Pólya's Enumeration Theorem and Some Applications

Cathaoir P. McShane

School of Mathematics and Physics
Queen's University Belfast

This dissertation is submitted for the degree of
MMath Mathematics

June 2024

DECLARATION

I hereby declare that except where explicit reference is made to the work of others, the contents of this dissertation are original and have not been submitted for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Cathaoir P. McShane
June 2024

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor Dr. Thomas Hüttemann. His expertise, patience and openness to discussion has been invaluable throughout my research journey. I would also like to give thanks to my family, close friends, and fellow students for their continual support. Many thanks go to the lecturers and staff of the school of mathematics and physics at Queen's University of Belfast for a thoroughly enjoyable and inspiring undergraduate experience.

ABSTRACT

Pólya's enumeration theorem (PET) was shown to answer a general class of enumeration problem in 1937. This thesis aims to present various formulations of this result and showcase its application to counting and enumeration problems. Preceding the first formulation of PET is a discussion of group actions, permutations and transformations between finite sets. A discussion of partitions and symmetric functions yields the other formulations from the first. An explicit relation between Burnside's lemma and PET is presented. Examples and applications are given throughout, including: counting the orbits of group actions, enumerating colourings of symmetric objects under group actions and enumerating derivatives of the benzene ring.

TABLE OF CONTENTS

1

INTRODUCTION

Renowned Hungarian mathematician George Pólya authored a paper in 1937 that has had a profound influence on combinatorial enumeration. It contained a theorem for solving a general class of enumeration problem. This theorem is often referred to as Pólya's enumeration theorem (PET). Alongside this, Pólya showcased applications of his theorem to the enumeration of graphs and chemical isomers. Said paper was first published in German, entitled *Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen*. which, in 1964, was translated into English in by British mathematician Ronald C. Read [8].

Austrian mathematician Martin Aigner includes Pólya's enumeration theorem in his book *Combinatorial theory* [1]. Published in 1979, this source looks to systematise a general theory for combinatorics and in doing so provides a neat algebraic approach to (at the time of its publishing) a blossomed field of Pólya theory. One drawback of this approach is that it is easy for a novice combinatorist to lose track of the motivation behind Pólya theory in the process.

This thesis aims to present Aigner's systematic approach, whilst reinforcing motivation through digestible examples. Two formulations of Pólya's enumeration theorem that can be applied to any problem from this general class are derived as well as three formulations that can be applied to subsets of this class. Another formulation that can be applied to a subset of this class is included without proof.

Enumeration and counting problems are scattered throughout. Some of which are solved with the help of computer algebra system GAP (Groups, Algebra and Programming) and general purpose programming language Python. In general, enumeration refers to listing items in a collection. We refer to the listed items as configurations. Let's look at a straightforward enumeration problem.

Example 1.0.1 (Motivating Example - Constructing Necklaces)
A necklace is comprised of a finite number of coloured beads arranged on a loop of string. We can represent a necklace as a word of length $n \in \mathbb{N}$ over some alphabet C of colours. Let $n = 4$ and $C = \{R, B\}$ where R denotes red and B denotes blue. Some necklaces and their corresponding word are shown below.

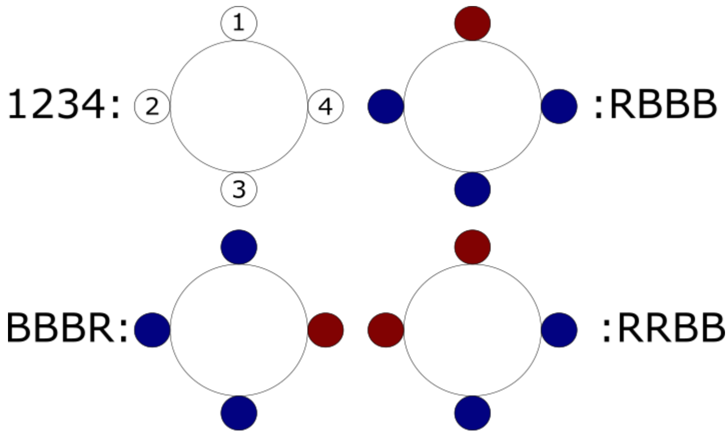


Fig. 1.1 Sample of necklaces for $n = 4$, $C = \{R, B\}$

In this scenario, there are two choices of for each letter. Due to this one may reason that there are $2 \cdot 2 \cdot 2 \cdot 2 = 16$ possible necklaces that can be constructed. However, if one was to wear the necklace in the top right of figure 1.1, they could rotate it around their neck and present it as the necklace in the bottom left.

