use Theorem 1.1 only to explain the context. We did not re-verify it, and our theorem does not depend on it.

Belolipetsky also proposed a computation. In [Bel04, Section 5.6] he suggests searching by computer for the torsion-free subgroups of the Coxeter group of index $14400 \chi(M)$, and remarks that “the computation looks more or less realistic”. Conder and Maclachlan [CM05] carried out such a search with low-index subgroup methods. They found a torsion-free subgroup Σ of index $115200 = 8 \cdot 14400$ in W . The manifold $\mathbb{H}^4/\Sigma$ is non-orientable with $\chi = 8$, and its orientable double cover has $\chi = 16$. They note that the index of a torsion-free subgroup of W is divisible by 14400, the order of the finite subgroup $\langle r_1, \dots, r_4 \rangle$, and they close with the remark that “There may still be torsion-free subgroups in Γ_1 of smaller index” (their Γ_1 is our W). What they exclude are torsion-free subgroups of smaller index that arise as an intersection of two subgroups of index at most 480 of W , or of two subgroups of index at most 4 in any of these. So the indices $14400m$ with $1 \leq m \leq 7$ were left open. Long [Lon08] gave further non-orientable examples with $\chi = 8$, that is, of the same index 115200 (we know that paper from its abstract only). His doctoral thesis [Lon07, Chapter 7] is the source closest to the present note. It calls it “an outstanding question” whether W has a torsion-free subgroup of index 14400, and it notes that the manifold of such a subgroup would have Euler characteristic 1. The thesis lists, for each index up to 720, the number of conjugacy classes of subgroups of W , and it gives three torsion-free subgroups of index 115200, none of them contained in W^+ , which it describes as different from the subgroup of Conder and Maclachlan. It contains two statements on the index 14400. Proposition 7.3.1 of the thesis says that there is no epimorphism of W onto $[5, 3, 3]$ with torsion-free kernel. This is the case of normal subgroups of Theorem 1.2: a torsion-free normal subgroup N of index 14400 meets the finite subgroup $\langle r_1, \dots, r_4 \rangle \cong [5, 3, 3]$ trivially, so this subgroup, which has order 14400, maps isomorphically onto W/N . Theorem 7.1 of the thesis says that if $G \leq W$ is torsion-free of index 14400 and N is the largest normal subgroup of W contained in G , then W/N has no epimorphism onto a sporadic simple group. The thesis leaves the question open; its conclusion says that “a possible approach towards determining the minimal index torsion-free subgroup of this group is outlined”.

The question goes back to Davis. In the remarks at the end of [Dav85] he notes that the symmetry groups of the two other tessellations of $\mathbb{H}^4$ by regular 120-cells, which are $[5, 3, 3, 3]$ and $[5, 3, 3, 4]$, have

Euler characteristics $1/14400$ and $17/28800$. He concludes that the second group has no torsion-free subgroup of index 14400, and he writes of the first that it “quite possibly does”. Theorem 1.2 below shows that it does not. In geometric form the question was taken up by Ratcliffe and Tschantz [RT00, Section 1.3], who consider the manifolds that can be glued from one or two regular 120-cells with dihedral angle $2\pi/3$. They write that a purely combinatorial search for such manifolds “is essentially intractable”, and that their searches for side pairings under “some simple restrictions” found no manifold. Theorem 1.2 settles the case of one cell by a complete search; the case of two cells is still out of reach (Section 8).

A closed hyperbolic 4-manifold with Euler characteristic 1 would be non-orientable, because the Euler characteristic of a closed orientable one is even. Parity alone does not exclude it. In the same section Ratcliffe and Tschantz report two closed non-orientable hyperbolic 4-manifolds with Euler characteristic 17, glued from two right-angled 120-cells; in terms of Davis’s remark these belong to the group $[5, 3, 3, 4]$, with index 28800. According to Chen [Che25, Section 5.3], they are the only closed hyperbolic 4-manifolds of odd Euler characteristic that he could find in the literature.

Emery [Eme14] proved that for $n > 4$ there is no compact arithmetic hyperbolic n -manifold M with $|\chi(M)| = 2$, and he notes that the case $n = 4$ is not solved and is reduced by Belolipetsky’s theorem to the Coxeter group $[5, 3, 3, 3]$. In Martelli’s survey [Mar18] the Conder–Maclachlan manifold is the smallest closed orientable hyperbolic four-manifold known, and the list of open questions includes: “What is the smallest volume of a closed oriented hyperbolic four-manifold?”

For the Coxeter group $[5, 3, 3, 5]$ the analogous question has a positive answer. The Davis manifold [Dav85] is glued from one 120-cell with dihedral angle $2\pi/5$ and corresponds to a torsion-free subgroup of index 14400 of $[5, 3, 3, 5]$ [CM05, Section 2]. According to [Mar18, Section 2], Ratcliffe and Tschantz [RT01] proved that it is the unique smallest manifold covering that simplex orbifold; we did not consult [RT01]. In dimension 3 the same method, the determination of the torsion-free subgroups of minimum index of a string Coxeter group, is classical. The manifolds glued from one Platonic solid were classified by several authors; see Everitt [Eve04] and the references there. Conder and Liversidge [CL23] treat all infinite groups $[p, q, r]$ with $p, q, r \geq 3$, $1/p + 1/q \geq 1/2$ and $1/q + 1/r \geq 1/2$ (we know that paper from its abstract only). The present note applies this method to the rank-5 group $[5, 3, 3, 3]$, whose cell group has order 14400.

Two papers with related titles treat different questions. Ma and Zheng [MZ21] classify the orientable small covers of the right-angled 120-cell (dihedral angle $\pi/2$); these are closed hyperbolic 4-manifolds of volume $16 \cdot \frac{34}{3}\pi^2$, that is, with $\chi = 136$. Conder and Kellerhals [CK22] construct cusped hyperbolic manifolds of dimension at most 5 from the Coxeter groups $[3, \dots, 3, 6]$. We know both papers from their abstracts.

1.2 The result

Let $H = \langle r_1, r_2, r_3, r_4 \rangle \leq W$. It is the Coxeter group $[5, 3, 3]$ of order 14400, the symmetry group of the regular 120-cell. The smallest index that a torsion-free subgroup of W could have is therefore 14400. A subgroup of this index is the same thing as a closed hyperbolic 4-manifold glued from a single 120-cell (Theorem 2.5).

Theorem 1.2 (computer-assisted). *The Coxeter group $W = [5, 3, 3, 3]$ has no torsion-free subgroup of index 14400.*

The proof is computer-assisted, and we state precisely what certifies it. Sections 2 and 3 reduce the statement, with complete proofs, to the following: there is no involution ρ of the set H that commutes with right multiplication by r_1, r_2, r_3 , satisfies $(\rho r_4)^3 = 1$, and makes the four finite groups $P_i = \langle r_j : j \neq i \rangle$, $i \leq 4$, act freely (Proposition 3.4). Sections 4 and 5 prove that two search programs

	program A	program B
slices (root classes / root types)	154	140
jobs (one log each)	383	935
unit of work	search nodes	trial gluings
total work	1 596 433 384	326 322 489 739
complete gluings reached	0	0
solutions	0	0
maximal depth of the search tree	6	6
cpu time	36.48 h	65.15 h
pruning rules (each proved necessary)	13	14

Table 1: The two enumerations behind Theorem 1.2. “Complete gluings reached” counts the complete candidate involutions handed to the final test. The slices of the two programs are different decompositions of the same set of solutions (Remark 3.13).

are exhaustive for such involutions, up to the relabellings $x \mapsto ax$, $a \in H$. The remaining part is two complete enumerations:

- *program A* (source `fsearch_production.c`): 154 slices, 383 jobs, 1 596 433 384 search nodes, 36.48 cpu-hours, 0 solutions;
- *program B* (sources `fps_final.c` and `fps_v7.c`): 140 slices, 935 jobs, 326 322 489 739 trial gluings, 65.15 cpu-hours, 0 solutions.

Both were run on an Apple M3 Max (macOS 26.6.2) with binaries compiled by Apple clang 21.0.0 and tables generated with Python 3.13.5. Each of the 383+935 logs ends with the normal termination line of its own job and reports 0 solutions. All logs are in the reproducibility package. Their aggregate SHA-256 checksums are

```

program A  22bb406ca14479324db761d10c4e6cb78bf4a799f3f3d47fbb5d3aca97c1912c
program B  e93013be9073d3e89fcb906648716dc9b646e0eaabc902f82cfda679822b2c9f

```

(the two recipes are given in Section 6). Table 1 collects the main numbers. A reader who wants to check the result has to check the mathematics of Sections 2 to 5 against the two sources, and either trust or repeat the 102 cpu-hours of computation. Section 7 describes what has been done to make the count 0 credible, and what has not.

Theorem 1.2 has three equivalent forms.

- Corollary 1.3.** (a) (Geometric form.) *Let P be the compact regular hyperbolic 120-cell with dihedral angle $2\pi/3$. No facet pairing of P gives a closed hyperbolic 4-manifold* (Definition 2.4).
- (b) (Index form.) *Every torsion-free subgroup of finite index of W has index $14400m$ with $m \geq 2$. The smallest index of a torsion-free subgroup of W is $14400m_0$ with $2 \leq m_0 \leq 8$.*
- (c) (Euler characteristic form.) *No finite-sheeted manifold cover of the orbifold $\mathbb{H}^4/W$, that is, no quotient $\mathbb{H}^4/G$ by a torsion-free subgroup $G \leq W$ of finite index, has Euler characteristic 1.*

Form (a) is a statement about isometric facet pairings whose quotient metric is hyperbolic. It is not a statement about arbitrary topological identifications of the facets of a 120-cell. In form (c) the word “finite-sheeted” cannot be omitted, since $\mathbb{H}^4$ itself is a manifold cover with Euler characteristic 1. The upper bound in (b) is the theorem of Conder and Maclachlan.

What this does and does not say about Problem 4.124. Theorem 1.2 decides none of the three questions of Problem 4.124. Those questions concern Euler characteristic 2, which for covers of $\mathbb{H}^4/W$ means index 28800. Our theorem is a related result on the reduction in the third remark to the problem. It excludes those torsion-free subgroups of index 28800 that are contained in a torsion-free subgroup of index 14400. Geometrically these are the double covers of manifolds glued from one 120-cell. If a one-cell manifold existed, it would be non-orientable (its Euler characteristic would be 1), and its orientation cover would be an arithmetic closed orientable hyperbolic 4-manifold with $\chi = 2$. Theorem 1.2 shows that no example arises in this way. It says nothing about the remaining subgroups of index 28800, which correspond to manifolds glued from two 120-cells that do not cover a one-cell manifold. It says nothing about closed hyperbolic 4-manifolds that do not cover $\mathbb{H}^4/W$, in particular nothing about non-arithmetic ones. Problem 4.124 stays open. The direct content of the theorem concerns the closing remark of Conder and Maclachlan, the computer search proposed by Belolipetsky, and the question of the smallest closed hyperbolic 4-manifold: the lower bound for the smallest index of a torsion-free subgroup of W goes from 14400 to 28800.

We tried to enumerate the case of index 28800 with both programs. Both attempts are incomplete, and we estimate that a complete enumeration with the present methods needs between 10^3 and 10^4 core-hours (Section 8). No statement about index 28800 is claimed.

1.3 Method and organization

A subgroup $G \leq W$ of index 14400 in which H acts freely on the cosets is described by one permutation ρ of the set H , the action of r_5 on $G \backslash W \cong H$. Geometrically, ρ pairs the 120 dodecahedral facets of the 120-cell and prescribes one of the 120 possible gluing maps for each pair; the relation $(r_4 r_5)^3 = 1$ says that three cells meet around every pentagon. Torsion-freeness is equivalent to the free action of the five maximal finite subgroups $P_1, \dots, P_5$ (Lemma 3.2). The search space is large (the first facet alone has $119 \cdot 120$ possible gluings), but the relation around the pentagons propagates strongly.

Both programs glue facets one after the other and discard a partial gluing only if it violates a condition that every solution extending it would have to satisfy. They differ in the conditions used. Program A propagates the pentagon relation and filters by the conjugacy class (“type”) of the gluing; program B propagates the free action of the vertex group $P_1 \cong S_5$ and uses the types only through one exact filter. They also differ in the symmetry reduction: program A uses 154 slices given by classes under conjugation by $K = \langle r_1, r_2, r_3 \rangle$, program B uses 140 slices given by classes under conjugation and inversion, and for 47 of its 53 expensive slices a further lex-leader reduction under the stabilizer of the first gluing. The two programs were written separately from the statement of Proposition 3.4, with different tables and data structures.

Section 2 introduces the group, the 120-cell and the honeycomb, and proves the equivalence of the geometric form. Section 3 contains the reduction. Sections 4 and 5 describe the two programs and prove their completeness, rule by rule. Section 6 documents the computation, Section 7 the controls and rejection tests, including the weaknesses that the verification found and how each was closed. Section 8 discusses index 28800.

Conventions. Groups act on the right on coset spaces and on the left on $\mathbb{H}^4$. A *hyperbolic manifold* is a connected Riemannian manifold of constant sectional curvature -1 ; *closed* means compact without boundary. $\varphi = (1 + \sqrt{5})/2 = 2 \cos(\pi/5)$. File names refer to the reproducibility package that accompanies this note; line numbers refer to the source files in that package.

$ T $	orders $ W_T $ (the subset T is written before the colon)	$\sum_T 1/ W_T $
0	$\emptyset: 1$	1
1	five groups of order 2	5/2
2	12: 10, 23: 6, 34: 6, 45: 6, the other six: 4	21/10
3	123: 120, 124: 20, 125: 20, 134: 12, 135: 8, 145: 12, 234: 24, 235: 12, 245: 12, 345: 24	13/20
4	1234: 14400, 1235: 240, 1245: 60, 1345: 48, 2345: 120	721/14400

Table 2: The proper standard parabolic subgroups of $W = [5, 3, 3, 3]$. The orders follow from the types of the subdiagrams ($H_4, H_3, A_4, A_3, A_2, I_2(5), A_1$ and their products). They were also recomputed with exact matrix groups by the scripts `verification_engine/own/h4_classes.py` and `independent_run_2/r2_group.py`.

2 The group, the 120-cell and the honeycomb

2.1 Notation and standard facts

Let $m_{12} = 5, m_{23} = m_{34} = m_{45} = 3$ and $m_{ij} = 2$ for $|i - j| \geq 2$. For $T \subseteq \{1, \dots, 5\}$ let $W_T = \langle r_i : i \in T \rangle$; we write W_{124} for $W_{\{1,2,4\}}$ and so on. We use the following subgroups throughout:

$$H = W_{1234}, \quad K = W_{123}, \quad D = W_{12},$$

$$P_i = W_{\{1, \dots, 5\} \setminus \{i\}} \quad (1 \leq i \leq 5), \quad H_i = W_{\{1,2,3,4\} \setminus \{i\}} \quad (1 \leq i \leq 4).$$

So $P_5 = H, H_4 = K$, and $H_i \leq P_i \cap H$. By the general theory of Coxeter groups, W_T is the Coxeter group of the subdiagram on T [Bou02, Ch. IV, §1], [Hum90, Chapter 5]. Every proper subdiagram is a diagram of a finite Coxeter group. The orders of the 31 proper subgroups W_T are listed in Table 2. In particular

$$|H| = 14400, \quad |K| = 120, \quad |D| = 10, \quad |P_4| = 240, \quad |P_3| = 60, \quad |P_2| = 48, \quad |P_1| = 120,$$

since H is of type H_4 , K of type H_3 , $P_4 = K \times \langle r_5 \rangle$, $P_3 = D \times \langle r_4, r_5 \rangle$, $P_2 = \langle r_1 \rangle \times \langle r_3, r_4, r_5 \rangle$ and $P_1 = \langle r_2, \dots, r_5 \rangle \cong S_5$. The subgroups H_1, H_2, H_3, H_4 have orders 24, 12, 20, 120, so that $[P_i : H_i] = 5, 4, 3, 2$ for $i = 1, 2, 3, 4$.

The simplex. The matrix $(-\cos(\pi/m_{ij}))$, with $m_{ii} = 1$, is tridiagonal. Its leading principal minors are 1, $(3 - \varphi)/4$, $(2 - \varphi)/4$, $(5 - 3\varphi)/16$ and $(3 - 2\varphi)/16$. The first four are positive and the last is negative, so the matrix has signature $(4, 1)$; all its proper principal submatrices are positive definite, because the proper subdiagrams are of finite type. Hence there is a compact simplex $\Delta \subset \mathbb{H}^4$, unique up to isometry, bounded by hyperplanes $\Pi_1, \dots, \Pi_5$ such that the dihedral angle between Π_i and Π_j is π/m_{ij} . It is one of the five compact Coxeter simplices of $\mathbb{H}^4$ (Lannér's list); see [Hum90, Chapter 6], [Dav08, Chapter 6], [CM05, Section 2]. We identify r_i with the reflection of $\mathbb{H}^4$ in Π_i . The following facts are standard [Vin85], [Dav08, Chapter 6], [Rat19, Chapter 7].

(G1) The reflections in $\Pi_1, \dots, \Pi_5$ generate a discrete subgroup of $\text{Isom}(\mathbb{H}^4)$ with the presentation of W . The simplices $w\Delta$, $w \in W$, called *chambers*, cover $\mathbb{H}^4$, they have pairwise disjoint interiors, and $w \mapsto w\Delta$ is injective. Every W -orbit meets Δ in exactly one point.

(G2) The stabilizer in W of a point $y \in \Delta$ is $W_{T(y)}$, where $T(y) = \{i : y \in \Pi_i\}$.

