Enumeration of Viral Capsid Assembly Pathways: Tree Orbits Under Permutation Group Action

Miklós Bóna*
Meera Sitharam†
Andrew Vince‡

February 12, 2010

Abstract

This paper uses combinatorics and group theory to answer questions about the assembly of icosahedral viral shells. Although the geometric structure of the capsid (shell) is fairly well understood in terms of its constituent subunits, the assembly process is not. For the purpose of this paper, the capsid is modeled by a polyhedron that is a subdivision of the icosahedron and whose facets represent the monomers. The assembly process is modeled by a rooted tree, the leaves representing the facets of the polyhedron, the root representing the assembled polyhedron, and the internal vertices representing intermediate stages of assembly (subsets of facets). Besides its virological motivation, the enumeration of orbits of trees under the action of a finite group is of independent mathematical interest. If G is a finite group acting on a finite set X, then there is a natural induced action of G on the set $\mathcal{T}_X$ of trees whose leaves are bijectively labeled by the elements of X. If G acts simply on X, then $|X| := |X_n| = n \cdot |G|$, where n is the number of G-orbits in X. The basic combinatorial results in this paper are (1) a formula for the number of orbits of each size in the action of G on $\mathcal{T}_{X_n}$, for every n, and (2) a simple algorithm to find the stabilizer of a tree $\tau \in \mathcal{T}_X$ in G that runs in linear time and does not need memory in addition to its input tree. These results help to clarify the effect of symmetry on the probability of the occurance of a particular assembly pathway for a icosahedral viral capsid, and more generally for any finite, symmetric macromolecular assembly.

2000 Mathematics Subject Classification: Primary 05C05, 05A15, 20B25, 92C50.

Key words: tree enumeration, generating function, group action, viral capsid assembly.

*supported in part by NSF grant DMS0714912 Department of Mathematics, University of Florida, Gainesville, FL 32611. Email: bona@math.ufl.edu

†corresponding author; supported in part by NSF grant DMSO714912; CISE dept., University of Florida, Gainesville, FL 32611. Email: sitharam@cise.ufl.edu

‡Department of Mathematics, University of Florida, Gainesville, FL 32611. Email: avince@ufl.edu.

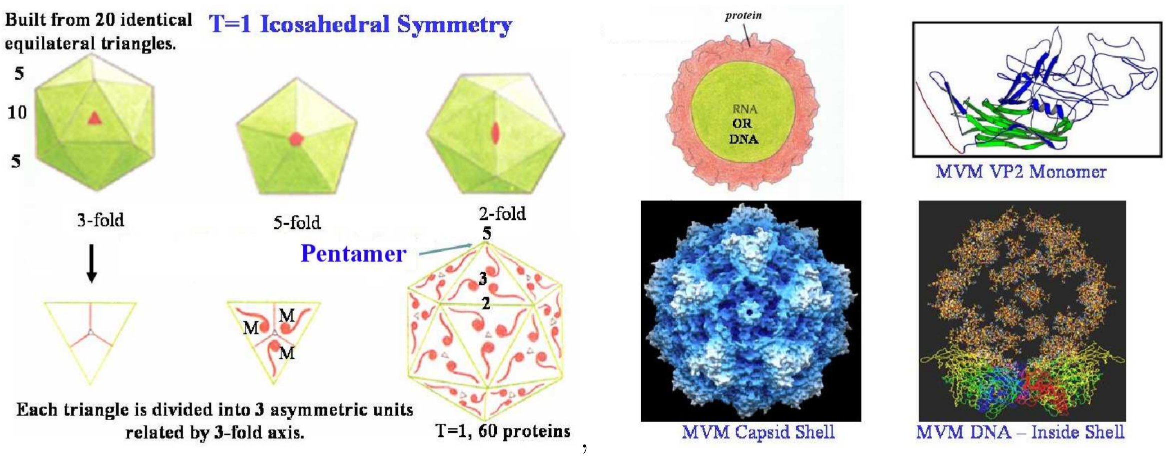


Figure 1: (Left) Basic Viral Structure. (Right) Minute virus of Mice X-ray and monomer structure (courtesy [1]).

1 Introduction

Viral shells, called *capsids*, encapsulate and protect the fragile nucleic acid genome from physical, chemical, and enzymatic damage. Francis Crick and James Watson (1956) were the first to suggest that viral shells are composed of numerous identical protein subunits called *monomers*. For many viruses, these monomers are arranged in either a helical or an icosahedral structure. We are interested in those shells that possess icosahedral symmetry. For many virus families the structure of the capsid is well understood and substantiated by crystallographic images. However, the viral capsid assembly process - like many other spontaneous macromolecular assembly processes - is not well understood, even for the simplest viral shells with the smallest number of monomers. In many cases, the capsid self-assembles spontaneously, rapidly and quite accurately in the host cell, with or without enclosing the internal genomic material, and without the use of chaperone, scaffolding or other helper proteins. This is the type of assembly we address here.

1.1 Viral Structure Basics

Icosahedral viral shells can be classified based on their polyhedral structure, facets corresponding to the monomers. Note that, while an icosahedron has 20 *faces*, an icosahedrally symmetric polyhedron can have several *facets* associated with each of these faces. The classical "quasi-equivalence theory" of Caspar and Klug [6] explains the structure of the polyhedral shell in the case where the monomers have very similar neighborhoods. According to the theory, the number of facets in a viral polyhedron is $60T$, where the T-*number* is of the form $h^2 + hk + k^2$. Here h and k are non-negative integers. In the case of a $T = 1$ polyhedron, there are 3 facets associated with each face of the icosahedron. See Figure 1 for basic icosahedral structure and X-ray structure of a $T = 1$ virus. Although the results of this paper apply to all icosahedrally symmetric polyhedra, or even polyhedra with other symmetry groups, we often use $T = 1$ polyhedra for the purpose of illustration.

1.2 Viral assembly models

Many mathematical models of viral shell assembly have been proposed and studied including [2, 4, 13, 15, 16, 17, 26, 27, 28]. Here we use the *GT (geometry and tensegrity)* model of [19]. In the GT model, information about the construction (or decomposition) of the viral shell is represented by an *assembly tree* (formal definition in Section 3). Informally, vertices of the tree represent subassemblies that do not disintegrate during the course of the assembly process. In an assembly tree, these subassemblies are partially ordered by containment, with the root representing the complete assembled structure, and the leaves representing the monomers. That is, we only consider trees that represent successful assemblies. See Figure 2 for our labeling of a $T = 1$ polyhedron, and Figures 3, 4, 5 for examples of assembly trees. In these Figures, for ease of viewing, the leaves of the original assembly trees (corresponding to monomers in the viral shell) have been deleted. The resulting leaves, that is, the parents of the original leaves, represent pentamers in the viral shell.

Besides being intuitive and analyzable, it was shown in [19] that the GT model's rough predictions fit experimental and biophysical observations of known viral assemblies, specifically those of the viruses MVM (Minute Virus of Mice), MSV (Maize Streak Virus) and AAV4 (Human Adeno Associate Virus).

The GT model was developed to answer questions that concern *only* the influence of two factors on the probability of classes of assembly trees. The icosahedral group acts simply on the set of facets of any icosahedrally symmetric polyhedron (monomers of the shell). A group G acts *simply* on a set X if, for any $x \in X$, if $g(x) = x$, then g must be the identity. Informally, the classes of assembly trees that the GT model considers are the orbits of assembly trees under the action of the icosahedral group, and are refered to as *assembly pathways* (see also Section 1.3 and formal definition in Section 3). See Figures 3, 4 for examples of distinct assembly trees that belong to the same orbit and hence are representatives of the same assembly pathway. We call these two factors the *geometric stability factor* and the *symmetry factor*. The higher these quantities, the higher the probability.

1.3 Factors influencing assembly

There are two main factors that influence the probability of an assembly pathway, the *geometric stability factor* and the *symmetry factor*. These are informally described next. Most of the results in [19] concern the geometric stability factor. The capsid is assembled in stages, the subassembly at each stage represented by a vertex of the assembly tree. Although it is an open problem to predict assembly intermediates, even for known structures, each subassembly must satisfy certain geometric constraints within or between monomers. These constraints are equalities and inequalities that involve distances,

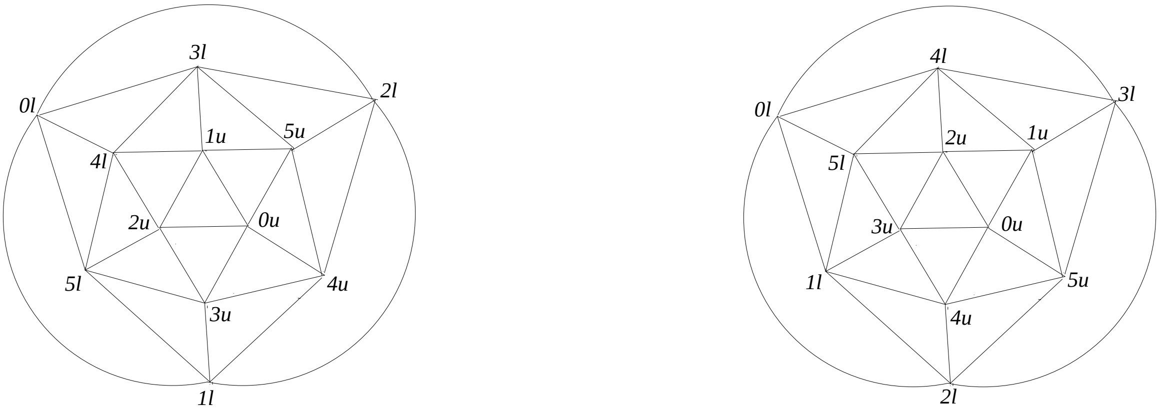


Figure 2: (Left) Vertex labeling of an icosahedron used in this paper. NOTE: each vertex label represents a pentamer in the corresponding $T = 1$ polyhedron, i.e, the collection of 5 monomers or facets surrounding the icosahedral vertex, as shown in the lower right corner of the (Left) of Figure 1 (Right) result of action of the permutation $(0u)(0l)(1u\ 2u\ 3u\ 4u\ 5u)(1l\ 2l\ 3l\ 4l\ 5l)$.

Figure 3: (Left) For ease of drawing, we have omitted the leaves in our $T = 1$ assembly trees. The leaves in our drawing represent the $T = 1$ polyhedron's pentamers (labeled by vertices of underlying icosahedron) as in Figure 2, and arrows representing action of permutation in Figure 2; (Right) result of action yields another tree in its orbit, i.e, another representative of the same assembly pathway.

Figure 4: (Left) another assembly tree with leaf labels as in Figure 4 (Right) result of action of permutation in Figure 2 yields another tree in its orbit, i.e, another representative of the same assembly pathway.