More formally, under consideration of rotation $RBBB$ is equivalent to $BBBR$. Note that neither of these necklaces are equivalent to $RBRB$ under rotation, as no rotation of a physical necklace could transform a blue bead into a red bead. After some thought, one will come to the conclusion that there are six mutually inequivalent necklaces:

$$\begin{gathered} \text{RRRR}, \quad \text{BRRR}, \quad \text{BBRR}, \\ \text{BBBB}, \quad \text{RBBB}, \quad \text{RBRB}. \end{gathered}$$

These are referred to as the configurations of necklaces up to rotation for $n = 4$ and two bead colours. The set of configurations of necklaces for any finite number of beads and any finite set of colours up to rotation can be enumerated using Pólya's enumeration theorem.

2

PRELIMINARIES

The structure of this chapter is three-fold, consisting of two main sections on group theory and permutations respectively, alongside a shorter section of miscellaneous definitions that will prove useful in the subsequent chapters.

2.1 Groups and Group Actions

Definition 2.1.1 (Group)
A *group* is a pair $(G, \cdot)$ consisting of set G and operation $\cdot : G \times G \to G$ satisfying the properties:

1. $\forall\, a, b, c \in G : a \cdot (b \cdot c) = (a \cdot b) \cdot c$;

2. $\exists\, id \in G\; \forall\, a \in G : id \cdot a = a = a \cdot id$;

3. $\forall\, a \in G\; \exists\, a^{-1} \in G : a^{-1} \cdot a = id = a \cdot a^{-1}$.

We can use the shorthand notation $G = (G, \cdot)$ and $a \cdot b = ab$ if the operation is contextually clear. Let $(G, \cdot)$ be a group.

Definition 2.1.2 (Subgroup)
A non-empty subset H of G is a *subgroup* of G, denoted $H \leq G$, if:

1. $\forall\, a, b \in H : a \cdot b \in H$;

2. $\forall\, a \in H : a^{-1} \in H$.

Definition 2.1.3 (Left Coset)
Let $H \leq G$. The *left coset* of H by some $g \in G$ is the set

$$gH = \{gh \mid h \in H\}.$$

Moreover, the collection of left cosets induces a disjoint partition of G, defined

$$G/H = \{gH \mid g \in G\}.$$

A *right coset* is defined in a similar fashion.

Definition 2.1.4 (Normal Subgroup)
Let $N \leq G$. N is a *normal subgroup* of G, denoted $N \trianglelefteq G$, if every left coset of N is also a right coset of N (and vice versa).

Lemma 2.1.5 (Cardinality of Coset)
For any subgroup $H \leq G$ and any $g \in G$ we have that

$$|H| = |gH|.$$

Proof. Define $\phi : H \to gH$ by $\phi(h) = gh$. Suppose $\phi(h_1) = \phi(h_2)$. Then $gh_1 = gh_2$ whence $h_1 = h_2$. Therefore ϕ is injective. Choose some $a \in gH$. By definition $a = gh$ for some $h \in H$. Then $\phi(h) = gh = a$. Therefore ϕ is surjective. Conclude that ϕ is bijective and the result follows.

As the left cosets of a group G induce a disjoint partition of G and they all have equal cardinality we get Lagrange's theorem. Note that, in the case G is finite, H divides G and this is equal to the number of left cosets.