In particular W is infinite, because Δ is compact and $\mathbb{H}^4$ is not. We write $F_i = \Delta \cap \Pi_i$ for the facets of Δ , v_i for the vertex of Δ opposite to F_i , and Π_i^+ for the closed half-space bounded by Π_i that contains Δ . A *mirror* is a hyperplane $w\Pi_i$; it is the fixed hyperplane of the reflection $wr_iw^{-1} \in W$. A mirror

never meets the interior of a chamber: if $s = wr_iw^{-1}$ fixed an interior point of $w'\Delta$, the chambers $w'\Delta$ and $sw'\Delta$ would have a common interior point, so $s = 1$ by (G1). Hence every chamber lies in one closed half-space of every mirror. Finally $r_5 \notin H$, because otherwise $W = H$ would be finite.

2.2 The polytope P_0

Let

$$P_0 = \bigcup_{h \in H} h\Delta, \quad \Phi_0 = P_0 \cap \Pi_5.$$

The group H fixes v_5 , the common point of $\Pi_1, \dots, \Pi_4$. By (G1) the 14400 simplices $h\Delta$ are distinct and have disjoint interiors. A *flag* of a convex polytope Q of dimension n is a chain $Q_0 \subset Q_1 \subset \dots \subset Q_{n-1}$ of faces with $\dim Q_j = j$.

- Lemma 2.1.** (i) $P_0 = \bigcap_{h \in H} h(\Pi_5^+)$. So P_0 is a compact convex polytope with non-empty interior.
- (ii) For $h \in H$ we have $h\Pi_5 = \Pi_5$ if and only if $h \in K$. The polytope P_0 has exactly 120 facets, namely the sets $h\Phi_0$ with $hK \in H/K$, and $\Phi_0 = \bigcup_{\kappa \in K} \kappa F_5$.
- (iii) Φ_0 has 12 two-dimensional faces, which are pentagons and are permuted transitively by K . The dihedral angle of P_0 along every two-dimensional face is $2\pi/3$.
- (iv) The isometries of $\mathbb{H}^4$ that map P_0 onto itself are exactly the elements of H , and H acts simply transitively on the flags of P_0 . Every isometry of the facet Φ_0 onto itself is the restriction of exactly one element of K .

So P_0 is a regular polytope with 120 dodecahedral facets and dihedral angle $2\pi/3$: a compact regular hyperbolic 120-cell. The 14400 simplices $h\Delta$ are its characteristic simplices in the sense of Coxeter [Cox73], and they correspond to its flags. We call the elements of H *flags*, the cosets xK *facets*, and we say that the flag x lies in the facet xK ; the facet xK corresponds to the facet $x\Phi_0$ of P_0 .

Proof. (i) Let $h \in H$. The mirror $h\Pi_5$ does not contain $v_5 = hv_5$, because $v_5 \notin \Pi_5$. The chamber Δ lies in one closed half-space of $h\Pi_5$ and contains v_5 , and v_5 lies in $h(\Pi_5^+)$. So $\Delta \subseteq h(\Pi_5^+)$ for all $h \in H$, and therefore $h'\Delta \subseteq h'h(\Pi_5^+)$ for all $h, h' \in H$. This proves $P_0 \subseteq Q := \bigcap_h h(\Pi_5^+)$. Conversely let $x \in Q$ and let $C = \Pi_1^+ \cap \dots \cap \Pi_4^+$, a cone with apex v_5 ; then $\Delta = C \cap \Pi_5^+$. The chambers that contain v_5 are the simplices $h\Delta$, $h \in H$, by (G1) and (G2). A small ball around v_5 meets only finitely many chambers, and all of them contain v_5 . So one of them, say $h\Delta$ with $h \in H$, contains an initial piece of the geodesic segment from v_5 to x . Then the cone hC contains the whole segment, so $h^{-1}x \in C$. As Q is H -invariant, $h^{-1}x \in Q \subseteq \Pi_5^+$. Hence $h^{-1}x \in \Delta$ and $x \in P_0$.

(ii) For $i \leq 3$ the reflection r_i commutes with r_5 , so it maps the mirror Π_5 of r_5 to itself. Hence K preserves Π_5 . Conversely let $h \in H$ with $h\Pi_5 = \Pi_5$. The line $L = \Pi_1 \cap \Pi_2 \cap \Pi_3$ contains v_4 and v_5 . It is preserved by r_5 , which commutes with r_1, r_2, r_3 , and r_5 fixes $v_4 \in L$ but not $v_5 \in L$. So r_5 acts on L as the reflection in the point v_4 , and L meets Π_5 orthogonally in v_4 . Thus v_4 is the point of Π_5 closest to v_5 , and h , which fixes v_5 and preserves Π_5 , fixes v_4 . Then h fixes the edge $[v_4, v_5]$ of Δ pointwise, and $h \in W_{123} = K$ by (G2).

Since K fixes v_5 , the half-space $h(\Pi_5^+)$ depends only on the coset hK . By (i), P_0 is the intersection of 120 half-spaces with distinct boundary hyperplanes, permuted transitively by H . The hyperplane Π_5 meets P_0 in $\Phi_0 \supseteq F_5$, which is three-dimensional. So each of the 120 hyperplanes carries a facet, and there are no other facets. It remains to show $\Phi_0 \subseteq \bigcup_{\kappa \in K} \kappa F_5$; the other inclusion is clear. Let $x \in \Phi_0$ be a point that lies in no face of codimension ≥ 2 of a chamber. Such points are dense in Φ_0 . Let $x \in h\Delta$ with $h \in H$. A mirror does not meet the interior of a chamber, so x lies in the relative

interior of exactly one facet hF_i of $h\Delta$, and the mirror Π_5 , which passes through x and misses the interior of $h\Delta$, equals $h\Pi_i$. Then $r_5 = hr_i h^{-1}$. Since $r_5 \notin H$, we get $i = 5$, hence $h\Pi_5 = \Pi_5$, $h \in K$ and $x \in hF_5$. The union $\bigcup_{\kappa} \kappa F_5$ is closed, so it contains Φ_0 .

(iii) We repeat the argument one dimension lower, inside Π_5 . The group K acts on Π_5 and fixes v_4 , and Φ_0 is the union of the 120 tetrahedra κF_5 . Two tetrahedra κF_5 and $\kappa r_i F_5$, $i \leq 3$, lie on different sides of their common triangle, so the relative interior of that triangle is interior to Φ_0 . On the other hand $\Phi_0 \subseteq P_0 \subseteq \kappa r_4(\Pi_5^+)$ by (i), and the hyperplane $\kappa r_4 \Pi_5$ meets Π_5 in the plane $\kappa(\Pi_4 \cap \Pi_5)$, because r_4 fixes $\Pi_4 \cap \Pi_5$ pointwise; so the triangle $\kappa(F_4 \cap F_5)$ lies in the boundary of Φ_0 in Π_5 . Hence this boundary is the union of the triangles $\kappa(F_4 \cap F_5)$, and every two-dimensional face of Φ_0 is the image under some $\kappa \in K$ of the face Ψ_0 that contains $F_4 \cap F_5$. The line $\Pi_1 \cap \Pi_2 \cap \Pi_5$ through v_3 and v_4 is orthogonal to the plane $\Pi_4 \cap \Pi_5$, because Π_1 and Π_2 are orthogonal to Π_4 and to Π_5 . So, as in (ii), an element of K that preserves the plane $\Pi_4 \cap \Pi_5$ fixes v_3 , v_4 and v_5 and lies in $W_{12} = D$ by (G2). The triangles $\kappa(F_4 \cap F_5)$ that lie in this plane are those with $\kappa \in D$, so $\Psi_0 = \bigcup_{d \in D} d(F_4 \cap F_5)$, and there are $|K|/|D| = 12$ faces. The triangle $F_4 \cap F_5$ has the vertices v_1, v_2, v_3 , and D is a dihedral group of order 10 of isometries of the plane $\Pi_4 \cap \Pi_5$ fixing v_3 . The reflection r_1 fixes v_2 and preserves the line through v_1 and v_2 , so $[v_1, v_2] \cup r_1[v_1, v_2]$ is a segment with midpoint v_2 . Hence Ψ_0 is a pentagon with centre v_3 , with the five vertices dv_1 and the five edge midpoints dv_2 , $d \in D$.

The facets $\Phi_0 \subset \Pi_5$ and $r_4 \Phi_0 \subset r_4 \Pi_5$ of P_0 both contain $F_4 \cap F_5$, hence Ψ_0 . The polytope P_0 lies in the wedge $\Pi_5^+ \cap r_4(\Pi_5^+)$, and this wedge is the union of the two wedges $\Pi_4^+ \cap \Pi_5^+$ and $r_4(\Pi_4^+ \cap \Pi_5^+)$ of angle $\pi/3$. So the dihedral angle of P_0 along Ψ_0 is $2\pi/3$. Every two-dimensional face of P_0 lies in a facet $h\Phi_0$ and is therefore the image of Ψ_0 under an element of H .

(iv) An isometry of $\mathbb{H}^n$ that preserves a convex n -polytope Q and a flag of Q is the identity. This follows by induction on n : the isometry preserves the hyperplane of the facet Q_{n-1} and is the identity on it by induction (for $n = 1$, an isometry of a line that preserves a segment and fixes one of its end points is the identity); so it is the identity or the reflection in that hyperplane, and the reflection does not preserve Q . Therefore the group of isometries preserving Q has at most as many elements as Q has flags. By (iii), Φ_0 has $12 \cdot 10 = 120$ flags, and K acts faithfully on Π_5 , because an element of K that is the identity on Π_5 and fixes v_5 is trivial. So the isometries of Π_5 preserving Φ_0 are exactly the restrictions of the 120 elements of K . By (ii), P_0 has $120 \cdot 120 = 14400 = |H|$ flags, and H preserves P_0 . So the isometries preserving P_0 are exactly the elements of H , and H is simply transitive on the flags. $\square$

Remark 2.2. A compact regular 120-cell in $\mathbb{H}^4$ with dihedral angle $2\pi/3$ is congruent to P_0 . Indeed, its characteristic simplex has the dihedral angles $\pi/5, \pi/3, \pi/3$ at the hyperplanes through the centre, the angle $\pi/3$ (half the dihedral angle of the polytope) at the facet hyperplane, and right angles otherwise [Cox73]; a compact hyperbolic simplex is determined up to isometry by its dihedral angles. So “the compact regular hyperbolic 120-cell with dihedral angle $2\pi/3$ ” is well defined up to isometry. In what follows P always denotes the polytope P_0 .

Lemma 2.3. *The polytopes wP_0 , $w \in W$, called tiles, cover $\mathbb{H}^4$ and have pairwise disjoint interiors, and $wP_0 = w'P_0$ if and only if $wH = w'H$. The facets of the tile wP_0 are the sets $wh\Phi_0$, $h \in H$. The reflection in the hyperplane of $wh\Phi_0$ is $whr_5(wh)^{-1} \in W$; it maps wP_0 to the tile whr_5P_0 , which is the only other tile that contains $wh\Phi_0$. Every tile can be reached from P_0 by finitely many such reflections in facets. The group of all isometries of $\mathbb{H}^4$ that permute the tiles is W .*

Proof. The first sentence follows from (G1), because P_0 is the union of the chambers $h\Delta$, $h \in H$. The facets of wP_0 are given by Lemma 2.1(ii). The hyperplane of $wh\Phi_0$ is $wh\Pi_5$, with reflection $whr_5(wh)^{-1}$, which maps $wP_0 = whP_0$ to whr_5P_0 . Two tiles with a common facet lie on different sides of it, and at most one tile can lie on a given side, so whr_5P_0 is the only other tile through $wh\Phi_0$. Every $w \in W$ is a product $h_1 r_5 h_2 r_5 \cdots r_5 h_k$ with $h_j \in H$, and the tiles $h_1 r_5 \cdots h_j r_5 P_0$, $j = 0, 1, \dots$,

form a chain in which each tile is the mirror image of the previous one in a common facet. Finally, an isometry f that permutes the tiles maps P_0 to some wP_0 ; then $w^{-1}f$ preserves P_0 and lies in H by Lemma 2.1(iv), so $f \in W$. $\square$

The set of tiles is the regular honeycomb $\{5, 3, 3, 3\}$ of Coxeter [Cox56]: three tiles meet at every two-dimensional face, since the dihedral angle is $2\pi/3$.

2.3 Facet pairings and the geometric form

Definition 2.4. Let $P = P_0$. A *facet pairing* of P consists of an involution $F \mapsto F'$ of the set of the 120 facets of P and, for every facet F , an isometry $\phi_F: F \rightarrow F'$ of F onto F' , such that $\phi_{F'} = \phi_F^{-1}$. Let $\sim$ be the equivalence relation on P generated by $y \sim \phi_F(y)$ for all facets F and all $y \in F$, let $M = P/\sim$ with the quotient topology and $q: P \rightarrow M$ the quotient map. The facet pairing *gives a closed hyperbolic manifold* if M is a topological manifold that carries a Riemannian metric of constant curvature -1 for which q restricted to the interior of P is a local isometry.

A facet may a priori be paired with itself ($F' = F$). The proof below shows that this does not happen when the pairing gives a hyperbolic manifold.

Theorem 2.5. (a) *If a facet pairing of P gives a closed hyperbolic manifold M , then M is isometric to $\mathbb{H}^4/G$ for a torsion-free subgroup $G \leq W$ of index 14400.*

(b) *If $G \leq W$ is a torsion-free subgroup of index 14400, then $\mathbb{H}^4/G$ is a closed hyperbolic manifold, P is a fundamental polytope for G , and there is a facet pairing of P that gives $\mathbb{H}^4/G$.*

In particular, Theorem 1.2 is equivalent to Corollary 1.3(a).

Proof. (a) *Step 1: developing.* M is compact, hence complete, so there is a Riemannian covering $p: \mathbb{H}^4 \rightarrow M$ whose group G of deck transformations is a discrete group of isometries acting freely, with $M = \mathbb{H}^4/G$. The polytope P is simply connected, so q lifts to a continuous map $\tilde{q}: P \rightarrow \mathbb{H}^4$ with $p\tilde{q} = q$. On the interior of P the map $\tilde{q}$ is a local isometry between open subsets of $\mathbb{H}^4$. Such a map is locally the restriction of an isometry of $\mathbb{H}^4$, two isometries that agree on a non-empty open set are equal, and $\text{int}(P)$ is connected; so $\tilde{q}$ is the restriction of one isometry ψ . Replacing p by $p\psi$ and G by $\psi^{-1}G\psi$, we may assume that $q = p|_P$. Since q is onto, the polytopes gP , $g \in G$, cover $\mathbb{H}^4$. An interior point of P lies in no facet, so its $\sim$ -class is a single point and q is injective on $\text{int}(P)$; as G acts freely, $\text{int}(P) \cap g\text{int}(P) = \emptyset$ for $g \neq 1$. So the sets gP form a locally finite tessellation of $\mathbb{H}^4$ on which G acts simply transitively.

Step 2: the tile across a facet. Let F be a facet of P and Π its hyperplane. We claim that there is an element $g_F \in G$, $g_F \neq 1$, such that g_FP lies on the other side of Π , F is a common facet of P and g_FP , $g_F(F') = F$ and $\phi_F = g_F^{-1}|_F$. Choose a point in the relative interior of F and a small ball B around it such that $B \cap P$ is a half-ball and $B \cap \Pi \subseteq \text{relint}(F)$. Only finitely many tiles meet B , and the tiles gP with $g \neq 1$ cover $B \setminus P$. Every point of the disc $B \cap \Pi$ is a limit of points of $B \setminus P$, so the disc is covered by the finitely many closed sets $gP \cap \Pi$, $g \neq 1$. One of them has non-empty interior in Π . So there are $g \neq 1$ and a non-empty open subset U of Π with $U \subseteq \text{relint}(F) \cap gP$. The convex set $P \cap gP$ has empty interior and contains U , so it lies in Π ; the points of U are boundary points of the convex polytope gP , so Π supports gP , $gP \cap \Pi$ is a facet of gP , and gP lies on the side of Π opposite to P . For $y \in U$ the points y and $g^{-1}y$ of P have the same image in M , so $y \sim g^{-1}y$. The $\sim$ -class of a point y in the relative interior of F is $\{y, \phi_F(y)\}$, because $\phi_F(y)$ lies in the relative interior of F' and in no other facet. As $g^{-1}y \neq y$, we get $g^{-1}y = \phi_F(y)$ for all $y \in U$. The isometric embeddings $g\phi_F$ and the inclusion of F into $\mathbb{H}^4$ agree on the open subset U of F , so they agree on F . Hence $g(F') = F$ and $\phi_F = g^{-1}|_F$, and we put $g_F = g$. If $F' = F$, then g maps the compact convex set F onto itself and fixes its circumcentre, which contradicts the freeness of the action. So $F' \neq F$.

Step 3: rigidity. We claim $g_FP = s(P)$, where s is the reflection in Π . Both $A = s(P)$ and $A' = g_FP$ are congruent to P , have F as a facet and lie on the same side of Π . Let u be an isometry with $u(A) = A'$. By Lemma 2.1 the group of isometries preserving A' is transitive on its facets, so we may assume $u(F) = F$. Again by Lemma 2.1(iv), applied to A , the isometry $(u|_F)^{-1}$ of F is the restriction of an isometry k that preserves A . Then uk maps A to A' and is the identity on F , so uk is the identity or s . It is not s , since $s(A) = P$ lies on the other side. Hence $A' = A$.