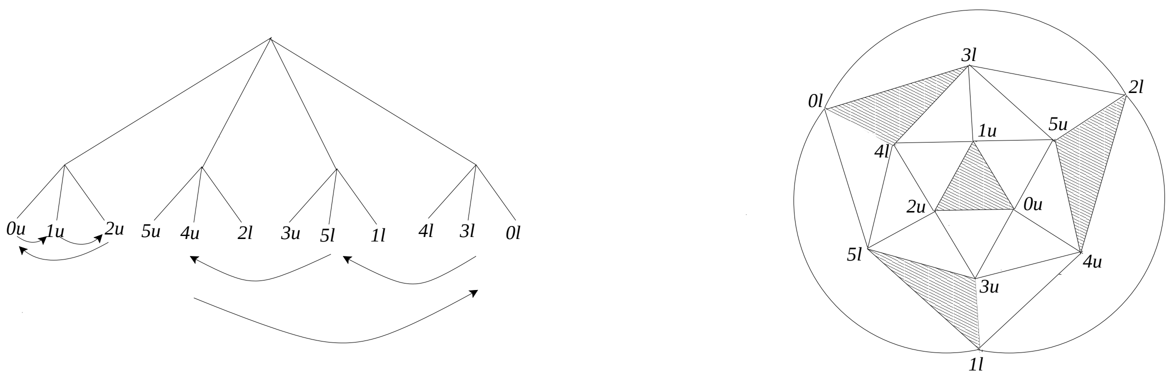


Figure 5: (Left) an assembly tree with leaf labels as in Figure 4 that is fixed by the permutation represented by the arrows, namely $(0u\ 1u\ 2u)(5u\ 4l\ 3u)(3l\ 5l\ 4u)(0l\ 1\ 2)$; (Right) shaded faces of the icosahedron represent the internal nodes of the tree, one level up from the leaves: one of these faces is fixed by the permutation and the other three map on to each other cyclically.

angles, and forces, obtained either from X-ray or from cryo-electro-microscopic information on the complete viral shell. In addition, each subassembly must be sufficiently constrained so as to be rigid. Some assembly trees fail to satisfy the rigidity condition and can never occur (probability 0). Such assembly trees are called *geometrically invalid.* A valid assembly tree can be assigned a non-zero probability according to how difficult it is to find a solution to the constraints on each subassembly. This probability - called the *geometric stability factor* - is computed in terms of the algebraic complexity of the configuration space determined by the constraints [19]. How the geometric stability factor is correlated with biochemical stability and entropy is also explained in [19]. This factor is computed by analyzing the corresponding subsystems of the given viral geometric constraint system. As done in [19], it is mathematically justified to [12] treat the rigidity aspect of the geometric stability factor as a generic, graph-theoretic property.

Based on the geometric stability factor, some assembly trees can be shown to never occur (have probability zero) since the subassemblies present in them are unstable (their geometric stability factor is zero). Such assembly trees are *geometrically invalid.*

The *symmetry factor* is defined as follows. As mentioned earlier, the icosahedral group acts on the set of assembly trees for a particular viral polyhedron P. The facets of P (representing viral monomers) are the leaves of the tree. However, Figures 3, 4, 5 all abbreviate the trees so that the leaves represent the 5 facets surrounding a vertex of the icosahedron, i.e., a pentamer. The non-leaf vertices are collections of facets (representing subassemblies of monomers. This action is defined precisely in Section 3. Each orbit under this action is called an *assembly pathway* and corresponds intuitively to a distinct type of assembly process for the viral capsid. The *cardinality* of an assembly pathway is the number of assembly trees in the orbit. The *symmetry factor* of an assembly pathway is its cardinality divided by the total number of assembly trees.

In [20], the authors observed the following attractive feature of the GT model of assembly. The two separate factors - geometric stability and symmetry - that influence the probability of the occurrence of a particular assembly pathway can be analyzed largely independently as follows. An obvious, but crucial, observation made in [20] is that the geometric stability factor and is an invariants of the assembly pathway. That is, it is the same for any assembly tree in the same orbit under the action of the icosahedral group.

We assume that each assembly tree is equally likely to occur. Thus the probability of the occurrence of a pathway is roughly proportional to some combination of the symmetric factor and the geometric stability factor. Additionally, the ratio of the orbit sizes of two trees τ_1 and τ_2 could serve as a rough estimate of the the ratio of the probabilities of the corresponding assembly pathways - *provided* that the former ratio is not cancelled out or reversed by the ratio of the geometric stability factor of τ_1 and τ_2. The paper [3] provides a proof that this kind of cancelling out would not generally take place, at least for valid pathways, for the following reasons. First, it is shown in [3] that the symmetry factor of a pathway increases with the depth of its representative tree τ. More precisely, it was proved that the size of the orbit of τ is bounded below by the depth of τ. Based on computational experiments, we conjecture that the orbit size in fact increases with the depth of τ, although this has not yet been proven, see Section 8. Moreover, it is known from [19] that the geometric stability generally increases with the depth of the tree (and this correlates with biophysical observations). Therefore, if the depth of τ_1 is greater than the depth of τ_2, then *both* the symmetric factor and the geometric stability factor of τ_1 are expected to be generally larger than the corresponding factors of τ_2.

2 Contributions and Related Work

Based on the observations in the last section, the paper [3] posed problems intended to isolate and clarify the influence of the symmetry factor on the probability of the occurrence of a given assembly pathway. Two specific problems were the following.

(i) Enumerate the valid assembly pathways of an icosahedrally symmetric polyhedron. More precisely, the problem is to determine the number of such assembly pathways of each cardinality.

(ii) Characterize and algorithmically recognize the set of assembly trees fixed by a given subgroup of the icosahedral group. This characterization problem is a step toward the solution of the enumeration problem (i). Algorithmic recognition of the group elements that fix a given assembly tree (the stabilizer of the tree in the given group) directly determines whether the given assembly tree has a given orbit size.

In this paper, we answer the above questions for general assembly trees, that is, *we drop the condition of validity.* Furthermore, in this paper, the geometric stability factor will be ignored, and thus “probability” will refer to the symmetry factor only. We expect that the techniques developed in this paper will help in answering questions (i) and (ii) in the presence of the validity condition as well. In addition, these techniques generalize to assembly trees for other symmetric polyhedra. Hence their application extends beyond viral capsids to other finite, symmetric, macromolecular assemblies.

For Problem (i), we develop an enumeration method using generating functions and Möbius inversion. For the algorithm in Problem (ii), we provide a simple permutation group algorithm and an associated data structure. The results of this paper work, not just for the icosahedral group, but also for any finite group G acting simply on a set X. Indeed, if G is a finite group acting on a set X, then there is a natural induced action of G on the set $\mathcal{T}_X$ of *assembly trees.* These are formally defined as rooted trees τ whose non-leaf vertices have at least two children and whose leaves are bijectively labeled by X.

Concerning Problem (i), Pólya theory gives a convenient method for counting orbits under a permutation group action. However, because of the intractability of computing the cycle index in our situation, we were not able to apply Pólya theory to Problem (i). Specifically, in order to use Pólya theory to even compute the size of the G-orbit of an assembly tree τ, it is not even clear how to define an appropriate group G_τ for which cycle indices will yield the sizes of these orbits; the cycle indices will be difficult to describe since they will depend on both G and the automorphism group of the specific tree τ. Similarly, the methods used in [14] for enumerating labeled graphs under a group action (as opposed to rooted labeled trees), do not apply in any straightforward manner. If one is only interested in asymptotic estimates for Problem (i), such as in viruses with large T-numbers, a possible avenue is to use the results of [7, 25] that estimate the asymptotic probabilities of logic properties on finite structures, especially trees. There are significant roadblocks, however, to applying these results to our problem. For example, in order to directly employ such a result, it would *at least* be necessary to show that the tree property of not being fixed by any group element (other than the identity) in a given permutation group is a monadic second order property. In addition it would be necessary to extend the result in [7] from the general class of trees in the denominator to the special class of *assembly* trees in the denominator. Finally, there is a rich literature on the enumeration of construction sequences of symmetric polyhedra and their underlying graphs [5, 9, 10]. Whereas these studies focus on enumerating construction sequences of *different* polyhedra with a given number of facets, our goal - of counting and characterizing assembly tree orbits - is geared towards enumerating construction sequences of a *single* polyhedron for any given number of facets.

Our method for solving Problem (i), on the other hand, finds an explicit formula (Theorem 3) for the number of assembly pathways of each possible cardinality. If G is a finite group acting simply on a finite set X, then

$$|X| = n \cdot |G|, \tag{1}$$

where n is the number of G-orbits in X. If $|X_n| = n \cdot |G|$, consider the natural induced action of G on the set $\mathcal{T}_{X_n}$ of assembly trees whose leaves are bijectively labeled by the elements of X_n. If G acts simply on X_n, then Our method finds the number of orbits of each possible size in the action of G on the set $\mathcal{T}_{X_n}$ of assembly trees, for every n. This leads to a formula for the probability of occurrence of any given assembly pathway (Corollary 5). In order to apply this formula, it is necessary to know the number of assembly trees fixed by each given subgroup of G. A generating function formula for this number of fixed assembly trees is given in Theorem 16 of Section 6. For the proof of Theorem 16, it is necessary to characterize the set of such fixed assembly trees. This is done in Theorem 9 of Section 5.

Concerning Problem (ii), the subject of permutation group algorithms is well developed (see for example [18]). The structure of the automorphism group of a rooted, labeled trees has been studied [11, 24], and algorithms for tree isomorphism and automorphism are well known [8, 22]. However, we have not encountered an algorithm in the literature for deciding whether a given permutation group element fixes a given rooted, labeled tree - thus determining the stabilizer of that tree in the given group. In Section 4 of this paper, we provide a simple and intuitive algorithm that is easy to implement, runs in linear time and operates in place on the input, without the use of extra scratch memory.

2.1 Organization

The remainder of the paper is organized as follows. Section 3 begins with preliminaries on group action on assembly trees. The section concludes with a formula for the number of assembly pathways of each cardinality under the action of a group G. This formula is in terms of the numbers t_H of assembly trees fixed by a subgroup H of G. Section 5 uses block systems to characterize the structure of assembly trees that are fixed by a given group H. This leads, in Section 6, to a generating function to enumerate them, i.e. to compute the numbers t_H. Section 4 gives an algorithm for finding the stabilizer of an assembly tree in a given group or, equivalently, for recognizing if an assembly tree has a given orbit size. In Section 7, the results are applied to the case of the icosahedral group and to $T = 1$ viral shell assembly pathways. Finally, Section 8 concludes with open problems. Our exposition contains several examples for illustrating concepts and results; in particular, one running example, introduced in Example 1, is carried throughout the paper.

3 Preliminaries on Assembly Pathways

All groups and graphs in this paper are assumed to be finite. A *rooted tree* is a tree with a designated vertex, called the *root*. We will use standard terminology such as *adjacent*, *child*, *parent*, *descendent*, *ancestor*, *leaf*, *subtree rooted at*, *root of the subtree* and so on. Two rooted trees τ and τ' are said to be *isomorphic* if there is a bijection - the *isomorphism* - f between the vertices of τ and τ' that preserves adjacency and the root. That is, the following hold:

- (u, v) is an edge in τ if and only if $(f(u), f(v))$ is an edge in τ',
- $f(r) = r'$, where r and r' are the roots of τ and τ' respectively.

In this case, we say $\tau \approx \tau'$ and also $f(\tau) = \tau'$.

A rooted tree τ for which each internal vertex has at least two children and whose leaves are bijectively labeled with elements of a set X is called an *assembly tree for* X. The 26 assembly trees with four leaves, labeled in the set $X = \{1, 2, 3, 4\}$ are shown in Figure 6. To each vertex v of τ, associate the set of all leaves of X that are descendents of v. The vertex v is labeled with this subset of X, and we often *make no distinction between the vertex and its label.* Note that the label of a vertex is the union of the labels of its children. So each assembly tree τ has a labeling $L : \tau \to 2^X$, taking each vertex to a nonempty subset of X. Call two labeled, rooted trees *identical* if there is an (unlabeled) isomorphism f between them such that, in addition,

- $L(f(u)) = L(u)$ for all vertices u.

In other words, two labeled trees are identical if there is a bijection between their vertex sets that preserves adjacency, the root, and the labels. In this case we write $\tau = \tau'$.

Let G be a group acting on a set X, and let τ be an assembly tree in $\mathcal{T}_X$. The action of G on X induces a natural action of G on the power set of X and thereby on the set of vertices (vertex labels) of τ. If $g \in G$, then define the tree $g(\tau)$ as the unique assembly tree whose set of vertex labels (including the labels of internal vertices) is $\{g(v) : v \in \tau\}$. Thus we have an induced action of G on $\mathcal{T}_X$. See Figures 3, 4 for an illustration. The tree $g(\tau)$ is clearly isomorphic to τ via g. Therefore, each orbit of this action of G on $\mathcal{T}_X$ consists of isomorphic trees. Any such orbit is called an *assembly pathway for* (G, X).

Example 1 *Klein 4-group acting on* $\mathcal{T}_4$.