Proposition 2.1.6 (Lagrange)
For any subgroup $H \leq G$ we have that

$$|H| \cdot |G/H| = |G|. \quad \square$$

Definition 2.1.7 (Left Group Action)
Let X be a set. G acts on the left of X if there is a map

$$\alpha : G \times X \to X$$

such that for all $g \in G$ and all $x \in X$ we have that $\alpha(g, x) = g \cdot x \in X$ satisfying:

1. $\forall\, a, b \in G, x \in X : a \cdot (b \cdot x) = (ab) \cdot x;$
2. $\forall\, x \in X : id \cdot x = x.$

In the previous definition, α is our *left group action*. A *right group action* can be defined in a similar fashion. An action is called *faithful* if and only if the identity is the only element in G that when acting on every element of X returns the same element. Suppose α is the left action of a group G on a set X.

Definition 2.1.8 (Orbit)
Let $x \in X$. The *orbit* of x is the set of positions x is mapped to by α

$$O(x) = \{gx \mid g \in G\}.$$

Definition 2.1.9 (Stabilizer)
Let $x \in X$. The *stabilizer* of x is the set of elements in G such that α maps x to itself

$$G(x) = \{g \in G : gx = x\}.$$

If $gx = x$ for some g, we say that x is *fixed* by g.

Definition 2.1.10 (Fixed Points)
Let $g \in G$. The *fixed points* of g are the set of elements in X fixed by g

$$X_g = \{x \in X : gx = x\}.$$

Proposition 2.1.11 (Stabilizer-Subgroup)
For all $x \in X$, $G(x)$ is a subgroup of G.

Proof. By 2.1.9, it is clear that $G(x)$ is a subset of G. Let $x \in X$. By 2.1.7[2.], x is mapped to itself by id, and so $id \in G(x)$. Therefore, $G(x)$ is non-empty. Let $g, h \in G(x)$. Clearly $g(hx) = g(x) = x$. By 2.1.7[1.], must also have $(gh)x = x$ implying $gh \in G(x)$. Since $x = gx$ can apply g^{-1} to both sides yielding $g^{-1}x = g^{-1}gx = (g^{-1}g)x = x \implies g^{-1} \in G(x)$. Conclude that $G(x)$ satisfies 2.1.2. □

Definition 2.1.12 (Quotient by Action)
The *quotient of α* is the set

$$O(X) = \{O(x) \mid x \in X\}.$$

2.2 Permutations of a Set

Definition 2.2.1 (Composite Function)
Let $f : A \to B$ and $g : B \to C$ be arbitrary functions. Then their *composite*, denoted by $g \circ f$, is defined

$$g \circ f : A \to C, \;\; a \mapsto g(f(a)).$$

Routinely, function composition is associative.

Definition 2.2.2 (Permutation)
A *permutation* is a bijective function from a set X to itself.

We are only concerned with permutations of finite sets. A set X is finite if there exists a bijection between X and $\{1, 2, \cdots, n\}$ for some $n \in \mathbb{N}$. It is common in discrete mathematics to refer to a set containing n elements as an n-set. By the aforementioned bijective correspondence, we can represent any n-set as the set of the first n natural numbers.

Let f be a permutation of an n-set X. There are two common notations used to represent this permutation. The first is listing the elements of X alongside their image under f:

$$f = \begin{pmatrix} 1 & 2 & \cdots & n \\ f(1) & f(2) & \cdots & f(n) \end{pmatrix}.$$

We can then write the permutation as the string $f(1)f(2)\cdots f(n)$. The second is decomposing the permutation into finite cycles [12]. Letting c_i denote one such cycle and r denote the number of cycles, we can represent f as:

$$f = \prod_{i=1}^{r} c_i.$$

A cycle c_i can be determined by choosing some $x \in X$ not present in any previously determined cycle and recording its image $f(x)$ if not equal to x. Then the image of this term is recorded if not equal to x, and so on. This process repeats until $f^{a_i}(x) = x$ for some $a_i \in \mathbb{N}$ at which point the cycle terminates.

The length of c_i is the number of elements it contains, which is precisely a_i. The previous algorithm can be applied until each element of X appears in some cycle.

The starting point of a cycle nor the way in which the decomposition is ordered is important. Denote the number of cycles of length k as $b_k(f)$.

Definition 2.2.3 (Cycle Type)
The *type* of a permutation f is the expression

$$\tau(f) := t_1^{b_1(f)} t_2^{b_2(f)} \cdots t_n^{b_n(f)}$$

where each t_i is an indeterminate and each $b_i(f)$ is as previously defined.