Step 4: the tessellation is the honeycomb. By Steps 2 and 3, applied to the tile gP and the pairing transported by g , the set $\mathcal{T} = \{gP : g \in G\}$ contains with a tile its mirror images in all its facets. It contains $P = P_0$. By Lemma 2.3 it contains every tile wP_0 , $w \in W$. These already cover $\mathbb{H}^4$ with disjoint interiors, so $\mathcal{T} = \{wP_0 : w \in W\}$. Every $g \in G$ permutes $\mathcal{T}$, hence $G \leq W$ by Lemma 2.3. The group G acts simply transitively on the tiles, that is, on W/H . So $W = GH$ and $G \cap H = 1$, and $[W : G] = |H| = 14400$. A discrete group of isometries of $\mathbb{H}^4$ that acts freely is torsion-free, because a non-trivial element of finite order fixes the centre of an orbit of the finite group it generates.

(b) W is discrete, so point stabilizers in W are finite, and the torsion-free group G acts freely on $\mathbb{H}^4$. As $[W : G] < \infty$ and Δ is compact, $\mathbb{H}^4/G$ is a closed hyperbolic manifold. For every $w \in W$ the finite group wHw^{-1} meets G trivially, so H acts freely on $G \backslash W$ from the right; since $|G \backslash W| = |H|$, there is one orbit, $W = GH$, and G acts simply transitively on the tiles wP_0 . For a facet F of P let $g_F \in G$ be the element such that g_FP is the tile on the other side of F (Lemma 2.3). Then $F' = g_F^{-1}(F)$ is a facet of P , the tile on the other side of F' is $g_F^{-1}P$, so $g_{F'} = g_F^{-1}$, and $\phi_F = g_F^{-1}|_F$ defines a facet pairing with $\phi_{F'} = \phi_F^{-1}$. Let $p : \mathbb{H}^4 \rightarrow \mathbb{H}^4/G$ be the projection. Then $p|_P$ is onto and constant on $\sim$ -classes. We show that $p(y) = p(y')$ for $y, y' \in P$ implies $y \sim y'$. Let $y' = gy$ with $g \in G$. The point y' lies in the tiles P and gP . Let $c\Delta$ be a chamber that contains y' . By (G1) and (G2) the chambers that contain y' are the chambers $cu\Delta$ with $u \in W_T$, $T = T(c^{-1}y')$, and W_T is generated by the r_i with $i \in T$. For $i \in T$ the chambers $cu\Delta$ and $cur_i\Delta$ lie in the same tile if $i \leq 4$, and in the two tiles cuP_0 and cur_5P_0 with the common facet $cu\Phi_0 \ni y'$ if $i = 5$. Every tile that contains y' contains a chamber that contains y' . It follows that there is a chain of tiles $P = T_0, T_1, \dots, T_k = gP$, all containing y' , such that T_{j-1} and T_j are equal or have a common facet that contains y' . Write $T_j = g_jP$ with $g_j \in G$, $g_0 = 1$, $g_k = g$, and put $y_j = g_j^{-1}y' \in P$. If $T_{j-1} \neq T_j$ have the common facet $E \ni y'$, then $F = g_{j-1}^{-1}E$ is a facet of P containing y_{j-1} , the tile across F is $g_{j-1}^{-1}g_jP$, so $g_{j-1}^{-1}g_j = g_F$ and $y_j = g_F^{-1}y_{j-1} = \phi_F(y_{j-1})$. Hence $y' = y_0 \sim y_1 \sim \dots \sim y_k = y$. So p induces a continuous bijection of the compact space $P/\sim$ onto the Hausdorff space $\mathbb{H}^4/G$, which is a homeomorphism, and p is a local isometry on $\text{int}(P)$. Thus the pairing gives the closed hyperbolic manifold $\mathbb{H}^4/G$. $\square$

2.4 Euler characteristic, and the forms (b) and (c)

Proposition 2.6. *Let $G \leq W$ be a torsion-free subgroup of finite index n . Then 14400 divides n , the quotient $\mathbb{H}^4/G$ is a closed hyperbolic manifold, and $\chi(\mathbb{H}^4/G) = n/14400$.*

Proof. For $T \subsetneq \{1, \dots, 5\}$ the finite group W_T acts freely on $G \backslash W$ from the right, because a point stabilizer is of the form $W_T \cap w^{-1}Gw$. So $|W_T|$ divides n ; for $W_T = H$ this gives $14400 \mid n$. As in the proof of Theorem 2.5(b), G acts freely on $\mathbb{H}^4$ and $\mathbb{H}^4/G$ is a closed hyperbolic manifold. The faces of the chambers form a W -invariant triangulation of $\mathbb{H}^4$. For $T \subsetneq \{1, \dots, 5\}$ the face $\Delta_T = \bigcap_{i \in T} F_i$ of Δ has dimension $4 - |T|$ and stabilizer W_T by (G2), and every face of a chamber is $w\Delta_T$ for exactly one T and one coset wW_T . The open faces map injectively to $\mathbb{H}^4/G$, since G acts freely and two points of a chamber in the same W -orbit are equal. The G -orbits of faces of type T correspond to the double cosets in $G \backslash W/W_T$, of which there are $n/|W_T|$. Hence, with Table 2,

$$\chi(\mathbb{H}^4/G) = \sum_T (-1)^{4-|T|} \frac{n}{|W_T|} = n \left(1 - \frac{5}{2} + \frac{21}{10} - \frac{13}{20} + \frac{721}{14400} \right) = \frac{n}{14400}. \quad \square$$

The number $1/14400$ is the rational Euler characteristic $\chi(W)$; see [CM05, Section 2], [Dav08, formula (16.8)], [Bro82, Chapter IX]. For one cell the count reads $1 - 60 + 240 - 300 + 120 = 1$: one 120-cell, 120/2 facets, 720/3 pentagons, 1200/4 edges and 600/5 vertices.

Proof of Corollary 1.3 from Theorem 1.2. (a) is Theorem 2.5. (b) By Proposition 2.6 the index of a torsion-free subgroup of finite index is $14400m$, and $m \neq 1$ by Theorem 1.2; the subgroup Σ of [CM05] has $m = 8$. (c) A subgroup $G \leq W$ acts freely on $\mathbb{H}^4$ if and only if it is torsion-free, and $\mathbb{H}^4/G \rightarrow \mathbb{H}^4/W$ is finite-sheeted if and only if $[W : G]$ is finite. The Euler characteristic of $\mathbb{H}^4/G$ is then $[W : G]/14400$ by Proposition 2.6. Conversely, each of (a), (b), (c) implies Theorem 1.2 by the same results. $\square$

Remark 2.7. A torsion-free subgroup G of index 14400 could not lie in W^+ , because a closed orientable hyperbolic 4-manifold has even Euler characteristic. Then $G \cap W^+$ would be a torsion-free subgroup of index 28800 inside W^+ . This is the sense in which Theorem 1.2 excludes one family of candidates for Problem 4.124.

3 The reduction

3.1 Finite subgroups and torsion

Lemma 3.1. *Every finite subgroup of W is conjugate to a subgroup of one of $P_1, \dots, P_5$.*

Proof. By a theorem of Tits, every finite subgroup of a Coxeter group is conjugate to a subgroup of a finite standard parabolic subgroup [Dav08, Theorem 12.3.4(i)], [Bou02, Ch. V, §4, Exercise 2 d)]. Here W is infinite and all W_T with $|T| = 4$ are finite, so the maximal finite standard parabolic subgroups are $P_1, \dots, P_5$. For this particular group there is also a direct argument. A finite group F of isometries of $\mathbb{H}^4$ fixes a point x , the circumcentre of an orbit. By (G1), $x = wy$ with $w \in W$ and $y \in \Delta$, and $w^{-1}Fw \leq W_{T(y)}$ by (G2). The five hyperplanes Π_i have no common point, so $T(y)$ omits some i , and $W_{T(y)} \leq P_i$. $\square$

Lemma 3.2. *A subgroup $G \leq W$ is torsion-free if and only if each of $P_1, \dots, P_5$ acts freely on the coset space $G \backslash W$ by right multiplication.*

Proof. The stabilizer of the coset Gw in P_i is $P_i \cap w^{-1}Gw$. If G is torsion-free, this finite group is trivial. Conversely, let all these stabilizers be trivial and let $g \in G$ have finite order. By Lemma 3.1, $w^{-1}gw \in P_i$ for some w and i , hence $w^{-1}gw \in P_i \cap w^{-1}Gw = 1$ and $g = 1$. $\square$

3.2 Gluings

Definition 3.3. A *gluing* is a map $\rho: H \rightarrow H$ such that

$$(R0) \quad \rho(\rho(x)) = x \text{ for all } x \in H;$$

$$(R1) \quad \rho(x\kappa) = \rho(x)\kappa \text{ for all } x \in H \text{ and } \kappa \in K;$$

$$(R2) \quad \theta_\rho^3 = \text{id}, \text{ where } \theta_\rho(x) = \rho(x)r_4.$$

Condition (R1) for $\kappa \in \{r_1, r_2, r_3\}$ implies it for all $\kappa \in K$. By (R1), ρ maps the facet xK onto the facet $\rho(x)K$ and is determined on xK by the image of one flag: a gluing pairs the facets, and for a pair of facets there are $|K| = 120$ possible gluing maps.

Proposition 3.4. (a) For every gluing ρ there is exactly one right action of W on the set H with $x \cdot r_i = xr_i$ for $i \leq 4$ and $x \cdot r_5 = \rho(x)$. Every right action of W on the set H that extends the right regular action of H arises in this way from exactly one gluing.

(b) Let $G \leq W$ be a subgroup of index 14400 such that H acts freely on $G \setminus W$. Then $x \mapsto Gx$ is a bijection of H onto $G \setminus W$, and the map ρ_G defined by $Gxr_5 = G\rho_G(x)$ is a gluing. The assignment $G \mapsto \rho_G$ is a bijection from the set of these subgroups onto the set of gluings; its inverse maps ρ to the stabilizer G_ρ of the point $1 \in H$ in the action of (a). The group G is torsion-free if and only if $P_1, \dots, P_4$ act freely on H in the action defined by ρ_G .

(c) For a gluing ρ and $a \in H$ let $\rho^a(x) = a\rho(a^{-1}x)$. Then ρ^a is a gluing and $G_{\rho^a} = aG_\rho a^{-1}$. Two subgroups as in (b) are conjugate in W if and only if their gluings lie in the same orbit $\{\rho^a : a \in H\}$.

A gluing is called *free* if $P_1, \dots, P_4$ act freely in the action of (a). A *solution* is a free gluing. The passage from ρ to ρ^a is called a *relabelling*. By Lemma 3.2 every torsion-free subgroup of index 14400 satisfies the hypothesis of (b). So:

Theorem 1.2 holds if and only if there is no solution. (1)

Proof. (a) W is defined by the relations $r_i^2 = 1$ and $(r_i r_j)^{m_{ij}} = 1$. The relations among $r_1, \dots, r_4$ hold for right multiplication in H . The relation $r_5^2 = 1$ is (R0). For $i \leq 3$ the word $r_i r_5$ acts as $x \mapsto \rho(xr_i)$, and by (R1) and (R0) its square is $x \mapsto \rho(\rho(xr_i)r_i) = \rho(\rho(x)) = x$; conversely $(r_i r_5)^2 = 1$ together with $r_5^2 = 1$ gives (R1) for $\kappa = r_i$. The word $r_5 r_4$ acts as θ_ρ , so $(r_5 r_4)^3 = 1$ is (R2), and $(r_4 r_5)^3$ is conjugate to $(r_5 r_4)^3$. This proves both statements.

(b) H acts freely on the set $G \setminus W$ of $14400 = |H|$ points, so the action is regular and $x \mapsto Gx$ is a bijection. Transporting the action of W on $G \setminus W$ to H gives an action as in (a), with r_5 acting as ρ_G ; so ρ_G is a gluing. The stabilizer of the point 1, which corresponds to the coset G , is G . Conversely let ρ be a gluing. The action of (a) is transitive, so $[W : G_\rho] = 14400$, and $G_\rho w \mapsto 1 \cdot w$ is an isomorphism of W -sets from $G_\rho \setminus W$ onto H that maps $G_\rho x$ to x for $x \in H$. Hence H acts freely on $G_\rho \setminus W$ and $\rho_{G_\rho} = \rho$. The last statement is Lemma 3.2, since $P_5 = H$ acts freely.

(c) The left multiplication $\lambda_a : x \mapsto ax$ commutes with the right action of H , and $\rho^a = \lambda_a \rho \lambda_a^{-1}$. So ρ^a satisfies (R0)–(R2), and λ_a is an isomorphism from the W -set defined by ρ onto the W -set defined by ρ^a . Therefore G_{ρ^a} is the stabilizer of the point $\lambda_a^{-1}(1) = a^{-1} = 1 \cdot a^{-1}$ in the action defined by ρ , which is $aG_\rho a^{-1}$. The conjugates of G_ρ in W are the stabilizers of the points of H , and every point of H is of the form $1 \cdot a^{-1}$ with $a \in H$. $\square$

For a gluing ρ and a flag x we call

$$\tau_\rho(x) = x^{-1}\rho(x) \in H$$

the *gluing element* at x ; thus $\rho(x) = x\tau_\rho(x)$.

Lemma 3.5. Let ρ be a gluing, $x, a \in H$ and $\kappa \in K$. Then

$$\tau_\rho(x\kappa) = \kappa^{-1}\tau_\rho(x)\kappa, \quad \tau_\rho(\rho(x)) = \tau_\rho(x)^{-1}, \quad \tau_{\rho^a}(ax) = \tau_\rho(x).$$

Proof. $\tau_\rho(x\kappa) = \kappa^{-1}x^{-1}\rho(x)\kappa$ by (R1); $\tau_\rho(\rho(x)) = \rho(x)^{-1}x$ by (R0); and $\tau_{\rho^a}(ax) = (ax)^{-1}a\rho(x)$. $\square$

Lemma 3.6 (necessary condition). Let ρ be a solution and $x \in H$. Then the element $r_5 \tau_\rho(x)^{-1}$ of W has infinite order. In particular, for $i \in \{1, 2, 3, 4\}$, no flag x satisfies $\rho(x) \in xH_i$, and no facet is glued to itself.

Proof. Let $z = \tau_\rho(x)$. In the action of Proposition 3.4(a) we have $x \cdot (r_5 z^{-1}) = \rho(x) z^{-1} = x$. So $r_5 z^{-1}$ lies in the stabilizer of x , which is a conjugate of G_ρ and hence torsion-free by Proposition 3.4(b). Moreover $r_5 z^{-1} \neq 1$, because $r_5 \notin H$. If $\rho(x) \in xH_i$, then $z \in H_i$ and $r_5 z^{-1}$ lies in the finite group P_i , a contradiction. The case $i = 4$, where $H_4 = K$, says that $\rho(x) \notin xK$. $\square$

A flag x with $\rho(x) \in xH_i$ is called a *loop*: for $i = 1, 2, 3, 4$ it means that x and $\rho(x)$ lie in the same cell and share their vertex, edge, pentagon or facet.

Remark 3.7. For this group the converse of the loop condition holds, although we do not use it: a gluing without loops is free. Indeed, P_i acts on the $[P_i : H_i] = 5, 4, 3, 2$ cosets of H_i as the full symmetric group, hence 2-transitively, so $P_i = H_i \cup H_i r_5 H_i$. A non-trivial element of H_i has no fixed point on H . If $h_1 r_5 h_2$ with $h_1, h_2 \in H_i$ fixes x , then $y = x h_1$ satisfies $\rho(y) = y h_1^{-1} h_2^{-1} \in y H_i$. This explains why the two programs, which enforce different consequences of freeness, prune in a very similar way (Section 7).

3.3 The exact finite-order test

By Lemma 3.6, an element $z \in H$ cannot be a gluing element of a solution if $r_5 z^{-1}$ has finite order in W . This is tested exactly as follows. Let $e_1, \dots, e_5$ be the standard basis of $\mathbb{Z}[\varphi]^5$ and let S_i be the matrix of the linear map

$$e_j \mapsto e_j + c_{ij} e_i, \quad c_{ii} = -2, \quad c_{12} = c_{21} = \varphi, \quad c_{23} = c_{32} = c_{34} = c_{43} = c_{45} = c_{54} = 1,$$

and $c_{ij} = 0$ otherwise. Since $c_{ij} = 2 \cos(\pi/m_{ij})$ for $i \neq j$, $r_i \mapsto S_i$ is the geometric representation of W , written over $\mathbb{Z}[\varphi]$. It is a faithful representation (Tits) [Bou02, Ch. V, §4, no. 4, Corollary 2], [Dav08, Theorem 6.12.10]. For $z = r_{w_1} r_{w_2} \cdots r_{w_l} \in H$ let

$$M_z = S_5 S_{w_l} \cdots S_{w_2} S_{w_1},$$

the image of $r_5 z^{-1}$. The matrices are computed with exact integer arithmetic: an element $a + b\varphi$ is a pair of integers of unbounded size, and $\varphi^2 = \varphi + 1$. An element z is *flagged* if M_z^e is the identity matrix for some $1 \leq e \leq 60$.

Lemma 3.8. *If $z \in H$ is flagged, then z is not a gluing element of any solution.*

Proof. If $M_z^e = 1$, then $(r_5 z^{-1})^e = 1$ in W because the representation is faithful. Now apply Lemma 3.6. $\square$

Only this implication is used by the programs. The converse also holds: by Lemma 3.1 an element of finite order of W is conjugate into some P_i , and the orders of the elements of $P_1, \dots, P_5$ divide 60 (the element orders are 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30 in H ; they divide 30 in P_4 and P_3 , 12 in P_2 and 60 in P_1). So z is flagged if and only if $r_5 z^{-1}$ has finite order. Since r_5 commutes with K , and since $z r_5 = (r_5 z^{-1})^{-1}$ is conjugate to $r_5 z$, the set of flagged elements is invariant under conjugation by K and under inversion. The elements of $H_1 \cup \dots \cup H_4$ are flagged, as the proof of Lemma 3.6 shows.