Consider the Klein 4-group $G = \mathbb{Z}_2 \oplus \mathbb{Z}_2$ acting on the set $X = \{1, 2, 3, 4\}$. Writing G as a group of permutations in cycle notation, this action is

$$G = \{(1)(2)(3)(4),\ (1\ 2)(3\ 4),\ (1\ 3)(2\ 4),\ (1\ 4)(2\ 3)\}.$$

For this example there are exactly 11 assembly pathways, which are indicated in Figure 6 by boxes around the orbits. There are four assembly pathways of size one, i.e., with one assembly tree in the orbit, three assembly pathways of size two, and four assembly pathways of size four.

An assembly tree τ is said to be *fixed* by an element $g \in G$ if $g(\tau) = \tau$. See Figure 5 for an illustration. For any subgroup H of G, let $t_X(H)$ denote the number of trees in $\mathcal{T}_X$ that are fixed by a subgroup H of G. Furthermore, let $\overline{t}(H) := \overline{t_X}(H)$ denote the number of trees in $\mathcal{T}_X$ that are fixed by all elements of H and by no other elements of G. In other words,

$$\overline{t}_X(H) = |\{\tau \in \mathcal{T}_X \mid stab_G(\tau) = H\}|. \tag{2}$$

Here $stab_G(\tau) := \{g \in G \mid g(\tau) = \tau\}$ is called the *stabilizer* of τ in G. In other words, $stab_G(\tau)$ is the set of all elements in G that fix τ. It is easy to prove that $stab_G(\tau)$ is a subgroup of G.

We will see in Section 6 that it is more natural to find $t_X(H)$, than $\overline{t}_X(H)$. The set enumerated by $t_X(H)$ may include trees that are fixed by larger subgroups H' such that $H \leq H' \leq G$. As the following theorem shows, the desired quantities $\overline{t}_X(H)$ can then be computed from the numbers $t_X(H)$ using Möbius inversion on the lattice of subgroups of G.

Theorem 2 *Let G be a group acting on a set X. If H is a subgroup of G, then*

$$\overline{t_X}(H) = \sum_{H \leq K \leq G} \mu(H, K)\, t_X(K),$$

where μ is the Möbius function for the lattice of subgroups of G.

Proof: Clearly $t_X(H) = \sum_{H \leq K \leq G} \overline{t_X}(K)$. The theorem follows from the standard Möbius inversion formula [23, page 333]. $\square$

The index of a subgroup H in G is the number of left (equivalently, right), cosets of H in G, and is denoted by $(G : H)$. By Lagrange's Theorem, this index equals $|G|/|H|$.

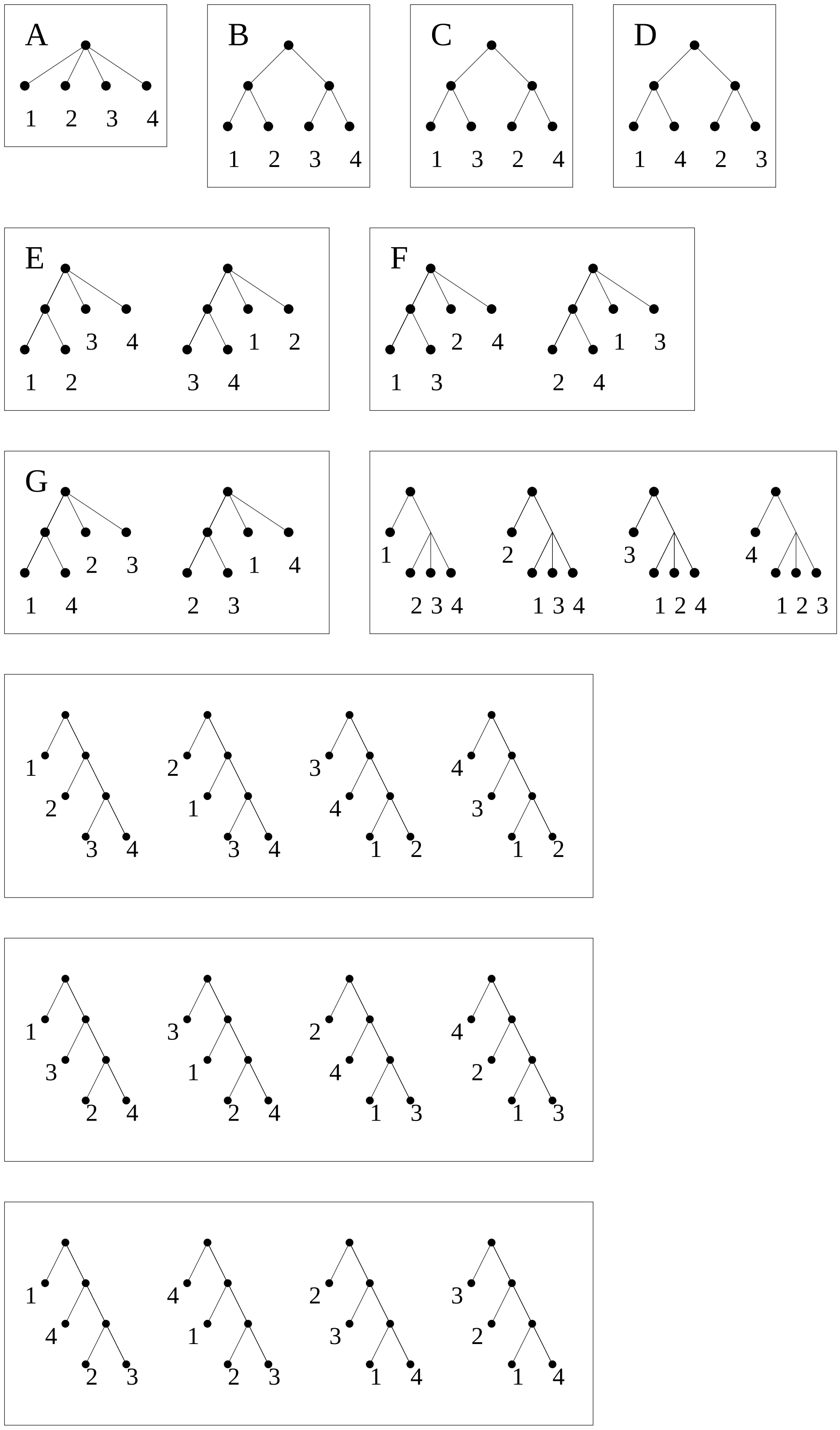


Figure 6: Klein 4-group acting on $\mathcal{T}_4$.

Theorem 3 *The number of trees in any assembly pathway for (G, X) divides $|G|$. If m divides $|G|$, then the number $N(m)$ of assembly pathways of cardinality m is*

$$N(m) = \frac{1}{m} \sum_{H \leq G \,:\, (G:H)=m} \bar{t}(H).$$

Proof: It is a standard consequence of Lagrange's Theorem that, for any assembly tree τ, the equality

$$|G| = |O(\tau)| \cdot |stab(\tau)|$$

holds, where $O(\tau)$ is the orbit of τ. This immediately implies the first statement of the theorem.

Let

$$\delta(\tau) = \begin{cases} 1 & \text{if } (G : stab(\tau)) = m \\ 0 & \text{otherwise} \end{cases} = \begin{cases} 1 & \text{if } |O(\tau)| = m \\ 0 & \text{otherwise.} \end{cases}$$

Now count, in two ways, the number of pairs (H, τ) where τ is an assembly tree, $H \leq G$ is the stabilizer of τ, and $(G : H) = m$:

$$\sum_{H \leq G \,:\, (G:H)=m} \bar{t}(H) = \sum_{\tau \in \mathcal{T}_X} \delta(\tau) = m\, N(m).$$

Indeed, to justify the first equality, note that for a fixed subgroup H that has index m in G, exactly $\bar{t}(H)$ trees τ will satisfy $\delta(\tau) = 1$. To justify the second equality, note that $\delta(\tau) = 1$ if and only if τ is one of m elements of an m-element pathway, and there are $N(m)$ such pathways. $\square$

Example 4 *Klein 4-group acting on $\mathcal{T}_4$ (continued).*

Theorem 3, applied to our previous example of $\mathbb{Z}_2 \oplus \mathbb{Z}_2$ acting simply on $\{1, 2, 3, 4\}$, states that the size of an assembly pathway must be 1, 2 or 4, since it must be a divisor of $4 = |\mathbb{Z}_2 \oplus \mathbb{Z}_2|$. To find the number of pathways of each size, note that G has three subgroups of order 2, namely

$$\begin{aligned} K_1 &= \{\, (1)(2)(3)(4), (1\ 2)(3\ 4) \,\}, \\ K_2 &= \{\, (1)(2)(3)(4), (1\ 3)(2\ 4) \,\}, \\ K_3 &= \{\, (1)(2)(3)(4), (1\ 4)(2\ 3) \,\}, \end{aligned}$$

and that

$$\begin{aligned} &\bar{t}(G) = 4, \\ &\bar{t}(K_1) = \bar{t}(K_2) = \bar{t}(K_3) = 2, \\ &\bar{t}(K_0) = 16, \end{aligned}$$

where K_0 denotes the trivial subgroup of order 1. The assembly trees in $\mathcal{T}_X$ that are fixed by all elements of G are shown in Figure 6, A, B, C, D. For $i = 1, 2, 3$, those assembly trees in $\mathcal{T}_X$ that are fixed by all elements of K_i and by no other elements of

G are are shown in Figure 6, E, F, G, respectively. The remaining 16 assembly trees in Figure 6 are fixed by no elements of G except the identity. Therefore, according to Theorem 3, the number of pathways of size 1, 2 and 4 are, respectively,

$$\begin{aligned} \bar{t}(G) &= 4, \\ \frac{1}{2}\left(\bar{t}(K_1)+\bar{t}(K_2)+\bar{t}(K_3)\right) = \frac{1}{2}(2+2+2) &= 3, \\ \frac{1}{4}\,\bar{t}(K_0) &= 4. \end{aligned}$$

A general formula for $\bar{t}(H)$ is the subject of Sections 5 and 6.

The set $\mathcal{T}_X$ of assembly trees can be made the sample space of a probability space $(\mathcal{T}_X, p)$ by assuming that each assembly tree $\tau \in \mathcal{T}_X$ is equally likely, i.e., $p(\tau) = 1/|\mathcal{T}_X|$. This assumption is in accordance with the fact that we disregard the geometric stability factor that would assign different probabilities to different assembly trees. With this assumption, it is clear that if O is an assembly pathway, then

$$p(O) = \frac{|O|}{|\mathcal{T}_X|}.$$

The following result follows immediately from Theorem 3.

Corollary 5 *If G acts on the set X and m divides $|G|$, then, with notation as in Theorem 3, there are exactly $N(m)$ assembly pathways with probability $\frac{m}{|\mathcal{T}_X|}$, and no other values can occur as the probability of an assembly pathway.*

Example 6 *Klein 4-group acting on $\mathcal{T}_4$ (continued).*

Again, for our example of $\mathbb{Z}_2 \oplus \mathbb{Z}_2$ acting simply on $\{1, 2, 3, 4\}$, application of Corollary 5 gives

$$\begin{array}{rll} 4 & \text{pathways with probability} & \dfrac{1}{26}, \\ 3 & \text{pathways with probability} & \dfrac{1}{13}, \\ 4 & \text{pathways with probability} & \dfrac{2}{13}. \end{array}$$

4 Algorithm for determining the stabilizer of an assembly tree in a given finite group

The algorithm in this section takes as input a finite permutation group G acting on a finite set X and an assembly tree $\tau \in \mathcal{T}_X$, and finds the stabilizer $stab_G(\tau)$. The idea behind the algorithm is encapsulated by the following proposition, whose proof follows directly from definitions given in Section 2. As defined in Section 3, the action of the permutation group G on X induces a natural action of G on $\mathcal{T}_X$.