Example 2.2.4
Let X be a 6-set. Define a permutation f of X as:

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 6 & 4 & 5 \end{pmatrix}.$$

Thus, have decomposition:

$$f = (1,2)(3)(4,6,5) = (3)(6,5,4)(2,1).$$

Hence, $\tau(f) = t_1^1 t_2^1 t_3^1$. It is common for the cycles of length one to be omitted from the finite cycle notation, but not from the cycle type:

$$f = (1,2)(4,6,5).$$

Example 2.2.5
Let X be a 3-set. The composition of two permutations of X is simply the composition of the associated functions. Define permutations f, g of X as:

$$f = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \quad g = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}.$$

Composing these functions (both ways):

$$f \circ g = \begin{cases} 1 \mapsto 2 \mapsto 1 \\ 2 \mapsto 1 \mapsto 3; \\ 3 \mapsto 3 \mapsto 2 \end{cases} \quad g \circ f = \begin{cases} 1 \mapsto 3 \mapsto 3 \\ 2 \mapsto 1 \mapsto 2. \\ 3 \mapsto 2 \mapsto 1 \end{cases}$$

$$f \circ g = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}; \quad g \circ f = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}.$$

Hence, the composition of permutations need not be commutative.

To avoid a discussion of permutations as products of transpositions, which is included in the same source, we quote the following proposition from Aigner [1] without proof.

Proposition 2.2.6 (Even Permutation)
Let f be a permutation of an n-set with cycle type $\tau(f) := t_1^{b_1(f)} \cdots t_n^{b_n(f)}$.

$$\begin{aligned} f \text{ even } &\iff b_2(f) + b_4(f) + \cdots \text{ is even} \\ &\iff n - \sum_{i=1}^{n} b_i(f) \text{ is even. } \square \end{aligned}$$

Definition 2.2.7
An *involution* is a permutation t of X that is its own inverse. Therefore

$$\forall x \in X : t \circ t(x) = x.$$

It follows that $b_n(f) = 0$ for $n > 2$.

2.3 Conventions and Miscellaneous Definitions

Conventions:

- id will always denote the identity element.
- Divisors of a natural number n are the positive divisors less than or equal n.
- Product and sums over an empty index set yield 1 and 0 respectively. That is

$$\prod_{k \in \emptyset} k = 1; \quad \sum_{k \in \emptyset} k = 0.$$

Let $v_1, \cdots, v_r$ be variables over some commutative ring with unity.

Definition 2.3.1 (Power Function)
The *power function* is defined

$$s_n(v_1, \cdots, v_r) := \sum_{i=1}^{r} v_i^n.$$

Definition 2.3.2 (Elementary Symmetric Function)
The *elementary symmetric function* is defined

$$a_n(v_1, \cdots, v_r) := \sum_{i_1 < \cdots < i_n} v_{i_1} v_{i_2} \cdots v_{i_n}.$$

The summation is performed over all $\binom{r}{n}$ possible products of n of the variables.

Example 2.3.3
Let $n = 2$ and $r = 3$. Have corresponding power function and elementary symmetric function:

$$\begin{aligned} s_2 &= \sum_{i=1}^{3} v_i^2 = v_1^2 + v_2^2 + v_3^2; \\ a_2 &= \sum_{1 \leq i_1 < j \leq 3} v_i v_j = v_1 v_2 + v_1 v_3 + v_2 v_3. \end{aligned}$$

Hence, if $n > r$, $a_n = 0$.

Definition 2.3.4 (Partition)
Let X be a finite set. A *partition* of X is a set P of disjoint subsets S_i called *parts* such that

$$\bigcup_i S_i = X.$$

The set of all partitions of X is denoted $\mathscr{P}(X)$. A partition P *refines* another Q, denoted $P \preccurlyeq Q$, if every part of P is contained within a part of Q.

3

PERMUTATION GROUPS

The first section of this chapter relates group actions and permutation groups. The second section introduces Burnside's lemma which allows us to determine the number of orbits for a given group action.

3.1 Faithful Actions as Permutation Groups

Let X be an n-set. Let α be an action on X by a group A.