3.4 Types and classes

The group K acts on H by conjugation, $z \mapsto \kappa^{-1} z \kappa$. The orbits are called *K-classes*; $[z]$ denotes the class of z and $[z]^{-1} = [z^{-1}]$ the *inverse class*. By Lemma 3.5 the class $[\tau_\rho(x)]$ depends only on the facet $F = xK$. It is called the *type* of F with respect to ρ . The partner facet $\rho(x)K$ has the inverse type, and a relabelling does not change the types: the facet axK has the same type with respect to ρ^a as xK with respect to ρ .

A K -class is called a *loop class* if it contains an element of $H_1 \cup H_2 \cup H_3 \cup H_4$, an *elliptic class* if it is not a loop class and its elements are flagged, and *admissible* otherwise. A *type class* is the union $[z] \cup [z]^{-1}$ of a K -class and its inverse class.

Proposition 3.9 (computed). *H has 188 K -classes: 17 loop classes with 480 elements in total (the 10 classes contained in K are among them), 17 elliptic classes with 1200 elements, and 154 admissible classes with 12720 elements. The 34 classes that are not admissible are equal to their inverse classes. Of the 154 admissible classes, 126 are equal to their inverse classes and 28 form 14 pairs of mutually inverse classes. So there are 140 admissible type classes. The flagged elements are exactly the 1680 elements of the 34 classes that are not admissible.*

This is the result of three computations with exact arithmetic that agree element by element: the table generator of program A, the table generator of program B, and the script `verification_engine/own/h4_classes.py` of an independent verification run, which builds H from matrices over $\mathbb{Z}[\varphi]$ and also confirms the number 188 by the Burnside count $\frac{1}{|K|} \sum_{\kappa \in K} |C_H(\kappa)|$ (Section 7). A fourth computation, written from scratch for the last verification run (`independent_run_2/r2_group.py`, Section 7.8), gave the same numbers, the same classes as the table of program A and the same flagged elements as the file of program B.

Corollary 3.10. *All 120 facet types of a solution are admissible.*

Proof. Let $z = \tau_\rho(x)$. If $[z]$ is a loop class, then $\kappa^{-1}z\kappa \in H_i$ for some $\kappa \in K$ and $i \leq 4$, so $\rho(x\kappa) = x\kappa\tau_\rho(x\kappa) \in x\kappa H_i$ by Lemma 3.5, contradicting Lemma 3.6. If $[z]$ is elliptic, z is flagged, contradicting Lemma 3.8. $\square$

3.5 Slices

The search space is cut into *slices* by normalizing the gluing of one facet. The two programs do this in two ways.

Slices of program A. Let $c_0, c_1, \dots, c_{153}$ be the admissible K -classes in a fixed order, and let $z_t \in c_t$ be a fixed representative. (Program A uses the order of its table file, which was chosen by a calibration run before the enumeration, and takes for z_t the smallest element of c_t in the numbering of the table. Any order and any representatives would do.) For $0 \leq t \leq 153$ let

$$\mathcal{S}_t = \{\rho : \rho \text{ is a solution, } \rho(1) = z_t, \text{ and every facet type of } \rho \text{ is one of } c_t, c_{t+1}, \dots, c_{153}\}.$$

Proposition 3.11. *Let ρ be a solution and let t be the smallest index such that c_t is the type of a facet of ρ . Then $\rho^a \in \mathcal{S}_t$ for some $a \in H$. Hence there is no solution if and only if $\mathcal{S}_0 = \dots = \mathcal{S}_{153} = \emptyset$.*

Proof. By Corollary 3.10 all types of ρ are among the c_j , so t exists. Let xK be a facet of type c_t and choose $\kappa \in K$ with $\kappa^{-1}\tau_\rho(x)\kappa = z_t$. Put $y = x\kappa$ and $a = y^{-1}$. Then $\rho^a(1) = y^{-1}\rho(y) = \tau_\rho(y) = z_t$ by Lemma 3.5. The solution ρ^a has the same types as ρ , so all of them have index $\geq t$. $\square$

If $c_t^{-1} = c_{t'}$ with $t' < t$, then $\mathcal{S}_t$ is empty: the facet K of the flag 1 has type c_t , so its partner facet has type $c_{t'}$. By Proposition 3.9 this happens for 14 values of t .

Slices of program B. Let $d_0, \dots, d_{139}$ be the admissible type classes in a fixed order, and $t_i \in d_i$ a fixed representative. (Program B numbers the type classes by their smallest elements in the numbering of its own table and takes the smallest element for t_i .) The type class of a facet is the type class containing its type; the two facets of a pair have the same type class. Let

$$\mathcal{S}'_i = \{\rho : \rho \text{ is a solution, } \rho(1) = t_i, \text{ and the type class of every facet is one of } d_i, \dots, d_{139}\}.$$

Proposition 3.12. *Let ρ be a solution and let i be the smallest index such that d_i is the type class of a facet of ρ . Then $\rho^a \in \mathcal{S}'_i$ for some $a \in H$. Hence there is no solution if and only if $\mathcal{S}'_0 = \dots = \mathcal{S}'_{139} = \emptyset$.*

Proof. Let xK be a facet with type class d_i . By Lemma 3.5 the gluing elements $\tau_\rho(y)$ of the 240 flags y of the two facets xK and $\rho(x)K$ are the elements $\kappa^{-1}\tau_\rho(x)^{\pm 1}\kappa$, $\kappa \in K$, that is, all elements of d_i . Choose y with $\tau_\rho(y) = t_i$ and put $a = y^{-1}$. Then $\rho^a(1) = t_i$, and the type classes are unchanged. $\square$

Remark 3.13. The numbers 154 and 140 are consistent. Program A has one slice for every admissible K -class; the 14 slices whose inverse class has a smaller index are empty, which leaves $126 + 14 = 140$. Program B has one slice for every admissible type class. The two decompositions use different orders, so the slices themselves are different sets and only the totals can be compared.

3.6 Partial gluings and the search scheme

Both programs build a gluing facet by facet. We fix the common terminology and prove once the argument that both completeness proofs use.

Definition 3.14. A state s consists of a set of facets, called the *glued* facets of s , with union $U_s \subseteq H$, and a map $\rho_s: U_s \rightarrow U_s$ with $\rho_s(\rho_s(x)) = x$, $\rho_s(x\kappa) = \rho_s(x)\kappa$ and $\rho_s(x) \notin xK$. The other facets are *unglued*. A state is *complete* if all facets are glued. A state s' *extends* s if $U_s \subseteq U_{s'}$ and $\rho_{s'}|_{U_s} = \rho_s$; a gluing ρ *extends* s if $\rho|_{U_s} = \rho_s$. If x and y lie in two different unglued facets of s , then $s + (x \mapsto y)$ denotes the state in which these two facets are glued in addition, by $x\kappa \mapsto y\kappa$ and $y\kappa \mapsto x\kappa$.

Lemma 3.15. *Let the solution ρ extend the state s , let F be an unglued facet of s and $x \in F$. Then $\rho(x)$ lies in an unglued facet of s different from F , and ρ extends $s + (x \mapsto \rho(x))$.*

Proof. If the facet F' of $\rho(x)$ were glued in s , then $\rho_s(F') = \rho(F') = F$ would be glued as well. $F' \neq F$ by Lemma 3.6. The rest is clear. $\square$

For every facet F we fix a flag $x_F \in F$, the *probe flag*. Consider a finite rooted tree in which every node N carries a state s_N , the state of a child extends the state of its parent, and every node carries lists $L_N(F) \subseteq H$ for some of the facets F that are unglued in s_N . Let $\mathcal{S}$ be a set of solutions.

Lemma 3.16 (search scheme). *Let $\rho \in \mathcal{S}$. Assume that the following holds for all nodes N whose state is extended by ρ .*

- (T) (Trials.) *At N a procedure $\text{trial}(s_N, F, y)$ is available, which may depend on N and which either fails or returns a state extending $s_N + (x_F \mapsto y)$. For every unglued facet F of s_N the trial of $y = \rho(x_F)$ does not fail and returns a state extended by ρ .*
- (L) (Lists.) *Every list $L_N(F)$ is either created at N or derived from a list $L_{N'}(F)$ of the parent N' of N . It contains every flag y with the following properties: y lies in an unglued facet of s_N different from F ; if the list is derived, $y \in L_{N'}(F)$; the trial of y in s_N does not fail; and y passes the additional tests of the procedure, if there are any. The additional tests never reject $y = \rho(x_F)$.*
- (B) (Branching.) *If s_N is not complete and no list of N is empty, then N carries at least one list, and for one of the facets F with a list the following holds: if $\rho(x_F) \in L_N(F)$ and the trial of $\rho(x_F)$ does not fail, then N has a child with state $\text{trial}(s_N, F, \rho(x_F))$.*

If the state of the root is extended by ρ , then the tree has a node whose state is complete and equal to ρ .

Proof. Let N be a node whose state is extended by ρ . We show first that every list $L_N(F)$ contains $y = \rho(x_F)$, by induction on the depth of N . By Lemma 3.15, y lies in an unglued facet different from F . If $L_N(F)$ is derived from a list of the parent N' , then $s_{N'}$ is extended by ρ too, and $y \in L_{N'}(F)$ by induction. The trial of y does not fail by (T), and y passes the additional tests. So $y \in L_N(F)$ by (L). Consequently no list of N is empty. If s_N is not complete, (B) and (T) give a child of N whose state is $\text{trial}(s_N, F, \rho(x_F))$ for some F and is extended by ρ . It has more glued facets than s_N . Starting from the root we reach, after at most 60 steps, a node whose state is complete and extended by ρ , hence equal to ρ . $\square$

In the applications a list is either created at the node from all flags of all unglued facets, or taken over from the parent with or without a new filtering. A list that is taken over unchanged satisfies (L) trivially. A program that follows every element of one list satisfies (B) for every ρ . The lemma shows that lists never have to be enlarged again, although the candidates were tested in an earlier state: what was refuted in a smaller state stays refuted. It also shows what has to be checked for each program: the root, the trial (T), the way lists are filtered (L), and the exhaustive branching (B). A search that prunes only in this way and reaches no complete state proves that $\mathcal{S}$ is empty, whether or not its final test of complete states is correct.

4 Program A

Program A consists of the table generator `programA/gen_tables.py` and the search engine `programA/fsearch_production.c`, from which the production binary `fsearch` was built. The same folder contains `gsearch.c`, a plain version of the search without candidate lists, which was used for reruns of complete slices (Section 7), and `aux_counter_csearch.c`, an auxiliary counter of the completions of a given partial gluing, written from the statement of Definition 3.3 alone and used only for cross-checks. All production runs used the options

-k 1 -root t -ell 1 -keys 8 -sub S I ,

which mean: one cell; the slice $\mathcal{S}_t$; the elliptic classes are excluded; up to 8 candidate lists are kept; the sub-slice I of S . In this section line numbers refer to `fsearch_production.c`.

4.1 Tables and data structures

The table file contains the right regular representation of H as four arrays $h \mapsto hr_i$ on the points $0, \dots, 14399$, with 0 the identity. The generator obtains it from the root system of type H_4 in floating point and then verifies it exactly: the four maps are fixed-point-free involutions, they satisfy every Coxeter relation at every point, they generate a transitive group, and the number of points is $14400 = |W(H_4)|$; so the arrays are the regular representation. All other tables are derived from these arrays with integer arithmetic: an enumeration $\kappa_0 = 1, \kappa_1, \dots, \kappa_{119}$ of K with its multiplication and inversion; the facets $0, \dots, 119$ (the left cosets xK) with a representative $\text{rep}(F)$ for each facet F , where facet 0 is K and $\text{rep}(0) = 1$; ridge representatives $\delta_1, \dots, \delta_{12} \in K$, one for every coset κD ; the K -classes with their flags (loop class, elliptic class) and inverse classes, ordered as described in Section 3.5; and, for every facet F and every $h \in H$, the class of $\text{rep}(F)^{-1}h$. The flagged classes are computed as in Section 3.3.

A flag is stored as a pair (F, p) , meaning $\text{rep}(F)\kappa_p$. The 12 *ridges* of a facet F correspond to the flags $\text{rep}(F)\delta_j$; the facet *adjacent* to F across the j -th ridge is the facet of $\text{rep}(F)\delta_j r_4$. The probe flag of F is $x_F = \text{rep}(F)$. A state s (Definition 3.14) is stored as an array that contains, for every glued facet F , the flag $\rho_s(x_F)$; then $\rho_s(x_F \kappa_p) = \rho_s(x_F)\kappa_p$. Every change of the state is recorded on a trail and undone when the search backtracks (lines 86–88).

A K -class c is called *allowed* in the slice t if c and c^{-1} are admissible and both have index at least t in the order $c_0, \dots, c_{153}$. The program tabulates this property (lines 415–423). If $\rho \in \mathcal{S}_t$, then every facet type of ρ is allowed, because the inverse class of a type is the type of the partner facet.

4.2 The rules

We list everything the engine does to a state, as rules (A1)–(A13). The rules (A2)–(A5) and (A7)–(A12) can discard something; (A1), (A6) and (A13) are stated for completeness.

- (A1) *Representation* (lines 54–58, 69–70, 81). A gluing of two facets is entered for both facets at once: if $\rho_s(x) = y$ with $x = (F, p)$ and $y = (F', p')$, then $\rho_s(x_F) = y\kappa_p^{-1}$ and $\rho_s(x_{F'}) = x\kappa_{p'}^{-1}$.

The procedure $\text{glue}(x, y)$ (lines 61–85) tries to pass from s to $s + (x \mapsto y)$. Let F, F' be the facets of x, y .

- (A2) If F or F' is glued in s : if both are glued and $\rho_s(x) = y$, nothing is done and the result is “known”; otherwise the result is “contradiction” (lines 63–66).
(A3) If $F = F'$, the result is “contradiction” (line 67).
(A4) Let c be the class of $x_F^{-1} y\kappa_p^{-1}$, the type that F would get, and c' the class of $x_{F'}^{-1} x\kappa_{p'}^{-1}$, the type that F' would get. If c or c' is not allowed, the result is “contradiction” (lines 71–76). This excludes (a) loop classes, (b) elliptic classes and (c) classes below the threshold t , for both facets.

Otherwise the state becomes $s + (x \mapsto y)$ and the result is “new”.

- (A5) *Propagation* (lines 127–142). The newly glued facets are put into a queue. For a facet F taken from the queue and every $j \leq 12$ such that the facet adjacent to F across its j -th ridge is glued, let

$$x_0 = x_F \delta_j, \quad x_4 = \rho_s(x_0) r_4, \quad x_3 = \rho_s(x_0 r_4) r_4,$$

and call $\text{glue}(x_3, x_4)$. If the result is “contradiction”, the propagation fails; if it is “new”, the two facets are added to the queue.

- (A6) *Candidates* (lines 223–233, 257). The candidates for the image of x_F , for an unglued facet F , are all flags of all unglued facets other than F .
(A7) *Type prefilter* (line 236). A candidate y for F is dropped if the class of $x_F^{-1} y$ is not allowed.
(A8) *Quick test* (lines 239–255 and 268–288). A candidate y for F , lying in the facet F' , is dropped if for some j such that the facet adjacent to F across its j -th ridge is glued, the flags

$$x_3 = \rho_s(x_F \delta_j r_4) r_4, \quad x_4 = y \delta_j r_4,$$

with facets F_3 and F_4 , satisfy one of the following conditions:

- (Q1) F_4 is glued, and F_3 is unglued or $\rho_s(x_4) \neq x_3$;
(Q2) F_4 is unglued and F_3 is glued;
(Q3) $F_3 = F_4$;
(Q4) $F_3 = F$ and $F_4 \neq F'$;
(Q5) $F_3 = F'$ and $F_4 \neq F$;

- (Q6) $F_3 \notin \{F, F'\}$ and $F_4 \in \{F, F'\}$;
- (Q7) F_3 and F_4 are unglued, different from each other and from F and F' , and the class of $x_{F_4}^{-1} x_3 \kappa_{p_4}^{-1}$ is not allowed, where $x_4 = (F_4, p_4)$.
- (A9) *Trial* (lines 201–207 and 369–372). The trial of a candidate y for F is $\text{glue}(x_F, y)$ followed, if the result is “new”, by the propagation (A5) started with the two facets. It fails if a contradiction occurs. When a trial is made only to filter a list, the state is restored afterwards.
- (A10) *Lists* (lines 209–264, 289–308, 334). A list for F is *created* at a node as the set of all candidates (A6) that pass (A7) and (A8) and whose trial (A9) does not fail. A list is passed on to the children of the node as long as F is unglued. A child uses an inherited list unchanged or *filters* it: a candidate is dropped if its facet has been glued in the meantime (line 302), if the quick test (A8) rejects it (line 303), or if its trial fails (line 304).
- (A11) *Empty lists* (lines 336, 352, 360, 362). A node is abandoned as soon as one of its lists is found to be empty.
- (A12) *Branching* (lines 365–374). A node that is not abandoned chooses one of its lists, say for the facet F , and makes, for every y in the list, the trial of y for F ; if it does not fail, the resulting state is the state of a child node. At the node of depth 1 of a job with the option `-sub  $S$   $I$` , only the candidates whose position in the list is congruent to I modulo S are followed (line 368).
- (A13) *Leaf test* (lines 327–332 and 145–176). When all facets are glued, the complete map ρ_s is tested from scratch: (R0), (R1), (R2), the absence of fixed points, transitivity, and that every orbit of P_i has $|P_i|$ points for $i = 1, \dots, 4$. It is reported as a solution if it passes.