Proposition 7 *Let the finite permutation group G act on a finite set X, and let $\tau \in \mathcal{T}_X$.*

1. *Let R be any set of elements of G that fix τ, and let C be any set of elements of G that do not fix τ. Then $\langle R \rangle$, the group generated by the elements of R, is a subgroup of $stab_G(\tau)$ and $\bigcup_{c \in C} c\langle R \rangle$, the union of the left cosets of $\langle R \rangle$ given by C, has an empty intersection with $stab_G(\tau)$.*

2. *An element $g \in G$ fixes τ if and only if for every vertex $v \in \tau$ with children $c_1(v), \ldots, c_k(v)$, the vertices $g(c_1(v)), \ldots, g(c_k(v))$ have a common parent in τ, and this parent is $g(v)$.*

4.1 Input and Data Structures

In this subsection, we give the detailed setup for our stabilizer-finding algorithm. The input of this algorithm is the set of elements g of the finite permutation group G acting on the finite set X and an assembly tree $\tau \in \mathcal{T}_X$.

We use a tree data structure, where each vertex has a child pointer to each of its children and a parent pointer to its parent. The root (and the tree τ itself) can be accessed by the root pointer. Furthermore, each permutation $g \in G$ on X is input as a set of g-pointers on the leaves. That is, a leaf labeled u has a g-pointer to the leaf labeled $g(u)$. However, the labels are not explicitly stored except at the leaves. This is a common data structure used in permutation group algorithms [18].

4.2 Algorithms

The first algorithm computes $stab_G(\tau)$, and it uses the second and third algorithms for determining whether a permutation g fixes τ. The correctness of the first algorithm follows directly from Proposition 7 (1), assuming the correctness of the latter algorithms. The last algorithm is recursive and operates in place with no extra scratch space. For each vertex v, working bottom up, it efficiently checks whether the image $g(v)$ is in τ. The correctness follows directly from Proposition 7 (2).

Algorithm Stabilizer
Input: assembly-tree $\tau \in \mathcal{T}_X$; permutation group, G
Output: generating set R_τ s.t. the group $\langle R_\tau \rangle$ generated by R_τ is exactly $stab_G(\tau)$.

$R := \{id\}$ `(currently known partial generating set of` $stab_G(\tau)$`)`
$C_R := \emptyset$ `(distinct left coset representatives of` $\langle R \rangle$
    `that are currently known to not fix` τ`)`
$U := G$ `(currently undecided elements of` G`)`
 `do until` $U = \emptyset$
 `let` $g \in U$
   `if` **Fixes**(g, τ)
    `then` $R := R \cup \{g\}$`; retain in` C_R `at most one`

```
            representative from any left coset of ⟨R⟩
    else C_R := C_R ∪ {g};
  U := (U \ (⟨R⟩ ⋃_{c∈C_R} c⟨R⟩)
  fi
 od
return R_τ := R.
```

Algorithm Fixes
Input: assembly-tree $\tau \in \mathcal{T}_X$; permutation g acting on X,
Output: "true" if g fixes τ; "false" otherwise.

```
if LocateImage(g, τ, root(τ)) = root(τ)
then return true
else return false.
```

Algorithm LocateImage
Input: assembly-tree $\tau \in \mathcal{T}_X$; permutation g acting on X; child pointer to a vertex $v \in \tau$ (root pointer if v is the root of τ).
Output: a parent pointer to the vertex $g(v) \in \tau$ - if it exists - such that g is the isomorphism mapping the subtree of τ rooted at v to the subtree of τ rooted at $g(v)$; if such a vertex $g(v)$ does not exist in τ, returns null.

```
if v is a leaf of τ (null child pointer)
then return g(v) (follow g-pointer)
else
 let c_1, ..., c_k be the children of v;
 if parent(LocateImage(g, τ, c_1)) =
  parent(LocateImage(g, τ, c_2)) = ...
  parent(LocateImage(g, τ, c_k)) =: w
 then return w
 else return null.
```

4.3 Complexity

Algorithm LocateImage follows each pointer (child, g, parent) exactly once as is illustrated by the example shown in Figures 6-9, and does only constant time operations between pointer accesses. Hence it takes at most $O(|X|)$ time. It operates in place and does not require any extra scratch space. **Algorithm Stabilizer**, in the worst case, can be a brute force algorithm that simply runs through all the elements of G instead of maintaining efficient representations of R, C_R and U. In this case, it takes no more than $O(|G||X|)$ time.

However, readers familiar with Sim's method for representing permutation groups [18] using so-called strong generating sets and Cayley graphs may appreciate the follow-

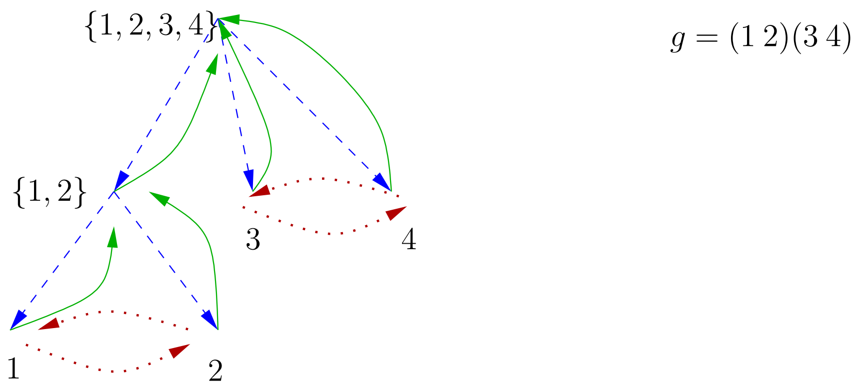


Figure 7: **LocateImage** is called on the root of the assembly tree τ shown on the left, for the permutation g shown on right. This permutation g fixes τ. The data structure representing the tree consists of the child (blue, dashed) and parent (green, solid) pointers and the data structure representing g - via its action on the leaf-label set X - consists of g-pointers (red, dotted).

ing remarks. Instead of specifying the input of our problem as we have done, we may assume that a Cayley graph is input, which uses a strong generating set of G. With this input representation, the time complexity of our algorithms can be significantly further optimized, the level of optimization depending on properties of the group G.

4.4 Example

The two examples shown in Figures 6-9 illustrate the algorithms **LocateImage** and **Fixes**. Figure 7 for the first example shows the assembly tree τ and the associated data structures, as well as the group element g. Figure 8 shows a run of **LocateImage** applied to τ, g at $root(\tau)$. The algorithm establishes that the given permutation g fixes the given assembly tree τ, whereby **Fixes** returns 'true.' The second example, in Figure 9, uses the same assembly tree τ, but a different permutation g. The run of **LocateImage** in Figure 10 is unsuccessful, whereby **Fixes** returns 'false.'

5 Block systems and Fixed Assembly Trees

The formulas in Section 3 for the number of orbits of each size and for the orbit sizes or pathway probabilities (Theorem 3 and Corollary 5) depend on the number of assembly trees fixed by a group. A formula for the number of such fixed trees is the subject of this and the next section.

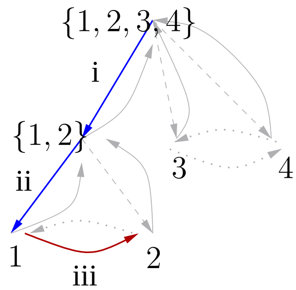
Recall that an assembly tree τ is fixed by a group G acting on X if $g(\tau) = \tau$ for all $g \in G$. Two main results of this section (Corollary 12 and Procedure 13) provide a recursive procedure for constructing all trees in $\mathcal{T}_X$ that are fixed by G. This leads, in the next section, to a generating function for the number of such fixed trees. The results

$g(1) = 2$

LocateImage(1)

LocateImage($\{1, 2\}$)

LocateImage(root)

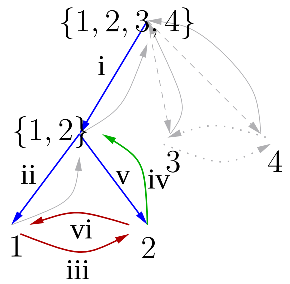


$g(2) = 1$

$g(1) = 2$; parent($g(1) = \{1, 2\}$)

LocateImage($\{1, 2\}$)

LocateImage(root)



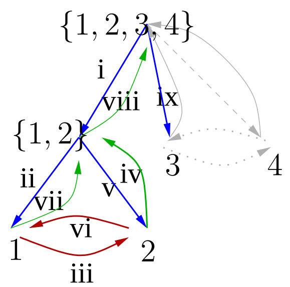
LocateImage(3)

$g(\{1, 2\}) = \{1, 2\}$

$g(2) = 1$; parent($g(2) = \{1, 2\}$)

$g(1) = 2$; parent($g(1) = \{1, 2\}$)

LocateImage(root)



g(root)=root; Fixes returns True

$g(4) = 3$; parent($g(4)$ = root)

$g(3) = 4$; parent($g(3)$ = root)

$g(\{1, 2\}) = \{1, 2\}$; parent($g(\{1,2\})$)=root

$g(2) = 1$; parent($g(2) = \{1, 2\}$)

$g(1) = 2$; parent($g(1) = \{1, 2\}$)

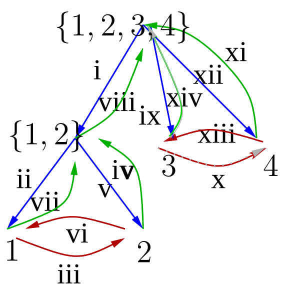
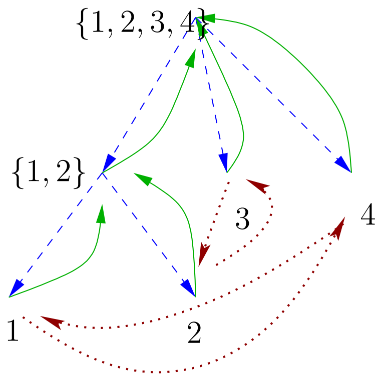
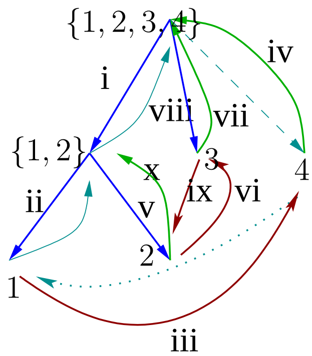


Figure 8: A successful run of **LocateImage**, where, for all vertices v in τ, the image $g(v)$ is established to be in the assembly-tree τ shown in Figure 7. On the right are pointers traversed so far, in traversed order. On the left is the current recursion stack of **LocateImage** calls (first call at the bottom), together with those vertices v ($\in X$ or $\subseteq X$) for which $g(v)$ has been established to be in τ, showing that g is an isomorphism between the two subtrees of τ rooted at v and at $g(v)$, respectively.



$g = (1\,4)(2\,3)$

Figure 9: **LocateImage** is called on the root of the same tree τ as in Figure 7, but for a different permutation g shown on right. In this case g does *not* fix τ.



g(root)=null; Fixes returns False

parent($g(3)$)$\neq$ parent($g(\{1,2\})$)

$g(3) = 2$; parent($g(3) = \{1,2\}$)

$g(\{1,2\}) = \{1,2\}$; parent($g(\{1,2\})$)=root

$g(2) = 3$; parent($g(2)$)=root)

$g(1) = 4$; parent($g(1)$)=root)

Figure 10: An unsuccessful run of **LocateImage**. Since $\{1,2\}$, 3 and 4 are the children of the root, when **LocateImage**(root) is called, it checks if their images under g have the same parent. So recursive calls are to **LocateImage**($\{1,2\}$), to **LocateImage**(3), and to **LocateImage**(4). **LocateImage**($\{1,2\}$) returns the root of τ as a candidate for $g(\{1,2\})$. But **LocateImage**(3) returns $\{1,2\}$ because the parent of $g(3) = 2$ is $\{1,2\}$. Hence **LocateImage**(root) returns null and **Fixes** returns 'false.'

in this section depend on a characterization (Theorem 9) of block systems arising from a group acting on a set.

For a group G acting on set X, a *block* is a subset $B \subseteq X$ such that for each $g \in G$, either $g(B) = B$ or $g(B) \cap B = \emptyset$. A *block system* is a partition of X into blocks. A block system $\mathbf{B}$ will be said to be *compatible* with the group action if $g(B) \in \mathbf{B}$ for all $g \in G$ and $B \in \mathbf{B}$. A characterization of complete block systems (Theorem 9) is relevant to the understanding of fixed assembly trees because of the following result. Let τ be any assembly tree in $\mathcal{T}_X$. For any vertex v of τ, recall that v is identified with and labeled by its set of descendent leaf-labels. Thus the set of labels of the children of the root is a partition of X.