Definition 3.1.1 (Set of Permutations)
The *set of permutations* of X is defined

$$S_X = \{f : X \to X \mid f \text{ bijective}\}.$$

As this solely depends on the cardinality of X, may instead notate as S_n [1].

Define id as the permutation such that $\forall x \in X : id(x) = x$. As every element of S_X is bijective, every element has an inverse that also resides in S_X. Moreover, any two elements of the set can be composed forming another element of the set. Finally, function composition is associative. Hence, S_X under function composition satisfies 2.1.1, yielding the symmetric group $(S_X, \circ)$.

Definition 3.1.2 (Permutation Group)
A *permutation group* of X is a subgroup of S_X.

We can define the left group action $\eta : S_X \times X \to X,\ (f, x) \mapsto f \cdot x$ as the "natural" action $f(x)$. A left action by a permutation group of S_X on X can be defined in the same natural way. Note that a permutation group may act on sets other than the underlying set X.

Proposition 3.1.3
The action α is equivalent to a group homomorphism from A to S_X.

Proof. Suppose α is a left group action. The proof for right group action is similar. For each $g \in A$ we have a function $\sigma_g : X \to X$ defined by $\sigma_g(x) = g \cdot x$. By 2.1.7[2.], σ_{id} is the identity function of X. For any $g, h \in A$, by 2.1.7[1.], have that $\sigma_g \circ \sigma_h = \sigma_{gh}$. Hence composition of these functions on X corresponds to the operation of elements in A. Moreover, σ_g is an invertible function since it has an inverse $\sigma_{g^{-1}}$ and $\sigma_g \circ \sigma_{g^{-1}} = \sigma_{id}$. Therefore, $\sigma_g \in S_X$ and $g \mapsto \sigma_g$ is a homomorphism $A \to S_X$.

Conversely, suppose we have a homomorphism $\beta : A \to S_X$. For each $g \in A$ we have a permutation $\beta(g)$ on X, and $\beta(gh) = \beta(g) \circ \beta(h)$. Setting $\beta(g)(x) = g \cdot x$ defines a group action of A on X, since the homomorphism properties of β yield the defining properties of a group action. □

Example 3.1.4
Let $G = (\mathbb{Z}_4, +)$ and $N = \{0, 1, 2, 3\}$. Define a left group action

$$\epsilon : G \times N \to N, \quad ([g], n) \mapsto g +_4 n.$$

Note that the identity of G is zero. This satisfies the requirements for a group action as addition is associative and $\epsilon([0], n) = n$ for each element of N. Letting $g = [2]$, we can determine how g acts on N:

$$[2] +_4 0 = 2, \quad [2] +_4 1 = 3;$$
$$[2] +_4 2 = 0, \quad [2] +_4 3 = 1.$$

Therefore, g corresponds to the permutation $\sigma_{[2]} = (0, 2)(1, 3)$.

In light of our previous discussion, α yields an associated group homomorphism $\beta : A \to S_X$. The image of β is a permutation group denoted A'. By the first isomorphism theorem, A' is isomorphic to $A/ker(\beta)$ [13]. In turn, β induces a injective group homomorphism $\gamma : A' \to S_X$, $[g]x \mapsto g \cdot x$. Due to the injective property, A' acts faithfully on the left of X. Permutation groups are used in this discussion to preclude non-faithful actions. Some common permutation groups are defined overleaf.

Definition 3.1.5 (Alternating Group)
The *alternating group* A_n is the group of even permutations of S_n and has order $\frac{n!}{2}$

Definition 3.1.6 (Cyclic & Dihedral Group)
Let the elements of X represent the vertices of a regular n-gon. The *cyclic group* C_n is the group of rotational symmetries on the vertices of said n-gon and has order n. The *dihedral group* D_n is the group of rotational and reflective symmetries on the vertices of said n-gon and has order $2n$.

A cyclic group is one that can be generated by a single element. Meaning every element in the group can be written as some power of this generating element. We introduce a fundamental theorem now for safekeeping.