The search starts with $\text{glue}(1, z_t)$ and its propagation (lines 445–453). If this fails, the job ends with 0 nodes. Otherwise the resulting state is the state of the node of depth 1, and the recursion begins (lines 310–376). A node whose state is complete applies (A13). Any other node keeps the inherited lists of the facets that are still unglued. If there is none, it selects up to 8 unglued facets (preferring facets with many glued neighbours) and creates their lists. It then filters the shortest list if that list is inherited, and all inherited lists if the shortest list has more than 8 elements, and branches on the shortest list. Which facets are selected, which lists are filtered and which list is branched on affects the running time only.

With the options of the production runs the following parts of the source are inactive: the orientation test (lines 77–80, 221, 228, 237, 421), the option `-nosame` (lines 72, 227, 251, 282), the special treatment of cells without glued facets (lines 224–226, 343 and 356, which matter only for several cells) and the recomputation of lists when a further cell is entered (line 336), the node and depth limits (lines 311, 313–326, 373), the option `-resel` (line 335), and the generic propagation for other Coxeter labels (lines 91–121). Every log records the command of its job (line `START`) and the option values (line `PARAMS`). The options `-maxnodes` and `-resel` are not printed in the line `PARAMS`; no command contains them.

4.3 Proofs of necessity

Lemma 4.1 (ridge relation). *Let ρ be a gluing and $x_0 \in H$. Then $\rho(\rho(x_0)r_4) = \rho(x_0r_4)r_4$.*

Proof. With $\theta = \theta_\rho$ we have $\theta^{-1}(x) = \rho(xr_4)$ by (R0). By (R2), $\theta(\theta(x_0)) = \theta^{-1}(x_0)$, that is, $\rho(\rho(x_0)r_4)r_4 = \rho(x_0r_4)$. $\square$

Geometrically: the three cells around a pentagon are joined by three gluings, and two of them determine the third.

Proposition 4.2. *Let $\rho \in \mathcal{S}_t$ extend the state s .*

- (a) *If $\rho(x) = y$, then $\text{glue}(x, y)$ does not result in a contradiction, and the state after it is extended by ρ .*
- (b) *The propagation (A5), started in s with any queue of glued facets, does not fail, and the state after it is extended by ρ .*
- (c) *If F is an unglued facet of s , the trial (A9) of $y = \rho(x_F)$ does not fail and results in a state extended by ρ . The candidate $y = \rho(x_F)$ is not dropped by (A7) or by (A8).*

Proof. (a) Let F, F' be the facets of x, y , so $\rho(F) = F'$. If F is glued in s , then $\rho_s(x) = \rho(x) = y$, hence F' is glued too, and (A2) gives “known”. The same holds if F' is glued, since $\rho(y) = x$. So (A2) never gives a contradiction. If both are unglued, then $F \neq F'$ by Lemma 3.6, so (A3) does not apply. The classes c and c' in (A4) are the types of F and F' with respect to ρ , because $\rho(x_F) = \rho(x\kappa_p^{-1}) = y\kappa_p^{-1}$ and likewise for F' . They are allowed, since $\rho \in \mathcal{S}_t$. So the result is “new”, and the new state $s + (x \mapsto y)$ is extended by ρ (Lemma 3.15).

(b) Whenever (A5) calls $\text{glue}(x_3, x_4)$, the current state s' is extended by ρ (by induction over the calls), and the facets of x_0 and x_0r_4 are glued in s' . So $\rho_{s'}(x_0) = \rho(x_0)$, $\rho_{s'}(x_0r_4) = \rho(x_0r_4)$, and $\rho(x_4) = x_3$ by Lemma 4.1. By (a) the call does not result in a contradiction and leaves a state extended by ρ .

(c) The first statement follows from (a), (b) and Lemma 3.15. The class tested in (A7) is the type of F with respect to ρ , which is allowed. For (A8) let $F' = \rho(F)$ and let j be as in (A8). Then $x_3 = \rho(x_F\delta_jr_4)r_4$ and $x_4 = \rho(x_F)\delta_jr_4 = \rho(x_F\delta_j)r_4$, so $\rho(x_4) = x_3$ by Lemma 4.1 and $\rho(F_4) = F_3$. We show that none of (Q1)–(Q7) holds. (Q1): if F_4 is glued in s , then $\rho_s(x_4) = \rho(x_4) = x_3$, so F_3 is glued and $\rho_s(x_4) = x_3$. (Q2): if F_3 is glued in s , so is $F_4 = \rho(F_3)$. (Q3): $F_3 \neq F_4$ by Lemma 3.6. (Q4), (Q5), (Q6): ρ exchanges F and F' and exchanges F_3 and F_4 ; so $F_3 = F$ if and only if $F_4 = F'$, and $F_3 = F'$ if and only if $F_4 = F$. (Q7): the class in question is the type of F_4 with respect to ρ , because $\rho(x_{F_4}) = \rho(x_4\kappa_{p_4}^{-1}) = x_3\kappa_{p_4}^{-1}$; it is allowed. $\square$

Theorem 4.3 (completeness of program A). *Let $0 \leq t \leq 153$ and $S \geq 1$, and assume that the table of Section 4.1 is correct. If $\mathcal{S}_t \neq \emptyset$, then one of the S jobs with the options `-k 1 -root t -ell 1 -keys 8 -sub S I`, $0 \leq I < S$, reaches a complete state, provided the S jobs run to their normal end. More precisely, every $\rho \in \mathcal{S}_t$ is the state of a leaf of exactly one of these jobs.*

Proof. Let $\rho \in \mathcal{S}_t$. Since $\rho(1) = z_t$ and $x_K = 1$, the state after $\text{glue}(1, z_t)$ and its propagation is extended by ρ , and this step does not fail, by Proposition 4.2. This state and the lists of the node of depth 1 are the same in all S jobs, because the program is deterministic and I is used only in (A12). Let F be the facet on which the node of depth 1 branches, let k be the position of $\rho(x_F)$ in its list (it is in the list by the argument of Lemma 3.16), and let $I \equiv k \pmod{S}$. We apply Lemma 3.16 to the search tree of the job I , with $\mathcal{S} = \mathcal{S}_t$. Condition (T) is Proposition 4.2(c). Condition (L) holds by (A6), (A7), (A8), (A10): a created list consists of all flags of all unglued facets other than F that pass the two additional tests (A7), (A8) and whose trial does not fail; a filtered list consists of the elements of the parent’s list whose facet is still unglued, which pass (A8) and whose trial does not fail; and by Proposition 4.2(c) the additional tests never reject $\rho(x_F)$. Condition (B) holds by (A11) and (A12): a node that is not abandoned has at least one list (it has an unglued facet, so it inherits or creates one), and it follows every element of one list, at depth 1 every element whose position is congruent to I modulo S , among them $\rho(x_F)$. A node is abandoned only because of an empty list or because its state is complete. Hence the tree of the job I contains a node with the complete state ρ . It is the only such node in the S trees, because different children of a node glue the same facet in different ways. $\square$

A complete state is counted in the field `leaves` of the log, and a job that was cut by a node or depth limit, or killed, does not print the final line `DONE` (lines 454–462). So the theorem gives the following criterion, which does not involve the leaf test (A13).

Corollary 4.4. *If all jobs of the slice t end with a line `DONE` that shows `leaves=0`, then $\mathcal{S}_t = \emptyset$.*

4.4 The empty slices

For 14 classes t , namely $t = 9, 21, 35, 53, 54, 56, 58, 65, 72, 76, 78, 82, 140$ and 141 , the inverse class of c_t has a smaller index than t . Then the class c_t itself is not allowed in the slice t , the initial glue(1, z_t) is refused by (A4), and the job reports `nodes=0`. This is correct: $\mathcal{S}_t = \emptyset$ for these t (Section 3.5), and a solution in which the type c_t occurs belongs, after relabelling, to the slice of its smallest type, where both c_t and c_t^{-1} are allowed. In 13 further slices ($t = 3, 7, 10, 11, 15, 23, 25, 26, 31, 32, 34, 152, 153$) the root gluing is accepted but one of the first lists is empty, and the job reports `nodes=1`.

5 Program B

Program B consists of the table generator `programB/gen_tables.py` and two versions of one search engine: `programB/fps_final.c` and `programB/fps_v7.c`. The second is the first plus two options, `-bw` and `-sym`; the file `programB/diff_fps_final_v7.txt` shows the difference. Program B treats a general string Coxeter group; we describe what it does for W and one cell (its “mode B”). The production runs used

```
-m B -t i -I tables/t5333.inadm -fd 2 -F 16 -F2 12 -dr 3 -br 1
and optionally -s j/16 and -bw 200 -sym,
```

which mean: one cell with the slice $\mathcal{S}'_i$; the file of flagged elements; the numbers of candidate lists that are created or filtered at depth 1, at depth 2 and below (16, 12 and 3); a rule for the choice of the branching list at depth 1; the sub-slice j of 16; and, for `fps_v7` only, a wider window for that choice and the symmetry reduction of Section 5.4. In this section line numbers refer to `fps_final.c` unless “v7” is written. We keep the notation of Section 3; the source calls the gluing σ , the group K is called H_f there, and H is called C .

5.1 Tables and the vertex model

The generator obtains the right regular representations of H and of the *vertex group* $V = P_1 = \langle r_2, \dots, r_5 \rangle \cong S_5$ by Todd–Coxeter coset enumeration over the trivial subgroup, and the orders 120, 48, 60, 240, 14400 of $P_1, \dots, P_5$ in the same way. With the option `--inadm` it writes a file with one entry for every $z \in H$, equal to 1 if M_z^{60} is the identity matrix (Section 3.3; the power is computed by repeated squaring with integers of unbounded size). By Section 3.3 these are the flagged elements. The engine derives all other tables when it starts (lines 156–267): the facets with probe flags x_F , the multiplication of K , left multiplications and inverses in H , the type classes, and the data of the vertex model described now. It refuses a file of flagged elements that is not a union of type classes (line 275). A type class is *allowed* in the slice i if it is one of $d_i, \dots, d_{139}$ (lines 846–852, 864, 883).

Let $C_v = H_1 = \langle r_2, r_3, r_4 \rangle$, a group of order 24 contained in H and in V . A *corner* is a coset xC_v in H : the 24 flags at one of the 600 vertices of the cell. Every corner u has a fixed representative flag $\text{rep}(u)$. In the W -set defined by a gluing ρ , the orbit of a flag under V consists of the flags around one vertex of the glued space; if ρ is free, it has 120 elements and meets five corners.

For the current state s the engine keeps a partition of the set of corners into *components*, a *root* corner in every component, and an element $E[u] \in V$ for every corner u . They satisfy:

- (INV) for every gluing ρ that extends s and every corner u with root a : $\text{rep}(u) = \text{rep}(a) \cdot E[u]$, where the dot is the action of W on H defined by ρ ;
- (DIS) for two different corners u, u' of one component: $E[u]C_v \neq E[u']C_v$.

At the beginning every corner is a component of its own, with $E[u] = 1$. Every component has a table that records which of the five cosets of C_v in V are occupied by its corners.

5.2 The rules

The procedure `define_pair`(X, Y) (lines 460–511) tries to pass from s to $s + (X \mapsto Y)$. Let F, F' be the facets of X, Y .

- (B1) If $F = F'$: conflict (line 464).
- (B2) If F or F' is glued in s : conflict (line 465).
- (B3) *Type filter* (lines 467–472). If the type class of $x_F^{-1}Y\kappa^{-1}$, where $X = x_F\kappa$, is not allowed: conflict.

Otherwise the pair is entered. Then, for each of the 20 vertices of the facet F , one flag $x = x_F\kappa'$ at that vertex is taken, with its image $y = \rho_s(x)$. Let t, u be the corners of x, y , and write $x = \text{rep}(t)a$, $y = \text{rep}(u)b$ with $a, b \in C_v$. The procedure `link_corners`(t, u, g) with $g = ar_5b^{-1} \in V$ (lines 402–446) is called.

- (B4) If t and u lie in the same component and $E[t]g \neq E[u]$: conflict (line 407).
- (B5) If t and u lie in different components with roots a' and b' , let $G = E[t]gE[u]^{-1}$. The component of b' is merged into the component of a' with the new values $GE[m]$ for its corners m (the roles are exchanged, with G^{-1} , if the component of a' is the smaller one). If a new value lies in a coset of C_v that is already occupied: conflict (lines 408–412).
- (B6) *Forced values* (lines 417–443). After a merge, for every corner m that received a new value and each of the four flags $X' = \text{rep}(m)a_k$, where $a_1, \dots, a_4$ represent the cosets $C_v/\langle r_2, r_3 \rangle$, let $c = E[m]a_kr_5$. If the coset cC_v is occupied by a corner w , let $d = E[w]^{-1}c \in C_v$ and $Y' = \text{rep}(w)d$. If the facet of X' is glued and $\rho_s(X') \neq Y'$: conflict. If the facet of X' is unglued: conflict if the facet of Y' is glued, or equal to the facet of X' , or if the type class of the gluing $X' \mapsto Y'$ is not allowed; otherwise the pair (X', Y') is put into a queue.
- (B7) *Propagation* (lines 513–523). For a pair (X', Y') taken from the queue: if the facet of X' is glued, conflict unless $\rho_s(X') = Y'$; otherwise, conflict if the facet of Y' is glued; otherwise `define_pair`(X', Y') is called.
- (B8) *Options* (lines 662–671). The options for the image of x_F , for an unglued facet F , are all flags of all unglued facets other than F .
- (B9) *Lists* (lines 684–696, 737, 747–754, 759–769, 789–791). The *trial* of an option Y for F is `define_pair`(x_F, Y) followed by the propagation (B7) (lines 537–543). A list is created from all options (B8) by deleting those whose trial ends in a conflict. A list is passed on to the children while its facet is unglued; a child uses it unchanged, or filters it by deleting the options whose trial ends in a conflict.

- (B10) *Empty lists* (lines 752, 768). A node is abandoned if a list that it has created or filtered is empty.

- (B11) *Branching* (lines 794–803). Otherwise the node chooses one of the lists it has created or filtered and follows every option of it whose trial has no conflict. At depth 1 of a job with the option `-s j/J`, only the options whose position in the list is congruent to j modulo J are followed (line 796).
- (B12) *Min-type rule* (line 883). The type classes $d_0, \dots, d_{i-1}$ are not allowed in the slice i . (This is part of the definition of “allowed” used in (B3) and (B6); we list it separately because it is the symmetry reduction of Proposition 3.12.)
- (B13) *Symmetry of the root gluing*, in `fps_v7` with `-sym` only: see Section 5.4.
- (B14) *Leaf test* (lines 546–575). When all facets are glued, the complete map is tested from scratch: it is an involution without fixed points, it satisfies the relations with $r_1, \dots, r_4$, the action is transitive, and every orbit of P_i has $|P_i|$ elements for $i = 1, \dots, 5$. The log counts the complete maps that pass (`solutions`) and those that fail (`complete_rejected`).

The search starts with the trial of the option t_i for the facet K , whose probe flag is 1 (line 888). The node of depth 1 creates 16 lists. A node of depth 2 filters up to 12 inherited lists and creates new ones up to a total of 12; a deeper node filters up to 3 inherited lists, and creates a list only if it has none. The node then branches on the shortest of the lists it has created or filtered; at depth 1 the choice is among the lists that are at most 1.25 times (`fps_final`) or 2 times (`fps_v7` with `-bw 200`) as long as the shortest one. A node always ends up with at least one such list: if it has an inherited list it filters at least one, and otherwise it creates one. As for program A, these choices affect the running time only.

The following parts of the source are inactive in the production runs: the orientation condition (lines 437, 466, 571, 624, 637, 667, 676, 850), the code for two cells (lines 475–491, 669, 675–681 and the branches marked `special`) and for the generic mode, the optional filter by products of gluing elements (option `-f`; lines 279–387 and 541), the plain search without lists (option `-old`; lines 604–647), the node limit (option `-L`; lines 606, 714), and the restriction options `-R`, `-C` (lines 849, 853–858). The first line of every log (`START`) records `orient=0`, `filter=0`, `minrule=0`, `old=0`, `restrict=-`, the file of flagged elements, and `admissible=140`; a restriction by `-R` or `-C` would lower the last number.

5.3 Proofs of necessity

Lemma 5.1. *The properties (INV) and (DIS) hold for every state that the engine reaches without a conflict.*

Proof. They hold at the beginning. The data change only in (B5). Let ρ be any gluing that extends the state in which `link_corners(t, u, g)` is called. Then $\rho(\text{rep}(t)a) = \text{rep}(u)b$, that is, $\text{rep}(t) \cdot (ar_5) = \text{rep}(u) \cdot b$, hence

$$\text{rep}(u) = \text{rep}(t) \cdot g, \quad g = ar_5b^{-1}. \quad (2)$$

By (INV) before the merge, $\text{rep}(t) = \text{rep}(a') \cdot E[t]$, $\text{rep}(u) = \text{rep}(b') \cdot E[u]$ and $\text{rep}(m) = \text{rep}(b') \cdot E[m]$ for the corners m of the second component. So $\text{rep}(b') = \text{rep}(a') \cdot E[t]gE[u]^{-1} = \text{rep}(a') \cdot G$ and $\text{rep}(m) = \text{rep}(a') \cdot GE[m]$: (INV) holds after the merge. (DIS) holds after the merge because (B5) reports a conflict otherwise. When the search backtracks, the previous data are restored (lines 448–457). $\square$

Proposition 5.2. *Let $\rho \in \mathcal{S}'_i$ extend the state s , and let $\rho(X) = Y$, where the facet of X is unglued in s . Then `define_pair(X, Y)` followed by the propagation (B7) ends without a conflict, in a state that is extended by ρ .*

Proof. We show that none of (B1)–(B7) reports a conflict as long as the current state is extended by ρ and the pair that is being entered is a restriction of ρ ; by induction over the calls of `define_pair` this proves the proposition.