Lemma 8 *Let G act on X, and let τ be an assembly tree for X that is fixed by G. If U is the set of children of the root of τ, then U is a block system that is compatible with the action of G in X.*

Proof: For any $v \in U$, let τ_v be the rooted, labeled subtree of τ that consists of root v and all its descendents. If τ is fixed by G, then $g(\tau) = \tau$ for each $g \in G$. In other words, $g(\tau_v) = \tau_u$ for some $u \in U$. This implies that $g(v) \cap v = \emptyset$ if $u \neq v$ or $g(v) \cap v = v$ if $u = v$. Hence U is block system that is compatible with the action of G on X. $\square$

The following notation will be used in this section. The set of orbits of G acting on X will be denoted by $\mathbf{O}$. For $H \leq G$, let $\mathbf{C}_H$ denote a set of (say left) coset representatives of H in G. Note that $|\mathbf{C}_H| = (G : H)$. For $r \in G$ and $Q \subseteq X$, let $r(Q) := \{r(q) : q \in Q\}$. A group G is said to act *simply* on X if the stabilizer of each x in X is the trivial group. In this paper, a *partition* Π of a finite set S into k parts is a set $\{\pi_1, \pi_2, \cdots, \pi_k\}$ of disjoint subsets so that $\cup_{i=1}^k \pi_i = S$. The subsets π_i are called the *parts* of the partition Π. The order of the parts of a partition is insignificant. That is, $\{\{1,3\},\{2,4\}\}$ and $\{\{2,4\},\{1,3\}\}$ are identical partitions of the set $\{1,2,3,4\}$. We nevertheless label the parts from 1 to k for convenience.

Theorem 9 *Let us assume that G acts simply on X. Let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_k\}$ be a partition of $\mathbf{O}$ into arbitrarily many parts, and let $\mathbf{H} = \{H_1, H_2, \ldots, H_k\}$ be an arbitrary set of distinct subgroups of G. For each i and each $O \in \pi_i$, let $Q_{i,O}$ be any single orbit of the simple action of H_i on O. (As G acts simply on X, so do its subgroups.)*

Let $Q_i = \cup_{O \in \pi_i} Q_{i,O}$. That is, Q_i contains one orbit of the action of H_i on O, for each O (often Q_i is a proper subset of O). Let $\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_k\}$. (See Example 10 for an illustration.) Let us denote by $(\Pi, \mathbf{H}, \mathbf{Q})$ the arrangement $\{(\pi_1, H_1, Q_1), \ldots, (\pi_k, H_k, Q_k)\}$ of each π_i in Π with a corresponding subgroup $H_i \leq G$ in $\mathbf{H}$, and $Q_i \in \mathbf{Q}$.

1. *The collection*

$$\mathbf{B}(\Pi, \mathbf{H}, \mathbf{Q}) = \bigcup_{i=1}^{k} \bigcup_{r \in \mathbf{C}_{H_i}} r(Q_i),$$

of blocks $r(Q_i)$ is a compatible block system for G acting on X.

2. *Every compatible block system for G acting on X is of the above form for some choice of* Π, $\mathbf{H}$, *and* $\mathbf{Q}$.

3. *Two such block systems* $\mathbf{B}(\Pi, \mathbf{H}, \mathbf{Q})$ *and* $\mathbf{B}(\Pi', \mathbf{H}', \mathbf{Q}')$ *are equal if and only if, there is a permutation p of the set of blocks of Π' so that for all $i \leq k$, we have $\pi'_{p(i)} = \pi_i$, and for all $i \leq k$, there exists a $g_i \in G$ so that $H'_{p(i)} = g_i H_i g_i^{-1}$, and $Q'_{p(i)} = g_i(Q_i)$.*

Example 10 *Figure 11 illustrates the various orbits defined in Theorem 9. In this example, the action of G on X consists of four orbits, O_a, O_b, O_c and O_d. The partition Π has two blocks, $\pi_1 = \{O_a, O_b\}$ and $\pi_2 = \{O_c, O_d\}$. The action of H_1 on π_1 creates orbits on O_a and O_b. We choose one of each, (shown in the corners), for the roles of Q_{1,O_a} and Q_{1,O_b}. Similarly, the action of H_2 on π_2 creates orbits on O_c and O'_d, and we choose one of each, (shown again in the corners), for the roles of Q_{2,O_a} and Q_{2,O_b}. Finally, $Q_1 = Q_{1,O_a} \cup Q_{1,O_b}$, and $Q_2 = Q_{2,O_c} \cup Q_{2,O_d}$.*

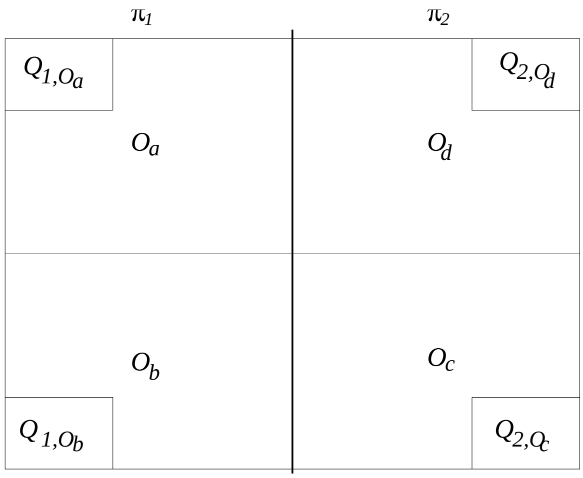


Figure 11: The families of orbits described in Theorem 9

.

Proof: (of Theorem 9) In order to prove Statement (1), let $H \in \mathbf{H}$ and $B = r(Q)$, where $Q = Q_i$ for some i. We first show that B is a block. There is a subset $A \subseteq X$ containing at most one element from each G-orbit such that $B = rH(A)$. If $g(B) \cap B \neq \emptyset$, then there are elements $a, a' \in A$ such that $grh(a) = rh'(a')$ for some $g \in G$ and $h, h' \in H$. Thus a and a' are in the same G-orbit, which implies that $a = a'$. Therefore, $grh(a) = rh'(a)$. Since G acts simply, this implies that $grh = rh'$, which in turn implies that gr and r are in the same coset of H in G. Therefore $g(B) = gr(Q) = r(Q) = B$. This proves, not only that B is a block, but that $\mathbf{B}(\Pi, \mathbf{H}, \mathbf{Q})$ is a block system, because $\mathbf{B}(\Pi, \mathbf{H}, \mathbf{Q})$ is a partition of X into blocks. Moreover, if $r(Q) \in \mathbf{B}(\Pi, \mathbf{H}, \mathbf{Q})$ and $g \in G$, then by definition $gr(Q) \in \mathbf{B}(\Pi, \mathbf{H}, \mathbf{Q})$, which shows that $\mathbf{B}(\Pi, \mathbf{H}, \mathbf{Q})$ is a block system compatible with G.

In order to prove Statement (2), let us denote the set of orbits of G in its action on X by $\{O_1, O_2, \ldots, O_n\}$. We first show that any block B in the action of G on X is of the form $B = B_1 \cup B_2 \cup \cdots \cup B_n$, where B_i is a single orbit of some subgroup $H \le G$ acting on O_i. Let $B_i = B \cap O_i$. Note that $B_i = \emptyset$ is a possibility, in which case we have $B = B_1 \cup B_2 \cup \cdots \cup B_m$, $m \le n$. Each B_i itself must be a block because, if $g(B_i) \cap B_i \neq \emptyset$, then $g(B) \cap B \neq \emptyset$. However, B is a block, so $g(B) \cap B \neq \emptyset$ implies that $g(B) = B$, and thus $g(B_i) = B_i$.

Let $H_i = \{h \in G \,|\, h(B_i) = B_i\}$. We claim that $H_1 = H_2 = \cdots = H_m$. To see this let $h \in H_i$. Since B is a block, either $h(B) \cap B = \emptyset$ or $h(B) = B$. However, $h(B) \cap B = \emptyset$ is impossible because $h(B_i) = B_i$. Hence $h(B) = B$. Now $B_j = B \cap O_j$ implies, for each j, that $h(B_j) = B_j$. Therefore $h_i \in H_j$ for all i, j. This verifies the claim, so let $H = H_1 = H_2 = \cdots = H_m$.

The proof that each block B is of the required form is complete if it can be shown that H acts transitively on B_i for each i. To see this, let $x, y \in B_i$. Since B_i lies in a single G-orbit, there is a $g \in G$ such that $g(x) = y$. Since B_i has been shown to be a block and $g(B_i) \cap B_i \neq \emptyset$, it must be the case that $g(B_i) = B_i$. Therefore $g \in H_i = H$.

To complete the proof of Statement (2), let $\mathbf{B}$ be any compatible block system for G acting on X. We have proved that if $B \in \mathbf{B}$, then $B = B_1 \cup B_2 \cup \cdots \cup B_m$, where B_i is a single orbit of some subgroup $H \le G$ acting on O_i. Because of the compatibility, the action of G on X induces an action of G on $\mathbf{B}$. The orbits under this action provide a partition Π of $\mathbf{O}$, a part $\pi \in \Pi$ consisting of all G-orbits acting on X contained in the union of a single G-orbit acting on $\mathbf{B}$. Consider any orbit W of B in this action. If B' is another element of W, then there is an $r \in G$ such that $B' = r(B)$. This shows that the blocks in $G(B)$ are of the desired form in Statement (1) of the theorem. Repeating this argument for each part in the partition Π completes the proof of Statement (2).

To prove Statement (3), we first show that if $Q \in \mathbf{Q}$ is the union of H-orbits and $H'(Q) = Q$, where $H, H' \in \mathbf{H}$, then $H' = H$. Restricting attention to just one orbit of G in its action on X, the equality $H'(Q) = Q$ implies that $H'(a') = H(a)$ for some a, a' in the same G-orbit acting on X. Let $g \in G$ be such that $a = g(a')$ and hence $Hg(a') = H'(a')$, which in turn implies that $hg(a') = a'$ for some $h \in H$. Because G acts simply, this implies that $g = h^{-1} \in H$, so $H(a') = H'(a')$, which again, by the simplicity of the action, implies that $H' = H$.