(B1) and (B2): Y lies in an unglued facet different from the facet of X by Lemma 3.15. (B3): the element $x_F^{-1}Y\kappa^{-1}$ equals $\tau_\rho(x_F)$, and its type class is allowed because $\rho \in \mathcal{S}'_i$. After the pair is entered, the state is still extended by ρ , so (INV) applies to ρ (Lemma 5.1).

(B4): if t and u have the same root a' , then by (INV) and (2) $\text{rep}(a') \cdot E[u] = \text{rep}(u) = \text{rep}(t) \cdot g = \text{rep}(a') \cdot E[t]g$. So the element $E[t]gE[u]^{-1}$ of $V = P_1$ fixes the flag $\text{rep}(a')$. Since ρ is free, it is trivial, and there is no conflict.

(B5): suppose that after the merge two different corners m, m' of the component with root a' had values in the same coset, say $E[m] = E[m']c$ with $c \in C_v$. By (INV), which holds after the merge by the proof of Lemma 5.1, $\text{rep}(m) = \text{rep}(a') \cdot E[m'] \cdot c = \text{rep}(m') \cdot c = \text{rep}(m')c$. So $\text{rep}(m)$ lies in the corner m' , and $m = m'$. Hence there is no conflict.

(B6): by (INV), $\rho(X') = \text{rep}(m) \cdot a_k r_5 = \text{rep}(a') \cdot E[m]a_k r_5 = \text{rep}(a') \cdot E[w]d = \text{rep}(w) \cdot d = Y'$, because $d \in C_v \leq H$. So the forced value is a restriction of ρ . If the facet of X' is glued, then $\rho_s(X') = \rho(X') = Y'$. If it is unglued, the facet of Y' is unglued and different (Lemma 3.15), and the type class of the gluing $X' \mapsto Y'$ is the type class of that facet with respect to ρ , which is allowed. So there is no conflict, and the queued pair is a restriction of ρ .

(B7): when the pair (X', Y') is taken from the queue, the state is still extended by ρ and $\rho(X') = Y'$. So either the facet of X' is glued and $\rho_s(X') = Y'$, or both facets are unglued and `define_pair` (X', Y') is called with a restriction of ρ . $\square$

Note where freeness enters: (B1) uses the free action of P_4 , (B4) that of P_1 , and (B3) uses Corollary 3.10; (B5) and the forced values of (B6) hold for every gluing. The forced values include the consequences of the ridge relation of Lemma 4.1, since r_4 and r_5 lie in V . The engine looks for forced values only from the corners that were moved by a merge; this cannot lose a solution, since fewer deductions only mean fewer conflicts.

Theorem 5.3 (completeness of `fps_final`). *Let $0 \leq i \leq 139$ and $J \geq 1$, and assume that the tables are correct. If $\mathcal{S}'_i \neq \emptyset$, then one of the J jobs of `fps_final` with the options `-m B -t i -I ... -s j/J`, $0 \leq j < J$, reaches a complete state, provided the J jobs run to their normal end.*

Proof. Let $\rho \in \mathcal{S}'_i$. The trial of $t_i = \rho(1)$ for the facet K ends without conflict in a state extended by ρ (Proposition 5.2). The lists of the node of depth 1 do not depend on j . Let F be the facet on which this node branches, k the position of $\rho(x_F)$ in its list, and $j \equiv k \pmod{J}$. We apply Lemma 3.16 to the tree of the job j . (T) is Proposition 5.2. (L) holds by (B8) and (B9), without additional tests. (B) holds by (B10) and (B11) and the remark that a node always has a list that it has created or filtered. $\square$

Corollary 5.4. *Suppose that every job of the slice i ends with a line `DONE` and reports no solution and no rejected complete map. Then $\mathcal{S}'_i = \emptyset$.*

For the jobs of `fps_final` this follows from Theorem 5.3, for the jobs of `fps_v7` with `-sym` from Theorem 5.7 below. The two counts are the fields `solutions` and `complete_rejected` of the log.

A job that reaches a node limit prints `ABORTED-AT-NODE-LIMIT` instead of `DONE`, and every error exit ends the program without a final line (last lines of `main`). So a log with a line `DONE` comes from a search that was traversed completely.

5.4 The symmetry of the root gluing: rule (B13)

Fix a slice i and write $t = t_i$. The *root gluing* is the map ρ_0 on $K \cup tK$ with $\rho_0(\kappa) = t\kappa$ and $\rho_0(t\kappa) = \kappa$ for $\kappa \in K$. Every $\rho \in \mathcal{S}'_i$ extends it. Let

$$\mathfrak{S} = \{g \in K \cup tK : \rho_0(g) = gt\}.$$

Lemma 5.5. (a) $\mathfrak{S}$ is the set of all $g \in H$ such that the left multiplication λ_g maps $K \cup tK$ onto itself and commutes with ρ_0 . In particular $\mathfrak{S}$ is a subgroup of H . Explicitly, $\mathfrak{S} = C_K(t) \cup \{t\kappa : \kappa \in K, \kappa t\kappa^{-1} = t^{-1}\}$.

(b) If $\rho \in \mathcal{S}'_i$ and $g \in \mathfrak{S}$, then $\rho^g \in \mathcal{S}'_i$.

Proof. (a) If λ_g preserves $K \cup tK$ and commutes with ρ_0 , then $g = \lambda_g(1) \in K \cup tK$ and $\rho_0(g) = \lambda_g(\rho_0(1)) = gt$. Conversely let $g \in K \cup tK$ with $\rho_0(g) = gt$, and let $\kappa \in K$. Then $g\kappa \in K \cup tK$ and $\rho_0(g\kappa) = \rho_0(g)\kappa = gt\kappa = \lambda_g\rho_0(\kappa)$. Further $\lambda_g(t\kappa) = gt\kappa = \rho_0(g\kappa)$ lies in $K \cup tK$, and $\rho_0(\lambda_g(t\kappa)) = g\kappa = \lambda_g\rho_0(t\kappa)$. So λ_g maps $K \cup tK$ into itself and commutes with ρ_0 . The elements with this property form a group. For $g = \kappa \in K$ the condition $\rho_0(g) = gt$ reads $t\kappa = \kappa t$; for $g = t\kappa$ it reads $\kappa = t\kappa t$.

(b) ρ^g is a solution with the same type classes as ρ (Proposition 3.4 and Lemma 3.5). Since $g^{-1} \in \mathfrak{S}$, we have $\rho(g^{-1}) = \rho_0(g^{-1}) = g^{-1}t$ and $\rho^g(1) = g\rho(g^{-1}) = t$. $\square$

Now consider the node of depth 1 of the jobs of the slice. Its state, its lists and the facet A on which it branches are the same in all jobs of the slice. Let $X_A = x_A$ be the probe flag of A and L the list of A . The flags are the numbers $0, \dots, 14399$, and $<$ refers to this numbering. For $g \in \mathfrak{S}$ let $B_g = g^{-1}A$, a facet, and let $h_g \in K$ be defined by $g x_{B_g} = X_A h_g$. For every gluing ρ ,

$$\rho^g(X_A) = g\rho(g^{-1}X_A) = g\rho(x_{B_g}h_g^{-1}) = g\rho(x_{B_g})h_g^{-1}. \quad (3)$$

The engine `fps_v7` with `-sym` adds the following to (B1)–(B12).

(B13) (a) *Removal* (v7 lines 688–712 and 854). Before the node of depth 1 branches, an option $Y \in L$ is removed if $gYh_g^{-1} < Y$ for some $g \in \mathfrak{S}$ with $gA = A$. The sub-slice index of (B11) refers to the positions in the reduced list (v7 line 862).

(b) *Lex-leader test* (v7 lines 466–472, 491, 865–868). In the subtree of an option Y of the node of depth 1, beginning with the trial of Y itself, `define_pair` reports a conflict if one of the two facets it has just glued is B_g for some $g \in \mathfrak{S}$ with $gA \neq A$ and $g\rho_s(x_{B_g})h_g^{-1} < Y$.

Let $\mathcal{S}_i^{\min}$ be the set of all $\rho \in \mathcal{S}'_i$ with $\rho(X_A) \leq \rho^g(X_A)$ for all $g \in \mathfrak{S}$.

Proposition 5.6. (a) For every $\rho \in \mathcal{S}'_i$ there is $g_0 \in \mathfrak{S}$ with $\rho^{g_0} \in \mathcal{S}_i^{\min}$. In particular $\mathcal{S}_i^{\min} \neq \emptyset$ if $\mathcal{S}'_i \neq \emptyset$.

(b) Let $\rho \in \mathcal{S}_i^{\min}$ and $Y = \rho(X_A)$. Then Y is not removed by (B13)(a), and (B13)(b) reports no conflict in the subtree of Y as long as the state is extended by ρ .

Proof. (a) By Lemma 5.5 the finitely many gluings ρ^g , $g \in \mathfrak{S}$, lie in $\mathcal{S}'_i$. Choose g_0 such that $\rho' = \rho^{g_0}$ has the smallest value at X_A among them. For $g \in \mathfrak{S}$ we have $(\rho')^g = \rho^{gg_0}$, which is one of them because $\mathfrak{S}$ is a group. So $\rho'(X_A) \leq (\rho')^g(X_A)$.

(b) If $gA = A$, then $B_g = A$ and $x_{B_g} = X_A$, so $gYh_g^{-1} = \rho^g(X_A) \geq Y$ by (3); Y is not removed. If $gA \neq A$ and B_g is glued in a state s extended by ρ , then $g\rho_s(x_{B_g})h_g^{-1} = g\rho(x_{B_g})h_g^{-1} = \rho^g(X_A) \geq Y$, so the test reports no conflict. Only strict inequalities lead to a removal or a conflict. $\square$

Theorem 5.7 (completeness of `fps_v7` with `-sym`). *Let $0 \leq i \leq 139$ and $J \geq 1$, and assume that the tables are correct. If $\mathcal{S}'_i \neq \emptyset$, then one of the J jobs of `fps_v7` with the options `-m B -t i -I ... -bw 200 -sym -s j/J`, $0 \leq j < J$, reaches a complete state, provided the J jobs run to their normal end.*

Proof. By Proposition 5.6(a) there is $\rho \in \mathcal{S}_i^{\min}$. We apply Lemma 3.16 with $\mathcal{S} = \mathcal{S}_i^{\min}$ and this ρ . Up to and including the creation of the lists of the node of depth 1, the engine does exactly what `fps_final` does (the test (B13)(b) is switched on only in v7 line 865), so the state of that node is extended by ρ and its lists satisfy (L). The removal (B13)(a) is an additional test on the list L of A ; it does not reject $Y = \rho(X_A)$ by Proposition 5.6(b). The lists of this node were created with the trial of `fps_final`, whereas its children are made with the stronger trial described next. This is covered by Lemma 3.16: a trial that fails without (B13)(b) also fails with it, so a list created with the weaker trial contains every option that (L) requires for the stronger one. Let k be the position of Y in the reduced list and $j \equiv k \pmod{J}$. In the job j the option Y is followed. From here on the trial is “`define_pair` with (B1)–(B7) and (B13)(b), followed by the propagation”. It satisfies (T) by Propositions 5.2 and 5.6(b). The lists of the nodes below satisfy (L): they are created or filtered with this trial, or inherited from the node of depth 1. (B) holds as in the proof of Theorem 5.3. $\square$

Because (B13)(b) can refuse an option of a list that was filtered before the test was switched on, `fps_v7` switches off, in the subtree of Y , an internal consistency check of `fps_final` (an option of a list that was just filtered must survive its trial; line 800, v7 line 867). This check is a safeguard and not a pruning rule. The order of $\mathfrak{S}$ is 240 divided by the number of elements of d_i . The logs of `fps_v7` record $|\mathfrak{S}|$, the order of the subgroup fixing A , and the number of removed options (line `SYM`).

5.5 Which engine produced which logs

The 140 slices were divided by a capped probe run into 87 *light* slices and 53 *heavy* slices (`programB/make_jobs.py`).

- The 87 light slices ($i = 5, 10, 11, 15, 17, 21, 22, 27, 28, 32, 36$, and $62 \leq i \leq 139$ except 66, 67) were run with `fps_final` as one job each (`programB/jobs_B_light.txt`; 87 logs).
- The heavy slices 0, 53, 58, 59, 66, 67 were run with `fps_final` in 16 sub-slices each (`programB/jobs_B_heavy.txt`; 96 logs).
- The other 47 heavy slices were run with `fps_v7` and the additional options `-bw 200 -sym` in 16 sub-slices each (`programB/jobs_B_resume.txt` and 11 sub-slices of the slice 2 that were run before that list was written; 752 logs).

No slice mixes the two engines; this matters because the sub-slices of a slice must split one and the same list. The 752 logs of `fps_v7` contain 315 824 348 226 of the 326 322 489 739 trials. So for program B the proof of Theorem 5.7 carries most of the weight. In 41 of these 47 slices $|\mathfrak{S}| \geq 2$; in the slices 8, 16, 39, 43, 56, 60 the group $\mathfrak{S}$ is trivial and `-sym` has no effect.

6 The computation

6.1 Environment and commands

Both enumerations were run on one machine, an Apple M3 Max with 16 cores and 128 GB of memory under macOS 26.6.2 (Darwin 25.6.0), which was shared with other work. The C programs were compiled with Apple clang 21.0.0 (clang-2100.3.34.2, target arm64-apple-darwin25.6.0). The table generator of program A was run with Python 3.13.5 and numpy 2.5.2; the generator and the drivers

of program B use only the standard library of Python 3.13.5. All jobs ran with reduced priority (`nice -n 10`).

Program A. The table, the binary and the jobs were produced by the following commands, in the working directory of the program:

```
python3 gen_tables.py --order out/calib_B.txt tab/t5333B.bin 5 3 3 3
cc -O3 -march=native -Wall -o fsearch fsearch_production.c
./fsearch tab/t5333B.bin -k 1 -root t -ell 1 -keys 8 -sub S I
-sol out/B/solutions.txt -id B_tt_sIofS
```

The file `programA/out/calib_B.txt` fixes the order of the classes. The job list `programA/out/jobs_B.txt` contains 383 jobs: for every $t \in \{0, \dots, 153\}$ and a number $S = S_t$ of sub-slices that was chosen from a preliminary cost estimate, the jobs $I = 0, \dots, S_t - 1$. There are 107 classes with $S_t = 1$, and S_t takes the values 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 17, 18 for 5, 8, 11, 6, 5, 3, 2, 1, 1, 2, 2, 1 classes. The jobs were run by the queue `programA/run_queue.py`, which writes one log per job: the line `START` with the time and the command, the three lines printed by the program (`PARAMS`, `DEPTH`, and `SLICE ... DONE` with the counters), and the line `END`. The first job started on 2026-10-03 at 18:26:48 and the last ended on 2026-10-04 at 00:54:29 (local time, UTC+3).

Program B. The commands were

```
python3 gen_tables.py 5,3,3,3 tables/t5333 --inadm
cc -O2 -o fps_final fps_final.c cc -O2 -o fps_v7 fps_v7.c
./fps_final tables/t5333.tab -m B -I tables/t5333.inadm -fd 2 -F 16 -F2 12 -dr 3 -br 1
-S out/runs/solutions_B.txt -t i [-s j/16] -id id
./fps_v7 tables/t5333.tab -m B -I tables/t5333.inadm -fd 2 -F 16 -F2 12 -dr 3 -br 1
-bw 200 -sym -S out/runs/solutions_B.txt -t i -s j/16 -id id
```

with the job files named in Section 5.5. The light slices were run one after the other by `programB/runlight.sh`, the heavy ones by the driver `programB/run_slices.py`. Both write the output of a job to a temporary file and give it the name of the log only if the last line is the line `DONE` of that job. The first log was written on 2026-10-03 at 20:49 and the last on 2026-10-04 at 11:22 (local time).

Table 3 lists the SHA-256 hashes of the sources, binaries, tables and job lists. On macOS for arm64 the linker signs a binary with an identifier derived from the name of the output file, so the hash of a binary is reproduced only if the output name is the one given above.

6.2 Results

Program A. All 383 logs contain the line `DONE` of their own job with `solutions=0` and `leaves=0`. The totals are 1596 433 384 nodes, 46 799 574 720 candidates examined in the creation of lists (`attempts`), 428 044 087 522 trials (`tries`), and 131 311.58 seconds of cpu time. The numbers of nodes at the depths 1, ..., 6 are 369, 278 165, 199 409 287, 1 279 176 706, 117 476 154 and 92 703. The search never went deeper than depth 6: after three or four branchings the propagation determines so much that every branch dies. The 14 jobs of the empty slices of Section 4.4 report 0 nodes.

Program B. All 935 logs end with the line `DONE` of their own job with `solutions=0`, and all report `complete_rejected=0`. Table 4 shows the totals by engine. The numbers of trials at the depths 1, ..., 6 are 200 652 480, 12 039 047 854, 148 709 163 452, 163 478 410 996, 1 895 037 701 and 176 321. Together with the trial of the root gluing, which every log counts as one trial of depth 0 (935 in all), they add up to the total. In 45 slices the node of depth 1 is already dead.

file	SHA-256
<i>program A</i>	
fsearch_production.c	6f4ad4337cc490fb1a3f640b64f16e8ba546dfe4e388eaab5974fac3851ac3ef
fsearch (binary)	1f5e15d2b28e759cc9b5fab0e5068a2516dce63cddb5c92d057d1cf027bac316
gen_tables.py	1d83b42afadee1e14f770eabdfb44b95147e034bfc9e60353ce61e405c586de1
out/calib_B.txt	671928558d7eda90fc96775ff22472c6a58f54c6668cf4cb309ab259068fdbc6
tab/t5333B.bin (table, 10.9 MB)	c43c166515ab7b70ce1e0a61790292d385414a3e452398e526e6559772ac76e5
out/jobs_B.txt	17d347ab0de12bdf55101c3f544a873a99449ab232a4a1e20aced5c6ad74306a
<i>program B</i>	
fps_final.c	a05d00b7a6fcb96353aacecfbb8d926f226f2123040191454ea0fe620a2669ac
fps_v7.c	f032e5f76396eb6a13fbd4869027786f69420af1e5f4d722b865fc1bf6301d12
fps_final (binary)	ac1a5420c4d22280b616444fb686c06eaf9401f68569640bea1e1e09ab953689
fps_v7 (binary)	17a9410a7afb357c4b437174cab13e3a659fa408975694149394ba42e7016cc6
gen_tables.py	0639de316948bdb732717f845ac20b7f58757f3641f3c171cdcd77b105623a0b
tables/t5333.tab	896270810cf047d700e2a9a657e505140d771444c2e1a6322bfd3365ccacf27d
tables/t5333.inadm	f5476e1a5541f425068d8156e4413966ed812c0adb866fe8eccb7ca6551a1127
jobs_B_light.txt	cccbc010dcacca78f7dc2136dfc4ff9665ff4fae01def3cd154fa2d7b3c42ade
jobs_B_heavy.txt	0891066a74e963dd85c6ca4d8db66a05254737dfc9231b1113fde25107f6d7e3
jobs_B_resume.txt	8cbbc67310113c38ac7442aab36060661f87be9b1bc3ee8e664d4f8707c4f352

Table 3: Hashes of the inputs of the two enumerations. The binaries and the three tables are not part of the reproducibility package; its script `setup.sh` rebuilds them, and we obtained the hashes above again when we ran it.

	slices	logs	work	cpu time
program A, fsearch	154	383	1 596 433 384 nodes	36.48 h
program B, fps_final, light slices	87	87	1 364 923 686 trials	0.24 h
program B, fps_final, heavy slices	6	96	9 133 217 827 trials	2.13 h
program B, fps_v7 with -bw 200 -sym	47	752	315 824 348 226 trials	62.78 h
program B, total	140	935	326 322 489 739 trials	65.15 h

Table 4: Work and cpu time, summed from the logs. All 1318 logs report 0 solutions.

No complete gluing was reached. In no production job of either program did the search arrive at a state in which all 120 facets are glued. So the leaf tests (A13) and (B14) were never executed in the production runs, and Theorem 1.2 rests entirely on the necessity of the pruning rules, that is, on Theorems 4.3, 5.3 and 5.7, on the agreement of the two sources with the rules as we have described them, and on the correctness of the tables. Section 7 describes how the last two points were tested.

Proof of Theorem 1.2. By (1) we have to show that there is no solution. By Proposition 3.11 it suffices that $\mathcal{S}_0 = \dots = \mathcal{S}_{153} = \emptyset$, and this follows from Corollary 4.4 and the 383 logs of program A. Independently, by Proposition 3.12 it suffices that $\mathcal{S}'_0 = \dots = \mathcal{S}'_{139} = \emptyset$, and this follows from Corollary 5.4 and the 935 logs of program B. $\square$

Each of the two enumerations alone proves the theorem, under the assumption that the corresponding program and its tables do what Section 4, respectively Section 5, says.

6.3 Logs, checksums and acceptance tests

The logs are text files of at most a few lines. We use two aggregate checksums. For program A: take from each of the 383 logs the line `SLICE ... DONE`, delete the field `cpu=...` at its end, sort the 383 lines, end each with a newline character, and apply SHA-256. For program B: for the 935 file names `B_t*.log` in sorted order, feed the bytes of the name followed by the 32 bytes of the SHA-256 digest of the file into one SHA-256. The results are the two values given after Theorem 1.2. The

script `checks/log_checksums.py` computes both, and `checks/SHA256SUMS_logs_programA.txt` and `checks/SHA256SUMS_logs_programB.txt` list the digests of the individual logs.

The summary scripts of the two programs accept a log if it contains a `DONE` line with the right identifier. An independent verification run found this too weak (Section 7.7) and wrote two strict checkers, `checks/check_p1.py` and `checks/check_p2.py`, which are the acceptance tests of the final state. They recompute the set of slices from the tables, respectively from the job files; require exactly one log per job and no other file; parse every line of every log; compare the identifiers, options and engine recorded in the log with the job; require `solutions=0`; check the internal consistency of the counters; and verify that the sub-slices of every slice split one and the same list, using the list lengths recorded in the logs and a probe of the first branching made with rebuilt binaries. Their outputs on the packaged logs are `checks/out_check_p1.txt` and `checks/out_check_p2.txt`; both end with `RESULT: PASS (0 failures)`.

7 Controls and rejection tests

A search that returns nothing is easy to get wrong. This section reports what was done to test the two programs, their tables and their logs. Most of it is the work of two independent verification runs, made after the enumerations were complete: a *mathematical* run, which re-derived all proofs, wrote a third search engine of its own and compared it with both programs, and a *computational* run, which examined the logs, binaries and job lists, repeated parts of the computation, and ran controls and rejection tests. Their scripts and outputs are in the folders `checks/`, `controls/`, `mutations/` and `verification_engine/` of the reproducibility package. We ran the controls, the mutation tests, the strict checkers and several scripts of the third engine again from the package; the outputs agreed with those of the verification runs (details in the file `README.md` of the package). A third verification run, which examined the written note and the package with code of its own, is described in Section 7.8.

7.1 Tables and binaries

All tables of both programs were regenerated from the sources and are bit-identical to the tables used in the production runs. The rebuilt binaries are bit-identical to the production binaries (Table 3). The source `fsearch_production.c` is a copy made after the binary was built; its hash equals the hash recorded for the source at the start of the production run.

The mathematical verification run constructed H independently, as a group of matrices over $\mathbb{Z}[\varphi]$ (`verification_engine/own/cox.py`, `h4_classes.py`), and recomputed the classes, the loop classes, the flagged elements and the inverse classes. The script `compare_tables.py` then compares every table entry that the engines read: the regular representations, the facets and their representatives, the ridge representatives, the 188 classes with their flags and inverse classes, all $120 \cdot 14400$ entries of the class table of program A, the representation of V used by program B, its 14400 flags and its 140 type classes. All entries agree. The computational run verified in addition that the two programs exclude the same set of 1680 elements and that the 140 type classes of program B correspond to the 126 self-inverse classes and the 14 pairs of program A (`checks/tables/map_classes.py`).

7.2 Reruns

With rebuilt binaries, 102 of the 383 jobs of program A and 52 of the 935 jobs of program B were run again: a random sample with a fixed seed, all jobs of program A that had taken less than two minutes, and a few longer ones. In all 154 reruns every counter of the log was reproduced (`checks/reruns/`). The reruns cover only 2.2% of the nodes of program A and 1.5% of the trials of program B; the long jobs were not repeated. For 25 jobs of program A the search was also repeated with the generic ridge

group and cell	labelled	classes	found by the programs
[3, 5, 3], icosahedron	432	7	A: 12 solutions in its slices, 7 classes, for $S = 1, 2, 3, 5, 16$; B: 12 (<code>fps_final</code>) and 8 (<code>fps_v7</code>), 7 classes, for $J = 1, 2, 3, 16$
[5, 3, 5], dodecahedron	698	12	A: 26 solutions, 12 classes; B: 26 and 16 solutions, 12 classes; same values of S and J
[3, 4, 3, 3], 24-cell	14999	149	A: 239 solutions in 41 slices, 149 classes; B: 337 and 267 solutions in 54 slices, 149 classes, for $J = 1$ and $J = 16$

Table 5: Positive controls with the production options (one cell in each case; [3, 4, 3, 3] has rank 5 and is Euclidean). The numbers of labelled solutions and of classes were obtained by the third engine, which implements Definition 3.3 literally. A “class” is an orbit under relabelling, that is, a conjugacy class of torsion-free subgroups.

propagation and without the quick test (A8): the search trees were identical, level by level. Before the verification, 103 complete slices of program A had been run a second time with the plain search `gsearch` (80 356 268 nodes, no solution; logs in `programA/out/B_gs/`); three of these were repeated with identical counters. A second random sample of 20 jobs was run again in the last verification run (Section 7.8).

7.3 Positive controls

The programs accept other string Coxeter groups, so they can be run on cases that have solutions. Table 5 lists the controls that were run with the production binaries and the production options, including the sub-slice options and, where it exists, the file of flagged elements. None of the numbers in the table is new: the one-cell manifolds of [3, 5, 3] and [5, 3, 5] are the icosahedral and dodecahedral manifolds of the classification described by Everitt [Eve04], and the first homology groups of our classes agree with his lists (for [5, 3, 5] we find nine classes up to the diagram automorphism against eight manifolds; two of our classes have equal invariants and presumably are two tessellations of one manifold, which we did not prove).

The third control is the most relevant one: the group [3, 4, 3, 3] of the Euclidean honeycomb by 24-cells, where the cell group has order 1152. It is our only control with one cell, rank 5 and last label 3, and we do not know another group of this shape that has solutions. Of the six groups $[p, q, r, 3]$ with $p, q, r \geq 3$ and a finite cell group $[p, q, r]$, the groups [3, 3, 3, 3] and [4, 3, 3, 3] are finite; [3, 3, 4, 3] has no torsion-free subgroup of index 384, the order of its cell group, because 384 is not divisible by the order 1152 of its finite subgroup [3, 4, 3]; [5, 3, 3, 3] is the subject of this note; and [3, 3, 5, 3], whose vertex group [3, 5, 3] is infinite, was not examined. The mathematical verification run enumerated all 14999 labelled solutions with its own engine and then compared the *sets* of solutions slice by slice: every slice of program A and of `fps_final` contains exactly the solutions predicted by Propositions 3.11 and 3.12, and every slice of `fps_v7` with `-sym` contains a relabelling under $\mathfrak{S}$ of every predicted solution ($|\mathfrak{S}|$ is up to 96 there). The elliptic filter of program A is not available for this group, because the label 4 needs $\sqrt{2}$ instead of φ ; program A was run with its other production options. In this control the leaf test of program B is exercised: `fps_final` rejects 27 complete gluings and `fps_v7` rejects 6.

Program B has in addition a validation suite on 24 small groups of rank 3 to 5, in which the engine is compared with a brute-force enumeration of subgroups of low index and with plain backtracking (`controls/programB/validate.sh`, `validate_v7.sh`): 35 comparisons for `fps_final` and 37 for `fps_v7` with `-sym`, all in agreement. This suite does not use the file of flagged elements or sub-slices.

Two further controls concern known manifolds. The Davis manifold is found by `fps_final` in two restricted searches in [5, 3, 3, 5] (one solution each, after 3042 448 and 12 445 487 trials; logs in

`controls/programB/out/davis/`); the solution is torsion-free and orientable with $H_1 = \mathbb{Z}^{24}$, the value given for the Davis manifold in [Mar18, Section 2.2]. In $[5, 3, 3, 3]$ itself the subgroup Σ of Conder and Maclachlan was converted into a gluing of eight 120-cells, and the gluing maps of the first n of its 960 facets were prescribed (`controls/programA/run_cm_controls.sh`). For $n = 60$ and $n = 50$ the plain search and the development version of the engine of program A rebuild Σ by propagation alone; for $n = 45$ and $n = 40$ they find two completions, Σ and a second torsion-free gluing of eight cells. Both are non-orientable; their first homology groups are $\mathbb{Z}^2 \oplus (\mathbb{Z}/2)^2$, as stated for Σ in [CM05], and $\mathbb{Z}^2 \oplus \mathbb{Z}/2$. We do not know whether the second one is among the examples of [Lon08] and claim nothing about it. This control shows that the pruning rules of program A, with the tables of the real problem, do not reject a genuine torsion-free gluing in $[5, 3, 3, 3]$. It uses the threshold $t = 0$ and several cells, and needs an option that the production binary does not have; so it tests the shared code of glue and propagation, not the production binary.

7.4 Mutation tests

To see whether the controls can detect errors, the sources were mutated on purpose and the mutants were run on the controls $[3, 5, 3]$ and $[5, 3, 5]$ (`mutations/`). For program A: a threshold that is too strong, a sub-slice selector that skips one residue class, a job list with one sub-slice missing, and a quick test that rejects too much all lose classes (for example 2 instead of 7 and 12 classes for the threshold, 5 instead of 7 for the quick test). Removing the loop test, lowering the threshold or removing the test of the inverse type leaves the classes unchanged and increases the number of nodes, as it should. Without any ridge propagation the search reaches hundreds of thousands of complete gluings, all of which the leaf test rejects. For program B: forbidding the root type after the root gluing, a wrong file of flagged elements, a wrong sub-slice selector and a lex-leader test with $\leq$ instead of $<$ lose classes on at least one of the controls; ignoring a holonomy conflict (B4) leaves the classes unchanged, and the leaf test then rejects 171 and 936 complete gluings. The mutant of the removal (B13)(a) with $\leq$ is not detected by $[3, 5, 3]$ and $[5, 3, 5]$, but by four other groups of the validation suite of `fps_v7`.

The mutation tests also showed what the small controls cannot see. The test “type and inverse type at least t ” is implemented twice in program A (in the table of allowed classes and for both facets in (A4)), and with the elliptic filter the loop classes are excluded twice; so a mutation of one of these lines has no effect at all. The group $[5, 3, 5]$ has last label 5 and does not exercise the ridge propagation for label 3 or the quick test, and it has no elliptic class and no pair of inverse classes; $[3, 5, 3]$ has one elliptic class, two pairs of inverse classes, and its whole search has 132 nodes. The code paths that matter for $[5, 3, 3, 3]$ are therefore covered mainly by other means: the control $[3, 4, 3, 3]$, the reruns with generic propagation, the table comparisons, and the comparisons with the third engine on the real problem described next.

7.5 Comparisons on the real problem

Cross-check of partial gluings. For 57 partial gluings of one cell of $[5, 3, 3, 3]$, taken from depths 2 and 3 of the search tree of program A, the number of all completions (without threshold) was computed with up to five programs: the plain and the list-based search of program A, the auxiliary counter, and the engine of program B with and without its type filter (`checks/crosscheck/`; the engine of program B needed a small patch to accept a prescribed partial gluing). Every finished run gave 0; in 51 of the 57 states all five programs finished. As all counts are zero, this test can only detect a program that finds a completion where another does not. On 353 partial gluings of $[3, 5, 3]$ and $[5, 3, 5]$, many of which have completions, the five programs gave equal counts.

The third engine. The engine `verification_engine/own/direct_mask.c` of the mathematical verification run propagates only the relation (R2) and tests freeness at the leaves; it uses its own tables and none of the data structures of the two programs. On the real problem it gave the following.

- It searched 52 of the 154 slices of program A completely, with the slice definition of Section 3.5: the 14 empty slices, 12 slices that die at the first node, and 26 larger ones (732 580 nodes in total). It found no solution, and in each of the 26 slices its number of nodes equals that of the plain search `gsearch` of program A. This is a spot check and not a third enumeration: the other 102 slices were beyond its means.
- It reproduced the first node of every slice of both programs: for program A the outcome (root refused, a list empty, or the length of the first branching list) in 154 of 154 slices; for program B the numbers of lists, trials and surviving options in 140 of 140 slices.
- At about 1400 sampled nodes of depth 2 to 4 it recomputed the candidate lists for the same partial gluing: 209 nodes of program A, 595 nodes of `fps_final` and 627 nodes of `fps_v7` with `-sym`. Every list that the program had just created or filtered was equal to the exact set of candidates computed by the third engine, and every inherited list contained that set, as Lemma 3.16 requires. For program B the third engine was extended by its own implementation of the vertex condition, and for `fps_v7` the lex-leader condition (B13) was evaluated independently; 495 options were excluded by that condition alone.
- For all 47 slices run with `-sym` it recomputed $|\mathfrak{S}|$, the subgroup fixing A , the number of removed options and the number of nodes of depth 2; all agree with the logs.

On a further sample of 47 nodes the vertex condition of program B removed no candidate beyond the ridge relation and the type filter, in agreement with Remark 3.7.

7.6 Manipulated logs

Copies of the logs were manipulated on purpose: a log deleted, truncated, changed to one solution, changed to a stopped run, replaced by the log of another sub-slice with edited identifiers, replaced by a run of the other engine, or an extra log added. The strict checkers reject all 14 manipulated sets and accept the two unmodified ones (`checks/manipulated_logs/`).

7.7 Weaknesses found, and how they were closed

The verification runs found the following weaknesses. None of them changed a count, but a reader should know them.

1. *Validation of program B without the production options.* The original validation suite of program B never used the file of flagged elements or the sub-slice option. Closed by the controls of Table 5, which use both, and by a control in which all 120 root flags of $[3, 5, 3]$ and $[5, 3, 5]$ were prescribed, with and without the file (432 and 698 labelled completions in all cases).
2. *Weak summary scripts.* The summary scripts of both programs accept a log by its DONE line and identifier only, do not notice a renamed log of another sub-slice or a mixture of engines in a slice, and always exit with status 0. Closed by the strict checkers of Section 6.
3. *Validation of program A not tied to commands.* Some validation scripts of program A call development binaries that were not kept, and one validation row has no command log. The controls of Section 7.3 were therefore run again with rebuilt binaries; the remaining rows of the

original validation table (among them cases with two cells) were not repeated and are not used in this note.