Now let us assume that $\mathbf{B}(\Pi, \mathbf{H}, \mathbf{Q}) = \mathbf{B}(\Pi', \mathbf{H}', \mathbf{Q}')$. Clearly, $\Pi = \Pi'$. It is sufficient to restrict our attention to just one of the parts in the partition $\Pi = \Pi'$, so we must show that $\{r(Q) : r \in \mathbf{C}_H\} = \{r(Q') : r \in \mathbf{C}_{H'}\}$ if and only if $H' = gHg^{-1}$, and $Q' = g(Q_i)$ for some $g \in G$. If $H' = gHg^{-1}$, and $Q' = g(Q)$ for some $g \in G$, then for any $r \in G$ we have $r(Q') = rH'(Q') = (rgHg^{-1})g(Q) = rgH(Q)$. This shows that $\{r(Q') : r \in \mathbf{C}_{H'}\} \subseteq \{r(Q) : r \in \mathbf{C}_H\}$, and the opposite inclusion is similarly shown. Conversely, assume that $\{r(Q) : r \in \mathbf{C}_H\} = \{r(Q') : r \in \mathbf{C}_{H'}\}$. Since $Q' \in \{r(Q') : r \in \mathbf{C}_{H'}\}$, we know that $Q' = r(Q)$ for some $r \in \mathbf{C}_H \subseteq G$. Now $(rHr^{-1})(Q') = (rHr^{-1})(r(Q)) = rH(Q) = r(Q) = Q'$. By the uniqueness result shown in the preceding paragraph, we get $H' = rHr^{-1}$. $\square$

Example 11 *Klein 4-group acting on $\mathcal{T}_4$ (continued).*

Continuing the example from the previous section with $G = \mathbb{Z}_2 \oplus \mathbb{Z}_2$ acting simply on $X = \{1, 2, 3, 4\}$, let $K = K_1 = \{(1)(2)(3)(4), (1\,2)(3\,4)\}$ and K_0 the trivial subgroup. There are 11 blocks in the action of K on X which are given below:

$$\{1, 2, 3, 4\}, \{1, 2\}, \{3, 4\}, \{1, 3\}, \{2, 4\}, \{1, 4\}, \{2, 3\}, \{1\}, \{2\}, \{3\}, \{4\}.$$

The seven block systems for the action of K on X can be found using Theorem 9. In what follows, $\{1, 2\}|\{3, 4\}$ denotes the orbit $\{1, 2\}, \{3, 4\}$ partitioned into the two parts $\{1, 2\}$ and $\{3, 4\}$, whereas $\{1, 2\}, \{3, 4\}$ denotes that same orbit partitioned the trivial way, into one part. Note that $\mathbf{B}\ (\ \{1, 2\}|\{3, 4\},\ \{K_0, K\}\ ,\ \{2\}|\{3, 4\}\ )$, for example, is not included in the list below. This is because, according to Statement (3) in Theorem 9,

$$\begin{aligned} &\mathbf{B}\ (\ \{1, 2\}|\{3, 4\},\ \{K_0, K\}\ ,\ \{2\}|\{3, 4\}\ ) = \\ &\mathbf{B}\ (\ \{1, 2\}|\{3, 4\},\ \{K_0, K\}\ ,\ \{1\}|\{3, 4\}\ ). \end{aligned}$$

Namely, for $g = (1\,2)(3\,4)$, we have $\{2\} = g(\{1\})$ and $K_0 = gK_0g^{-1}$.

$$\begin{aligned} \mathbf{B}\ (\ \{1, 2\}, \{3, 4\},\ \{K\},\ \{1, 2, 3, 4\}\ \ ) &= \ (1\,2\,3\,4) \\ \mathbf{B}\ (\ \{1, 2\}, \{3, 4\},\ \{K_0\},\ \{1, 3\}\ ) &= \ (1\,3)(2\,4) \\ \mathbf{B}\ (\{1, 2\}, \{3, 4\},\ \{K_0\},\ \{1, 4\}\ ) &= \ (1\,4)(2\,3) \\ \mathbf{B}\ (\ \{1, 2\}|\{3, 4\},\ \{K, K\},\ \{1, 2\}, \{3, 4\}\ ) &= \ (1\,2)(3\,4) \\ \mathbf{B}\ (\{1, 2\}|\{3, 4\},\ \{K, K_0\},\ \{1, 2\}, \{3\}\ ) &= \ (1\,2)(3)(4) \\ \mathbf{B}\ (\{1, 2\}|\{3, 4\},\ \{K_0, K\},\ \{1\}, \{3, 4\}\ ) &= \ (1)(2)(3\,4) \\ \mathbf{B}\ (\{1, 2\}|\{3, 4\},\ \{K_0, K_0\},\ \{1\}, \{2\}\ ) &= \ (1)(2)(3)(4) \end{aligned}$$

Let $\tau \in \mathcal{T}_X$ be a tree fixed by G in its action on T_X. If U denotes the set of children of the root of τ, recall that Lemma 8 states that the set U of labels is a block system. Recall that the label of a vertex is the set of labels of its leaf descendents, and also the label of a vertex is the union of the labels of its children. According to Theorem 9, any block system is of the form

$$\mathbf{B}(\Pi, \mathbf{H}, \mathbf{Q}) = \bigcup_{i=1}^{k} \bigcup_{r \in \mathbf{C}_{H_i}} r(Q_i).$$

We will use the notation τ_{rQ} to denote the subtree τ_u, $u \in U, u = r(Q)$, rooted at u. Theorem 9 leads to the characterization of assembly trees fixed by given a group G as stated in Corollary 12 below.

Corollary 12 *Let us assume that G acts simply on X and that $\tau \in \mathcal{T}_X$. Let U be the set of children of the root of τ and, for each $u \in U$, let τ_u be the rooted, labeled subtree of τ that consists of root u and all its descendents. Then, with notation as in Theorem 9, the tree τ is fixed by G if and only if, for some Π, $\mathbf{H}$ and $\mathbf{Q}$, the following two conditions are satisfied.*

1. *The equality $U = \mathbf{B}(\Pi, \mathbf{H}, \mathbf{Q})$ holds. In particular, for each $Q \in \mathbf{Q}$ and $g \in G$, there is a subtree τ_Q and a subtree τ_{gQ}.*

2. *For every $Q \in \mathbf{Q}$ and every $g \in G$, the equality $\tau_{gQ} = g(\tau_Q)$ holds.*

Proof: Let us assume that τ is fixed by G. Condition (1) follows immediately from Lemma 8 and Theorem 9. Concerning Condition (2), for any $g \in G$, the set of leaves of $g(\tau_Q)$ is $g(Q)$. Hence for τ to be fixed by G it is necessary that $g(\tau_Q) = \tau_{gQ}$.

Conversely, let us assume that Conditions (1) and (2) hold. For any $g \in G$ we must show that $g(\tau) = \tau$. By Condition (1), it is sufficient to show that g acting on τ permutes the set of subtrees in such a manner that $g(\tau_{rQ}) = \tau_{grQ}$ for every $H \in \mathbf{H}$, Q the corresponding element of $\mathbf{Q}$, and every $r \in \mathbf{C}_H$. However, by Condition (2), $g(\tau_{rQ}) = gr(\tau_Q) = \tau_{grQ}$. □

Theorem 14 below states that the following recursive procedure constructs any assembly tree $\tau \in \mathcal{T}_X$ fixed by G. This will be used to prove Theorem 17 in the next section.

Procedure 13 *Recursive construction of any assembly tree fixed by a group G:*

(1) Partition the set $\mathbf{O}$ of G-orbits of X: $\Pi = \{\pi_1, \pi_2, \ldots, \pi_k\}$. Note that the parts of Π are labeled $1, 2, \cdots, k$ in some arbitrary way.

(2) For each $i = 1, 2, \ldots, k$, choose a subgroup $H_i \leq G$. (If Π has only one part then $H_i = G$ is not allowed.)

(3) For each i, choose a single orbit of H_i acting on each of the G-orbits in π_i, and let Q_i be the union of these H_i-orbits.

(4) Recursively, let τ_{Q_i} be any rooted tree whose leaves are labeled by Q_i and which is fixed by H_i.

(5) Let $S_i = \{\ r(\tau_{Q_i}) \mid r \in \mathbf{C}_{H_i}\}$ and $S = \cup_{i=1}^{k} S_i$.

(6) Let τ be the rooted tree whose children are roots of the trees in S.

Theorem 14 *The set of assembly trees constructed by Procedure 13 is the set of assembly trees fixed by the group G.*

Proof: In the notation of Theorem 9, Steps (1), (2), and (3) are choosing $(\Pi, \mathbf{H}, \mathbf{Q})$. Steps (4), (5), and (6) are ensuring that $U = \mathbf{B}(\Pi, \mathbf{H}, \mathbf{Q})$. Note that the restriction in Step (2) is because otherwise the root of the resulting tree in Step (6) would have only one child. Note also that in Step (5), S_i does not depend on the particular set of coset representatives. This follows directly from Step (4).

It is now sufficient to show the following. For any assembly tree τ satisfying $U = \mathbf{B}(\Pi, \mathbf{H}, \mathbf{Q})$ for some $(\Pi, \mathbf{H}, \mathbf{Q})$, Condition (2) in Corollary 12 holds if and only if τ is constructed by Procedure 13. To show that any assembly tree τ constructed by Procedure 13 satisfies Condition (2), note that Step (4) implies that, if $Q \in \mathbf{Q}$ corresponds to $H \in \mathbf{H}$, then $H(Q) = Q$ and hence $h(\tau_Q) = \tau_{hQ} = \tau_Q$ for all $h \in H$. For $g \in G$, if $g = rh$, where $h \in H$, then $g(\tau_Q) = rh(\tau_Q) = r(\tau_{hQ}) = r(\tau_Q) = \tau_{rQ}$, the last equality from Step (5). Again, because $H(Q) = Q$, we have $g(\tau_Q) = \tau_{rQ} = \tau_{rhQ} = \tau_{gQ}$.

Conversely, if τ satisfies Condition (2) in Corollary 12, then consider the trees τ_{Q_i} $i = 1, 2, \ldots, k$. These are trees whose leaves are labeled by Q_i. in Step (4) of Procedure 13. Moreover, by Condition (2) we have $h(\tau_{Q_i}) = \tau_{hQ_i} = \tau_{Q_i}$ for all $h \in H_i$, so τ_{Q_i} is fixed by H_i. By Step (5) of Procedure 13 and Condition (2) of Corollary 12 we have $r(\tau_{Q_i}) = \tau_{rQ_i}$ for all $r \in \mathbf{C}_{H_i}$. Therefore the tree τ is constructed by Procedure 13. □

Remark 15 *Enforcing uniqueness in the construction.*

The construction in Procedure 13 is not unique, in that it may produce the same fixed assembly tree multiple times depending on the choices in Steps 2 and 3. Condition (3) in Theorem 9 shows that we may enforce uniqueness if we make the following two restrictions.

(a) If we choose $(\Pi, \mathbf{H}, \mathbf{Q}) = \{(\pi_1, H_1, Q_1), \ldots, (\pi_k, H_k, Q_k)\}$ in Steps 1 and 2 of the procedure while constructing a tree τ, and if we also have $(\Pi, \mathbf{H}', \mathbf{Q}') = \{(\pi_1, H'_1, Q'_1), \ldots, (\pi_k, H'_k, Q'_k)\}$ during the construction of another tree τ', then to ensure that $\tau \neq \tau'$ we need to ensure that for at least one i, the group H'_i should not be conjugate to H_i in G.

(b) Consider the construction of two trees τ and τ' with corresponding $(\Pi, \mathbf{H}, \mathbf{Q})$ and $(\Pi, \mathbf{H}', \mathbf{Q}')$ such that for each i, the subgroup H_i is a conjugate of the subgroup H'_i. Further assume that in Step (3) for the tree τ the element $g_i \in H_i$ is such that $g_i H_i g_i^{-1} = H'_i$. Then while constructing tree τ', we need to ensure that there is at least one i such that $Q'_i \neq g_i(Q_i)$. (Note that for a given index i, there may well be several elements $g_i \in G$ so that $g_i H_i g_i^{-1} = H'_i$ holds, and all those are subject to this restriction.)

Example 16 *Klein 4-group acting on $\mathcal{T}_4$ (continued).*

With $G = \mathbb{Z}_2 \oplus \mathbb{Z}_2$ acting on $X = \{1, 2, 3, 4\}$, consider the assembly trees τ fixed by the subgroup $K = \{(1)(2)(3)(4), (1\,2)(3\,4)\}$. There are exactly six such trees, those in the orbits A, B, C, D, E of Figure 6. These correspond (not in corresponding order) to the block systems in Example 11. Because of the restriction in Step (2) of Procedure 13, the first block system in the list in Example 11 is ignored.

5.1 When the action of G on X is not simple

Let us assume that G acts on X, but not necessarily simply. For $q \in X$, let S_q denote the stabilizer of q in G. For a subset $Q \subseteq X$, let

$$S_Q = \bigcup_{q \in Q} S_q.$$

If G acts simply on X, then the stabilizer of any $x \in X$ is the trivial subgroup. Therefore, in this case, it is clear that $S_Q \subset H$ for any $Q \subseteq X$ and $H \leq G$. In the general case, when G acts not necessarily simply on X, let us call a pair (H, Q) *viable* if

$$\mathbf{S}_Q \subseteq H.$$

If only viable pairs (H_i, Q_i) are allowed in the hypothesis of Theorem 9, then the theorem is valid in the general, not necessarily simple, case. Since this general version of Theorem 9 and associated analogs of Procedure 13 and Theorem 14 are not needed in subsequent sections, and the proofs are relatively straightforward extensions, we omit them.

6 Enumerating Fixed Assembly Trees

Let us assume in this section that G acts simply on each of an infinite sequence $X_1, X_2, \ldots$ of sets where, by formula (1) in Section 2, we have $|X_n| = n|G|$. In other words, n is the number of orbits of G in its action on X_n. Denote by $t_n(G)$ the number of trees in $\mathcal{T}_n := \mathcal{T}_{X_n}$ that are fixed by G. In this section we provide a formula for the exponential generating function

$$f_G(x) := \sum_{n \geq 1} t_n(G) \, \frac{x^n}{n!}$$

for the sequence $\{t_n(G)\}$. If G is the trivial group of order one, then let us denote this generating function simply by $f(x)$. This is the generating function for the total number of rooted, labeled trees with n leaves in which every non-leaf vertex has at least two children. For $H \leq G$, let

$$\widehat{f}_H(x) = \frac{1}{(G:H)} \, f_H\left((G:H)x\right).$$

Theorem 17 *The generating function $f_G(x)$ satisfies the following functional equations:*

$$1 - x + 2f(x) = \exp\left(f(x)\right),$$

and for $|G| > 1$,

$$1 + 2f_G(x) = \exp\left(\sum_{H \leq G} \widehat{f}_H(x)\right).$$

Proof: The first formula is proved in [21], page 13. For $|G| > 1$, we use the standard exponential and the product formulas for generating functions.

The proof of the second formula uses two well known results from the theory of exponential generating functions, the "product formula" and the "exponential formula". In Procedure 13, give Steps (3) and (4) the name *putting an H_i-structure on π_i*. According to Theorem 14, the number of trees $t_n(G)$ fixed by G equals the number of ways to partition the set of orbits of G acting on X_n and to place an H-structure on each part in the partition, for some subgroup $H \leq G$, keeping the uniqueness Remark 15 in mind.

In Step (3) of Procedure 13, since G acts simply and the number of H_i-orbits in one G-orbit is $|G|/|H_i| = (G : H_i)$, the number of possible choices for Q_i (the union of these single H_i-orbits) is $(G : H)^m$. Hence, in accordance with Step (4) of Procedure 13, the generating function for the number of ways to place an H-structure is basically $f_H((G : H)x)$.

However, this must be altered in accordance with the uniqueness requirements in Remark 15. Let $\mathbf{N}$ denote a set consisting of one representative of each conjugacy class in the set of subgroups of G. By Statement (a) in Remark 15, only subgroups in $\mathbf{N}$ are considered. Let $N(H) := \{g \in G \,|\, gHg^{-1} = H\}$ denote the *normalizer* of H in G. By Statement (b), there has to be an index i so that $g(Q_i) \neq Q_i'$. However, $g(Q_i) = g'(Q_i)$ will occur for every i if and only if g and g' are in the same coset of H in G. Therefore, the generating function for the number of ways to place an H-structure is $\frac{1}{(N(H):H)}\, f_H((G : H)x)$.

The exponential formula states that the generating function $g_H(x) = \sum_{n \geq 0} a_n \frac{x^n}{n!}$ for the number of ways a_n to partition the set of G-orbits acting on X_n and, on each part π in the partition, place an H-structure (same H) is

$$g_H(x) := \exp\left(\widehat{f}_H(x)\right).$$

Here we assume that $a_0 = 1$.

The generating function for the number of ways to partition the set of orbits, i.e., choose $\Pi = (\pi_1, \pi_2, \dots, \pi_k)$ and, on each part of the partition, place an H-structure, one H from each conjugacy class in $\mathbf{N}$ is

$$\begin{aligned}\prod_{H \in \mathbf{N}} g_H(x) &= \prod_{H \in \mathbf{N}} \exp\left(\frac{1}{(N(H) : H)}\, f_H((G : H)x)\right) \\ &= \exp\left(\sum_{H \in \mathbf{N}} \frac{1}{(N(H) : H)}\, f_H((G : H)x)\right).\end{aligned}$$

Note that we have not taken the restriction in Step (2) of Procedure 13 into consideration. Taking the partition of the orbit set into just one part and placing on that part a G-structure results in counting the number of fixed trees a second time. Also since the constant term in $\prod_{H \in \mathbf{N}} g_H(x)$ is 1,

$$\begin{aligned}1 + 2\, f_G(x) &= \exp\left(\sum_{H \in \mathbf{N}} \frac{1}{(N(H) : H)}\, f_H((G : H)x)\right) \\ &= \exp\left(\sum_{H \leq G} \frac{1}{(G : H)}\, f_H((G : H)x)\right).\end{aligned}$$

Here the last equality holds because $f_H(x)$ depends only the conjugacy class of H in G and

$$\frac{1}{(N(H):H)}\Big/\frac{1}{(G:H)} = \frac{(G:H)}{(N(H):H)} = (G:N(H)) = |\mathbf{N}|.$$

$\square$

Example 18 *Klein 4-group acting on $\mathcal{T}_4$ (continued).*

Consider $G = \mathbb{Z}_2 \oplus \mathbb{Z}_2$ acting on X_n. Recall that $|X_n| = 4n$, the integer n being the number of G-orbits. In this case $\mathbf{N} = \{K_0, K_1, K_2, K_3, G\}$, where K_0 is the trivial group and

$$\begin{aligned}
K_1 &= \{\,(1)(2)(3)(4), (1\,2)(3\,4)\,\},\\
K_2 &= \{\,(1)(2)(3)(4), (1\,3)(2\,4)\,\},\\
K_3 &= \{\,(1)(2)(3)(4), (1\,4)(2\,3)\,\}.
\end{aligned}$$

The functional equations in the Statement of Theorem 17 are

$$\begin{aligned}
1 - x + 2f(x) =&\ \exp\left(f(x)\right)\\
1 + 2f_{K_i}(x) =&\ \exp\left(\frac{1}{2}\,f(2x) + f_{K_i}(x)\right) \quad \text{for } i = 1, 2, 3, \text{ and}\\
1 + 2f_G(x) =&\ \exp\left(\frac{1}{4}\,f(4x) + \frac{1}{2}\,f_{K_1}(2x) + \frac{1}{2}\,f_{K_2}(2x) + \frac{1}{2}\,f_{K_3}(2x) + f_G(x)\right).
\end{aligned}$$

Using these equations and MAPLE software, the coefficients of the respective generating functions provide the following first few values for the number of fixed assembly trees. For the first entry $t_1(G) = 4$ for the group G, the four fixed trees are shown in Figure 6 A, B, C, D. For trees with eight leaves there are $t_2(G) = 104$ assembly trees fixed by $G = \mathbb{Z}_2 \oplus \mathbb{Z}_2$, and so on.

$$\begin{aligned}
t_n(K_0) &\quad : \quad 1, 1, 4, 26, 236, 2752\\
t_n(K_i) &\quad : \quad 1, 6, 72, 1312, 32128, 989696\\
t_n(G) &\quad : \quad 4, 104, 4896, 341120, 31945728, 3790876672.
\end{aligned}$$

Theorem 17 provides the generating function for the numbers $t_n(H)$ of fixed assembly trees in the action of any subgroup $H \leq G$ on X_n. What is required for Problem (i) described in Sections 2 and 3 are the numbers $\bar{t}_n(H)$ of assembly trees that are fixed by H, but by no other elements of G. In Example 16, for $G = \mathbb{Z}_2 \oplus Z_2$ acting on $X = \{1, 2, 3, 4\}$, there are six trees that are fixed by the subgroup $K = \{\,(1)(2)(3)(4), (1\,2)(3\,4)\,\}$. However, of these six, four (A, B, C, and D in Figure 6) are also fixed by G. Therefore there are only two assembly trees fixed by K and no other elements of G (these are E, and F in Figure 6). In general, as shown Theorem 2, Möbius inversion [23] can be used to calculate the values of $\bar{t}_X(H)$ from the values of $t_X(H)$.

7 The Icosahedral Group and Viral Interpretation

For completeness, the results of the previous sections are applied to the icosahedral group and interpreted in the context of the $T = 1$ viral capsid.

An *isometry* of 3-space is a bijective transformation that preserves length, and an isometry is called *direct* if it is orientation preserving. Rotations, for example, are direct, while reflections are not. A *symmetry* of a polyhedron is an isometry that keeps the polyhedron, as a whole, fixed, and a *direct symmetry* is similarly defined. The *icosahedral group* is the group of direct symmetries of the icosahedron. It is a group of order 60 denoted G_{60}.

As mentioned earlier, the viral capsid is modeled by a polyhedron P with icosahedral symmetry, whose set X of facets represent the protein monomers. The icosahedral group, acts on P and hence on the set X. It follows from the quasi-equivalence theory of the capsid structure that G_{60} acts simply on X. Formula (1) of Section 2 gives $|X| := |X_n| = 60n$, where n is the number of orbits. Not every n is possible for a viral capsid; n must be a T-number as defined in the introduction. Before the number of orbits of each size for the action of G_{60} on the set $\mathcal{T}_n := \mathcal{T}_{X_n}$ of assembly trees can be determined, basic information about the icosahedral group is needed.

The group G_{60} consists of:

- the identity,
- 15 rotations of order 2 about axes that pass through the midpoints of pairs of diametrically opposite edges of P,
- 20 rotations of order 3 about axes that pass through the centers of diametrically opposite triangular faces, and
- 24 rotations of order 5 about axes that pass through diametrically opposite vertices.

There are 59 subgroups of G_{60} that play a crucial role in the theory. Besides the two trivial subgroups, they are the following:

- 15 subgroups of order 2, each generated by one of the rotations of order 2,
- 10 subgroups of order 3, each generated by one of the rotations of order 3,
- 5 subgroups of order 4, each generated by rotations of order 2 about perpendicular axes,
- 6 subgroups of order 5, each generated by one of the rotations of order 5,
- 10 subgroups of order 6, each generated by a rotation of order 3 about an axis L and a rotation of order 2 that reverses L,

- 6 subgroups of order 10, each generated by a rotation of order 5 about an axis L and a rotation of order 2 that reverses L,
- 5 subgroups of order 12, each the symmetry group of a regular tetrahedron inscribed in P.

From the above geometric description of the subgroups, it follows that all subgroups of a given order are conjugate in the group G_{60}. Representatives of the conjugacy classes of the subgroups of the icosahedral group are denoted by $G_0, G_2, G_3, G_5, G_6, G_{10}, G_{12}, G_{60}$, where the subscript is the order of the group. The set of subgroups of G_{60} forms a lattice, ordered by inclusion. A partial Hasse diagram for this lattice $\mathbf{L}$ is shown in Figure 12. The number on the edge joining G_i (below) and G_j (above) indicate the number of distinct subgroups of order i contained in each subgroup of order j. The number in parentheses on the edge joining G_i (below) and G_j (above) indicate the number of distinct subgroups of order j containing each subgroup of order i. It is well-known that any finite partially ordered set P admits a Möbius function $\mu : P \times P \to \mathbb{Z}$. The Möbius function of $\mathbf{L}$ is shown in Table 1. The entry in the table corresponding to the row labeled G_i and column G_j is $\mu(G_i, G_j)$.

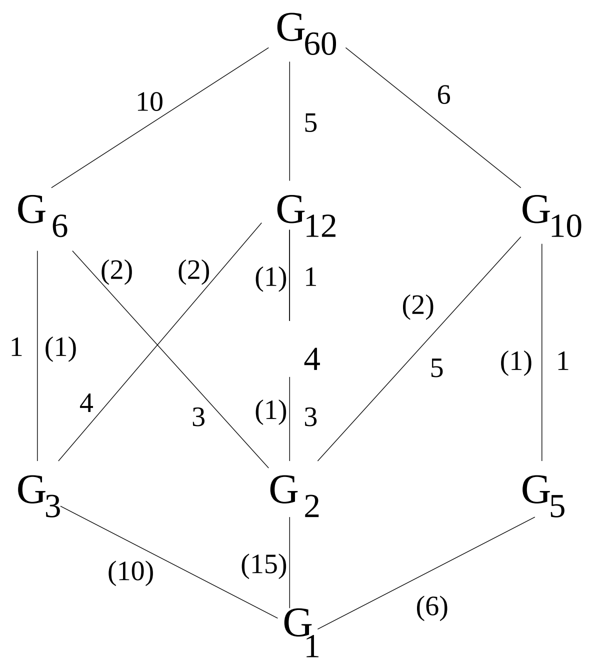


Figure 12: Partial Hasse diagram for the lattice of subgroups of the icosahedral group.