4. *Source newer than binary.* See Section 7.1. The source of the plain search `gsearch.c` was changed after the production run had started; the stored logs of `gsearch` are reproduced by the present source on the slices that were rerun.
5. *Leaf tests never used.* Stated in Section 6. This is not a defect, but it moves all the weight to the pruning rules. It is the reason for the rule-by-rule proofs of Sections 4 and 5, for the mutation tests and for the comparisons of Section 7.5.
6. *Most of program B depends on `-sym`.* 47 of the 53 heavy slices, with 97% of the trials, were run with the engine `fps_v7` and the reduction (B13); in six of them about half of the options of the first branching are refused by the lex-leader test. The first summary of the results named only `fps_final`. The reduction is an additional symmetry argument and needs its own justification: this is the proof in Section 5.4, together with the mutation tests, the control [3, 4, 3, 3], and the recomputation of the reduction on all 47 slices. For these slices the independence of the two programs remains the main safeguard.
7. *Minor points of the code.* (a) The log of program A does not print the node limit; this is harmless, because a run that reaches a node limit prints `STOPPED` instead of `DONE`. (b) In `fps_v7` the internal consistency check mentioned in Section 5.4 is switched off in the whole subtree of an option and not only where the lex-leader test can interfere; this weakens a safeguard, not the search. (c) The first line of the logs of `fps_final` does not print all steering options; they were confirmed by recomputing the first node of every slice.

Five jobs of program B were started twice, because the driver was interrupted during their first start; the partial output of the first start was overwritten, and each of these jobs has one complete log.

What was *not* done: the long jobs were not repeated; there is no third complete enumeration; and Belolipetsky's theorem, which we use only for context, was not re-verified.

7.8 A further verification run

After this note and its package had been written, a third verification run, again AI-assisted, examined both. It read the three sources completely and compared them with the rules (A1)–(A13) and (B1)–(B14), including the converse question whether anything is discarded outside these rules; it read the proofs of Sections 2 to 5 again; and it wrote new code from scratch, which is in the folder `independent_run_2/` of the package. The results are as follows.

- *Classes and tables.* H was generated once more as a group of matrices over $\mathbb{Z}[\varphi]$. The numbers of Proposition 3.9 and of Table 2 were obtained again. The flagged elements were determined by the literal test of Section 3.3. For each of the other 12720 elements z the run found a power of the matrix of zr_5 whose trace has absolute value greater than 5 in a real embedding of $\mathbb{Z}[\varphi]$. A 5×5 matrix of finite order is diagonalizable with roots of unity as eigenvalues, so this proves that zr_5 , and hence its inverse r_5z^{-1} , has infinite order. Every table entry that the engine of program A uses in the production runs, among them the $120 \cdot 14400$ entries of its class table, and the file of flagged elements of program B agree with this construction.
- *First node.* A new implementation of the root step and of the trial (the gluing of Definition 3.3, closed under the ridge relation of Lemma 4.1, with the type conditions of the slice) reproduced the first node of every slice: for program A the 14 refused roots, the 13 slices with an empty first

list and the length of the first branching list in the other 127 slices; for program B the numbers of lists, trials and surviving options in all 140 slices. With its own implementation of the choice of the branching list, of the sub-slice selection and of the rule (B13), the run also reproduced the number of nodes of depth 2 for every job separately: in 356 of 356 jobs of program A and in 890 of 890 logs of program B, together with the line `SYM` of all 752 logs of `fps_v7`. So every job followed its own share of one and the same list. Nothing below depth 2 was recomputed in this run.

- *Controls.* A literal enumeration for $[3, 5, 3]$ gave 432 solutions in 7 classes. Program A was run on $[3, 4, 3, 3]$ with its production options and gave 239 solutions in 149 classes, as in Table 5.
- *Reruns.* A new random sample of 12 jobs of program A and 8 jobs of program B, drawn from the jobs with a recorded cpu time between 5 and 300 seconds that had not been run again before, reproduced every counter. Altogether 174 jobs have now been run twice; they contain 5.0% of the nodes of program A and 1.9% of the trials of program B.
- *Package.* The hashes of Table 3, the strict checkers, the log checksums, the positive controls of Table 5 and all totals of Section 6 were reproduced from an extracted copy of the package.

The run found no error in the proofs and no difference between the rules and the code. It required corrections of the text, among them the references to [RT00, Lon07] and a more complete list of the inactive parts of the two sources.

8 What remains

The smallest index. By Corollary 1.3(b) the smallest index of a torsion-free subgroup of W is 14400 m_0 with $2 \leq m_0 \leq 8$. The values $m_0 = 2, \dots, 7$ are open.

Index 28800. The case that matters for Problem 4.124 is $m = 2$ with the additional condition $G \leq W^+$: two 120-cells, and a gluing ρ of $\{0, 1\} \times H$ that reverses the parity of the flags. Both programs have a mode for this case; they cut it into 72 slices and one additional slice (program A) and into 780 slices (program B). We do not reproduce the corresponding coverage arguments, since no result is claimed. Neither enumeration is complete, and the following numbers prove nothing about index 28800. They are taken from capped probe runs, whose logs are in the folder `index28800_probes/` of the package.

- Program A probed 9 of its 72 slices with a limit of 150 000 nodes each. Two finished without a solution (one of them has an empty root); seven reached the limit, at depth 8 or 9 of the search. The additional slice in which no facet is glued inside its own cell is empty.
- Program B finished its 206 cheapest slices (137 228 027 trials, no solution). Of 12 further slices, probed with a limit of $3 \cdot 10^7$ trials each, five ended at once and seven reached the limit.

The extrapolations made from these probes during the development of the two programs are 10^3 to 10^4 core-hours for program A and $5 \cdot 10^2$ to $5 \cdot 10^3$ core-hours for program B. They are crude, because they assume that the visited part of the tree is typical.

The search tree explodes for a structural reason. With one cell, every gluing constrains the same 120 facets, the ridge relation forces further gluings at once, and every branch dies at depth at most 6. With two cells, after the first gluing between the cells the gluings of the facets of the second cell are hardly constrained until many of its facets are glued. The working notes of program A record one example: two of the 3347 nodes of depth 2 of one slice led to 573 nodes at depth 3, 5747 at depth 4

and more than seven million at depth 5. A complete enumeration of index 28800 needs either a much stronger propagation between the two cells, for instance pairwise compatibility relations between the candidate lists of adjacent facets, or a further symmetry reduction, or a computing budget that we do not have. We leave it open, and with it Problem 4.124 in the arithmetic case. A non-orientable closed hyperbolic 4-manifold with $\chi = 2$ covering $\mathbb{H}^4/W$ would correspond to a torsion-free subgroup of index 28800 not contained in W^+ ; this case was not attempted at all.

Verification. The claims were checked as follows.

1. The two enumerations were made by two programs that were written separately from the statement of Proposition 3.4, with different tables, data structures, pruning rules and symmetry reductions (Sections 4 and 5). Both report 0 solutions.
2. Two independent verification runs, which were AI-assisted, examined the result after the enumerations were complete. The first re-derived every proof of Sections 2 to 5, including a verdict with a proof for each of the 13 + 14 pruning rules, checked the cited theorems on finite subgroups and on the faithfulness of the geometric representation in the text of [Dav08], wrote a third search engine and made the comparisons of Section 7.5. The second checked the coverage of the job lists, the integrity of the 1318 logs, the correspondence of binaries and sources, reran 154 jobs, and ran the controls, mutation tests and manipulated-log tests of Section 7. Both runs confirmed the result. They required corrections of the wording (the Euler characteristic form needs “finite-sheeted”; the role of `fps_v7` had to be stated) and found the weaknesses listed in Section 7.7.
3. For this note all proofs were written out again and compared, rule by rule and line by line, with the two sources. The tables and binaries were rebuilt from the reproducibility package with the hashes of Table 3; the strict checkers, the log checksums, the positive controls of both programs, the Conder–Maclachlan control, all mutation tests, the manipulated-log tests, the comparison of the stored reruns, the probes of the first branching and seven scripts of the third engine (classes, table comparison, first node of all slices of both programs, the two recomputations of the symmetry reduction, sub-slice consistency) were run again from the package and reproduced the recorded outputs.
4. A third AI-assisted verification run then examined the written note and the package (Section 7.8): the proofs, the agreement of the rules with the three sources, the classes and the first node of every slice with new code, a new sample of reruns, the numbers, the quotations and the literature. It did not repeat the long jobs, recomputed nothing below depth 2 of the search trees, and did not read [Lon07, RT01].
5. After that run the thesis [Lon07] was obtained. Its Chapter 7, the closing remarks of [Dav85] and Section 5.3 of [Che25] were read, and the account of earlier work in the introduction and in this section was revised accordingly. The statements, the proofs and the computations of the note were not changed.

Not checked by anyone: a complete repetition of the long jobs (the jobs that were run twice contain 5.0% of the nodes of program A and 1.9% of the trials of program B), a third complete enumeration, Belolipetsky’s theorem, and everything concerning index 28800 beyond the probes described above. The remaining risk is an error common to both programs in a place that the controls, the mutation tests and the partial third enumeration do not reach; we consider it small but cannot exclude it.

Scope and priority. The result is Theorem 1.2 with its equivalent forms: the non-existence of a torsion-free subgroup of index 14400 in $[5, 3, 3, 3]$, of a closed hyperbolic manifold glued from one 120-cell with dihedral angle $2\pi/3$, and of a finite-sheeted manifold cover of $\mathbb{H}^4/W$ with Euler characteristic 1. It is a computer-assisted theorem. It makes no claim on Problem 4.124 of [BKR26]: the three questions of that problem concern Euler characteristic 2 and remain open, also in the arithmetic case. To our knowledge the result is new. The question itself is not new. Davis [Dav85] raised the possibility of a torsion-free subgroup of index 14400, [RT00] states the search problem for one or two 120-cells, [Lon07, Chapter 7] calls the question outstanding and contains the case of normal subgroups, and [CM05, Lon07] approach the question through subgroups of low index. We read [CM05] in full, the relevant statements of [Bel04, Bel07] in the arXiv versions, [Eme14, Mar18] in renderings of the arXiv versions, and the relevant section of [RT00] in a rendering of the arXiv version of the proceedings. Of [Lon07] we read the abstract, Section 1.3, Chapter 7 and Chapter 8 in the scanned copy of the university repository; we did not check its proofs. We read the closing remarks of [Dav85] in a scan of the journal pages, and the introduction and Section 5.3 of [Che25] in the arXiv version. We know [Lon08, CL23, MZ21, CK22] from their abstracts only, and we quote the result of [RT01] from [Mar18]. The problem statement is quoted from the dataset record [ula26]; the remarks to the problem and its attribution were read in the authors’ preliminary version of [BKR26], and the published volume was not consulted. Searches of the arXiv listings, of OpenAlex (including the works that cite [CM05, Lon08, Eme14, Bel04]), of Crossref and of zbMATH, and five web searches, found no work that decides whether $[5, 3, 3, 3]$ has a torsion-free subgroup of index 14400 or 28800, or whether a closed hyperbolic 4-manifold with Euler characteristic 1 or 2 exists. Apart from [Lon07], theses and unpublished computations are not covered by these searches, and specialists may know more; this negative search is not a proof of priority. The method is classical [Eve04, CM05, CL23], and we claim no novelty for the control cases: the one-cell manifolds of $[3, 5, 3]$ and $[5, 3, 5]$, the Davis manifold and the subgroup of Conder and Maclachlan are known. Of the cited books, the statements taken from [Dav08] (Theorems 12.3.4(i) and 6.12.10, formula (16.8)) were checked in the text; the exercise of [Bou02] was identified through the citation in [Dav08]; the other books are cited for standard facts at the level of chapters.

AI-assisted tools were used to search the problem corpus and the literature, to find the arguments, to write and run the search and verification programs, and to help prepare this manuscript. The author has reviewed the proofs, the computations and the sources and is responsible for all claims. The note has not been peer reviewed.

References

- [Bel04] Mikhail Belolipetsky. On volumes of arithmetic quotients of $SO(1, n)$. *Annali della Scuola Normale Superiore di Pisa, Classe di Scienze* (5), 3(4):749–770, 2004. arXiv:math/0306423. doi:10.2422/2036-2145.2004.4.04.
- [Bel07] Mikhail Belolipetsky. Addendum to: On volumes of arithmetic quotients of $SO(1, n)$. *Annali della Scuola Normale Superiore di Pisa, Classe di Scienze* (5), 6(2):263–268, 2007. arXiv:math/0610177. doi:10.2422/2036-2145.2007.2.02.
- [BKR26] R. İnanc Baykur, Robion Kirby, and Daniel Ruberman, editors. *K3: A New Problem List in Low-Dimensional Topology*, volume 295 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2026. Problem 4.124 is in Chapter 4, “4-manifolds” (pp. 179–289 according to the publisher’s record of the chapter, doi:10.1090/surv/295/04). We used the authors’ preliminary version, in which the problem is on pp. 292–293. doi:10.1090/surv/295.

- [Bou02] Nicolas Bourbaki. *Lie Groups and Lie Algebras, Chapters 4–6*. Elements of Mathematics. Springer, Berlin, 2002.
- [Bro82] Kenneth S. Brown. *Cohomology of Groups*, volume 87 of *Graduate Texts in Mathematics*. Springer, New York, 1982. doi:10.1007/978-1-4684-9327-6.
- [Che25] Jacopo G. Chen. Non-cobordant hyperbolic manifolds. Preprint, arXiv:2501.11610, 2025. Version 1 of 20 January 2025.
- [CK22] Marston Conder and Ruth Kellerhals. Small-volume hyperbolic manifolds constructed from Coxeter groups. *Bulletin of the London Mathematical Society*, 54(5):1705–1720, 2022. doi:10.1112/blms.12651.
- [CL23] Marston Conder and Georgina Liversidge. Small volume 3-manifolds constructible from string Coxeter groups of rank 4. *Journal of Algebra*, 635:775–789, 2023. doi:10.1016/j.jalgebra.2023.06.043.
- [CM05] Marston Conder and Colin Maclachlan. Compact hyperbolic 4-manifolds of small volume. *Proceedings of the American Mathematical Society*, 133(8):2469–2476, 2005. doi:10.1090/S0002-9939-05-07634-3.
- [Cox56] H. S. M. Coxeter. Regular honeycombs in hyperbolic space. In *Proceedings of the International Congress of Mathematicians, Amsterdam, 1954, Vol. III*, pages 155–169. North-Holland, Amsterdam, 1956.
- [Cox73] H. S. M. Coxeter. *Regular Polytopes*. Dover Publications, New York, third edition, 1973.
- [Dav85] Michael W. Davis. A hyperbolic 4-manifold. *Proceedings of the American Mathematical Society*, 93(2):325–328, 1985. doi:10.1090/S0002-9939-1985-0770546-7.
- [Dav08] Michael W. Davis. *The Geometry and Topology of Coxeter Groups*, volume 32 of *London Mathematical Society Monographs Series*. Princeton University Press, Princeton, NJ, 2008. doi:10.1515/9781400845941.
- [Eme14] Vincent Emery. On compact hyperbolic manifolds of Euler characteristic two. *Algebraic & Geometric Topology*, 14(2):853–861, 2014. doi:10.2140/agt.2014.14.853.
- [Eve04] Brent Everitt. 3-manifolds from Platonic solids. *Topology and its Applications*, 138(1–3):253–263, 2004. arXiv:math/0104182. doi:10.1016/j.topol.2003.08.025.
- [Hum90] James E. Humphreys. *Reflection Groups and Coxeter Groups*, volume 29 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1990. doi:10.1017/CB09780511623646.
- [Lon07] Cormac Diarmuid Long. *Epimorphic images of simplicial Coxeter groups and some associated hyperbolic manifolds*. PhD thesis, University of Southampton, 2007. Record 466178 of the repository of the university, <https://eprints.soton.ac.uk/466178/>.
- [Lon08] Cormac Long. Small volume closed hyperbolic 4-manifolds. *Bulletin of the London Mathematical Society*, 40(5):913–916, 2008. doi:10.1112/blms/bdn077.
- [Mar18] Bruno Martelli. Hyperbolic four-manifolds. In Lizhen Ji, Athanase Papadopoulos, and Shing-Tung Yau, editors, *Handbook of Group Actions, Vol. III*, volume 40 of *Advanced Lectures in Mathematics*, pages 37–58. International Press, Somerville, MA, 2018. arXiv:1512.03661.

- [MZ21] Jiming Ma and Fangting Zheng. Orientable hyperbolic 4-manifolds over the 120-cell. *Mathematics of Computation*, 90(331):2463–2501, 2021. doi:10.1090/mcom/3625.
- [Rat19] John G. Ratcliffe. *Foundations of Hyperbolic Manifolds*, volume 149 of *Graduate Texts in Mathematics*. Springer, Cham, third edition, 2019. doi:10.1007/978-3-030-31597-9.
- [RT00] John G. Ratcliffe and Steven T. Tschantz. Gravitational instantons of constant curvature. In: *Cosmological Topology in Paris 1998* (Vincent Blanlœil and Boudewijn F. Roukema, eds.), electronic workshop proceedings, arXiv:astro-ph/0010170, 2000. An article with the same title appeared in *Classical and Quantum Gravity* 15 (1998), no. 9, 2613–2627, doi:10.1088/0264-9381/15/9/009; we did not consult it.
- [RT01] John G. Ratcliffe and Steven T. Tschantz. On the Davis hyperbolic 4-manifold. *Topology and its Applications*, 111(3):327–342, 2001. doi:10.1016/S0166-8641(99)00221-7.
- [ula26] ulamai. UnsolvedMath. Hugging Face dataset, <https://huggingface.co/datasets/ulamai/UnsolvedMath>, 2026. Version 1.6.0; record KP-4.124.
- [Vin85] È. B. Vinberg. Hyperbolic reflection groups. *Russian Mathematical Surveys*, 40(1):31–75, 1985. doi:10.1070/RM1985v040n01ABEH003527.