7.1 Interpretation for $T = 1$ capsids

For $|X| = 60$, i.e., for the $T = 1$ polyhedral case, using Theorem 17 and MAPLE software, the generating functions $f_{G_i}(x)$ were computed, and hence their coefficients

	G_1	G_2	G_3	G_4	G_5	G_6	G_{10}	G_{12}	G_{60}
G_1	1	−1	−1	2	−1	3	5	4	60
G_2	0	1	0	−1	0	−1	−1	0	0
G_3	0	0	1	0	0	−1	0	−1	2
G_4	0	0	0	1	0	0	0	−1	0
G_5	0	0	0	0	1	0	−1	0	0
G_6	0	0	0	0	0	1	0	0	−1
G_{10}	0	0	0	0	0	0	1	0	−1
G_{12}	0	0	0	0	0	0	0	1	−1
G_{60}	0	0	0	0	0	0	0	0	1

Table 1: The values of the Möbius function of the subgroup lattice of G_{60}.

$t_{60/i}(G_i)$ which count assembly trees that are fixed by any copy of G_i were also computed. Note that since $|X| = 60$, the number of orbits of G_i in its action on X is $60/i$. Substituting these values into Theorem 2 and using the Möbius Table 1 yields the following numerical values for $\bar{t}_{60/i}(G_i)$, the number of assembly trees over X with $|X| = 60$ that are fixed by G_i but by no other elements of G_{60}. In other words, these are the numbers of trees whose stabilizer in G_{60} is exactly G_i.

$$
\begin{aligned}
\bar{t}_{60}(G_1) &= 192446551013243739472018473092218757112034675453 2\\
&\qquad 23663299651157554321390236282894103246708400665785 37680\\
\bar{t}_{30}(G_2) &= 1670856367100496379411587456529324583988755126499875584\\
\bar{t}_{20}(G_3) &= 1008715729445173142872099594 4759704\\
\bar{t}_{15}(G_4) &= 100413426735302700145351712 13312\\
\bar{t}_{12}(G_5) &= 20540071766413107840\\
\bar{t}_{10}(G_6) &= 61346927354448105268\\
\bar{t}_6(G_{10}) &= 223503950260\\
\bar{t}_5(G_{12}) &= 16865654580\\
\bar{t}_1(G_{60}) &= 204
\end{aligned}
$$

Substituting the above numbers into Theorem 3 and Corollary 5, we arrive at the number of assembly pathways of each possible probability (symmetry factor). More precisely, this probability must be of the form $m/|\mathcal{T}|$, where m divides evenly into 60 and $|\mathcal{T}|$ is the total number of assembly trees. In this case the number of assembly trees with probability $m/|\mathcal{T}|$ is

$$\frac{1}{m}\,(\#\text{ subgroups of order } 60/m \text{ in } G_{60})\,\bar{t}_i(G_{60/i}).$$

The exact numbers appear in the table below.

204	assembly pathways with probability $1/|\mathcal{T}|$
16865654580	assembly pathways with probability $5/|\mathcal{T}|$
223503950260	assembly pathways of probability $6/|\mathcal{T}|$
61346927354448105268	assembly pathways of probability $10/|\mathcal{T}|$
10270035883206553920	assembly pathways of probability $12/|\mathcal{T}|$
33471142245100900048450570711O4	assembly pathways of of probability $15/|\mathcal{T}|$
5043578647225865714360497972379852	assembly pathways of probability $20/|\mathcal{T}|$
8354281835502481897057937282646622 91994377563249937792	assembly pathways of probability $30/|\mathcal{T}|$
320744251688739565786697455153697928 520057792422039438832751929257202 317060471490172077847334442975628	assembly pathways of probability $60/|\mathcal{T}|$

It is worth comparing the first and last elements of the above list. While the individual pathways of corresponding to a stabilizer of type G_1 have a symmetry factor that is 60 times more than those that correspond to type G_{60}, there are about 10^{99} times more of the former type of pathway. That is, the probability that a randomly selected pathway corresponds to a stabilizer of type G_1 is 10^{99} times higher than the probability that a randomly selected pathway corresponds to the stabilizer of type G_{60}. In general, the probability that a randomly selected pathway has a stabilizer of type G_i is very sensitive to changes in i. The higher i is, the lower this probability is. The rate of change is much steeper than the rate of change in i.

8 Conclusion and Open Problems

We have developed an algorithmic and combinatorial approach to a problem arising in the modeling of viral assembly. Our results illustrate, not only that problems arising from structural biology can be of independent mathematical interest, but also that mathematical methods have a direct application in structural biology.

More specifically, we have developed techniques to analyze the probability of a capsid forming along a given assembly pathway. One remaining issue is how to extend these techniques to finding the probability of *valid* assembly pathways as defined in Section 1. Valid assembly trees can be characterized combinatorially, using generalized notions of connectivity of the graph of the polyhedron that models the capsid, i.e. the graph whose edges are the edges of the polyhedron [19]. Combining such graph theoretic restrictions with our techniques will likely require new ingredients. A second important issue is how to extend our techniques to *nucleation* in viral shell assembly. Mathematically [3], the problem is to estimate the proportion of valid assembly trees that have a subtree whose leaves form a specific subset of facets, for example a trimer or a pentamer, in the underlying polyhedron. Disrupting probable nucleations - for example by altering the interacting monomer residues that drive the nucleation - is an effective way to arrest assembly of the viral capsid and thus in controlling infectious diseases.

In addition to the above extensions of the theory, there is scope to tighten some results of the paper. For example, a finer complexity analysis for Algorithm **Stabilizer** could be based on using Sim's algorithm, strong generating sets, and the Cayley graph for G as input.

The result of [3] only tells us that the symmetry factor in the probability of an assembly pathway is at least the depth of the pathway. We in fact conjecture that the symmetry factor increases with the depth of the pathway. Proving this conjecture would, as discussed at the end of Section 1, strengthen the motivation for studying the symmetry factor.

A study of unlabeled trees that are g-unfixable may lead to relevant related results. Call a tree *g-unfixable* if there is no leaf-labeling so that the resulting labeled tree is fixed by the permutation g, and let us say that a tree is *G-unfixable* if it is g-unfixable for every nontrivial element of the group G. These properties are interesting for at least two reasons. First, they clarify the minimum quantifiable information in a labeled tree that is necessary to decide if it is fixed by a group element g: if the underlying unlabeled tree is g-unfixable, then the information in the labeling is unnecessary to make this decision. This may lead to efficient algorithms that use properties of the automorphism group of the tree to help in deciding whether a given labeled tree is fixed by the given group. Second, in the language of formal logic, these properties are likely to be monadic second order expressible [7, 25], permitting the application of limit laws for the asymptotic probabilities of finite structures satisfying such properties.

References

[1] M. Agbandje-McKenna, A.L. Llamas-Saiz, F. Wang, P. Tattersall and MG Rossmann. Functional implications of the structure of the murine parvovirus, minute virus of mice. *Structure*, 6:1369–1381, 1998.

[2] B. Berger and P.W. Shor. Local rules switching mechanism for viral shell geometry, *Technical report, MIT-LCS-TM-527*, 1995.

[3] M. Bóna and M. Sitharam Influence of symmetry on probabilities of icosahedral viral assembly pathways, *Computational and Mathematical Methods in Medicine: Special issue on Mathematical Virology, Stockley and Twarock Eds*, 2008.

[4] B. Berger, P. Shor, J. King, D. Muir, R. Schwartz and L. Tucker-Kellogg. Local rule-based theory of virus shell assembly, *Proc. Natl. Acad. Sci. USA*, 91:7732–7736, 1994.

[5] Gunnar Brinkmann and Andreas Dress. A constructive enumeration of fullerenes, *Journal of Algorithms.*, 23:345–358, 1997.

[6] D. Caspar and A. Klug. Physical principles in the construction of regular viruses, *Cold Spring Harbor Symp Quant Biol*, 27:1–24, 1962.

[7] K.J. Compton. A logical approach to asymptotic combinatorics II: monadic second-order properties, *J. Comb. Theory Ser. A* 50(1):110–131, 1989.

[8] W.H.E. Day. Optimal algorithms for comparing trees with labeled leaves, *Journal of Classification*, 2(1):7–26, 1985.

[9] M. Deza and M. Dutour. Zigzag structures of simple two-faced polyhedra, *Combin. Probab. Comput.*, 14(1-2):31–57, 2005.

[10] M. Deza, M. Dutour, and P. W. Fowler. Zigzags, railroads, and knots in fullerenes, *Chem. Inf. Comp. Sci.*, 44:1282–1293, 2004.

[11] P. Gawron, V. V. Nekrashevich, and V. I. Sushchanskii, Conjugacy classes of the automorphism group of a tree *Mathematical Notes* 65(6):787-790, 1999.

[12] Jack E. Graver, Brigitte Servatius, and Herman Servatius. *Combinatorial Rigidity,* Graduate Studies in Math., AMS, 1993.

[13] J. E. Johnson and J. A. Speir. Quasi-equivalent viruses: a paradigm for protein assemblies, *J. Mol. Biol.*, 269:665–675, 1997.

[14] M.H. Klin. On the number of graphs for which a given permutation group is the automorphism group (Russian), *English translation: Kibernetika* 5:892-870, 1973.

[15] C. J. Marzec and L. A. Day. Pattern formation in icosahedral virus capsids: the papova viruses and nudaurelia capensis β virus, *Biophys*, 65:2559–2577, 1993.

[16] D. Rapaport, J. Johnson and J. Skolnick. Supramolecular self-assembly: molecular dynamics modeling of polyhedral shell formation, *Comp Physics Comm*, 1998.

[17] V. S . Reddy, H. A. Giesing, R. T. Morton, A. Kumar, C.B. Post, C. L. Brooks, and J. E. Johnson. Energetics of quasiequivalence: computational analysis of protein-protein interactions in icosahedral viruses. *Biophys*, 74:546–558, 1998.

[18] Á. Seress. Permutation Group Algorithms, *Cambridge University Press*, 2003.

[19] M. Sitharam and M. Agbandje-McKenna. Modeling virus assembly using geometric constraints and tensegrity:avoiding dynamics, *Journal of Computational Biology*, 13(6):1232–1265, 2006.

[20] M. Sitharam and M. Bóna. Combinatorial enumeration of macromolecular assembly pathways, In *Proceedings of the International Conferecnce on bioinformatics and applications*. World Scientific, 2004.

[21] R. Stanley. Enumerative Combinatorics, Volume 2, *Cambridge University Press*, 1999.

[22] G. Valiente. Algorithms on Trees and Graphs, *Springer*, 2002.

[23] J.H. van Lint and R.M. Wilson. A Course in Combinatorics, *Cambridge University Press*, 2006.

[24] S. G. Wagner. On an identity for the cycle indices of rooted tree automorphism groups *Electronic Journal of Combinatorics*, 13:450–456, 2006.

[25] A. R. Woods, Coloring rules for finite trees and probabilities of monadic second order sentences, *Random Structures and Algorithms*, 10(4):453–485, 1998.

[26] A. Zlotnick, R Aldrich, J. M. Johnson, P. Ceres, and M. J. Young. Mechanisms of capsid assembly for an icosahedral plant virus, *Virology*, 277:450–456, 2000.

[27] A Zlotnick. To build a virus capsid: an equilibrium model of the self assembly of polyhedral protein complexes. *J. Mol. Biol.*, 241:59–67, 1994.

[28] A. Zlotnick, J. M. Johnson, P.W. Wingfield, S.J. Stahl, and D. Endres. A theoretical model successfully identifies features of hepatitis b virus capsid assembly, *Biochemistry*, 38:14644–14652, 1999.