Theorem 3.1.7 (Fundamental Theorem of Cyclic Groups)
Suppose $G = \langle g \rangle$ is a cyclic group of order n. Then G has the following properties:

1. *Every subgroup of G is cyclic;*
2. *The order of any subgroup of G divides n;*
3. *For any k dividing n, the subgroup $\langle g^{\frac{n}{k}} \rangle$ is a unique subgroup of order k. [3]*

Proof. Let $H \leq G$. The trivial subgroups are of course cyclic. Assume H is non-trivial and choose $g^m \in H$ with m being minimal. Clearly $\langle g^m \rangle \subseteq H$. Choose $g^k \in H$ such that $k = mq + r$ with $0 \leq r \leq m$. It follows that $r = k - qm$ and so $g^r = g^k (g^m)^{-q} \in H$. The minimal condition of m forces $r = 0$. As $id \cdot (g^m)^q = g^k$ we conclude that $g^k \in \langle g^m \rangle$. (1.)

Taking H as before, let $n = qm + r$ with $0 \leq r \leq m$. Similarly, $g^r \in H$ and the minimal condition forces $r = 0$. Therefore $n = qm$ and

$$|H| = |\langle g^m \rangle| = \frac{n}{\gcd(n, m)} = \frac{n}{m}.$$

Conclude that $m|H| = n$. (2.)

Observe that for any k dividing n we have

$$|\langle g^{\frac{n}{k}} \rangle| = |g^{\frac{n}{k}}| = \frac{n}{\gcd(n, \frac{n}{km})} = k.$$

Thus $\langle g^{\frac{n}{k}} \rangle$ is a subgroup of order k. To see this is unique let $K \leq G$ be such that $|K| = k \mid n$. By the first two parts: $H = |g^m|$ with m dividing n. Then

$$k = |K| = |\langle g^m \rangle| = \frac{n}{\gcd(m,n)} = \frac{n}{m}.$$

This yields $m = \frac{n}{k}$ and $K = \langle g^m \rangle = \langle g^{\frac{n}{k}} \rangle$. (3.) □

Example 3.1.8
Let a 3-set T represent the vertices of an equilateral triangle. Let $\sigma = \{1,2,3\}$ and $\phi = \{2,3\}$. Then $\langle \sigma \rangle = C_3 = \{id, \sigma, \sigma^2\}$ and $\langle \sigma, \phi \rangle = D_3 = \{id, \sigma, \sigma^2, \phi, \phi\sigma, \phi\sigma^2\}$.

Let G be a permutation group of S_X and consider the following relation on X.

$$x \sim_G y \iff \exists g \in G : gx = y$$

This is routinely an equivalence relation: each element of X is related to itself by the identity permutation. Suppose that $x \sim_G y$. Then there exists some g in G permuting x to y. As G is a group there must also exist an inverse that permutes y to x. Supposing further that $y \sim_G z$, there must exist some $h \in G$ permuting y to z and thus $g \circ h$ permutes x to z.

The equivalence classes are precisely the orbits of elements in X and therefore partition X. We denote the collection of these orbits by $O(X)$ as was done in the preliminaries (2.1.12).

Example 3.1.9 (Diagonal Reflection of a Square)
Let a 4-set S represent the vertices of a square positioned as illustrated. Let G be the permutation group generated by a reflection $\phi = (1,3)$. Note that $\phi^2 = id$. As G acts on S and we can thus list the orbits, stabilizers and fixed points.

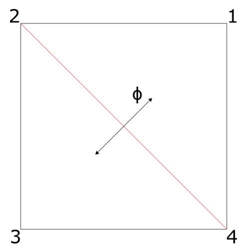


Fig. 3.1 G acting on S.

Orbits of S:	Stabilizers of S:	Fixed points of G:
$O(2) = \{2\}$;	$G(2) = \{id, \phi\} = G(4)$;	$X_{id} = \{1,2,3,4\}$;
$O(4) = \{4\}$;	$G(1) = \{id\} = G(3)$.	$X_\phi = \{2,4\}$.
$O(1) = \{1,3\} = O(3)$.		

Lemma 3.1.10 (Orbit-Stabilizer)
Let G be permutation group of S_X. Choosing some $x \in X$ we have the following relationship between its orbit and stabilizer

$$|G| = |O(x)| \cdot |G(x)|.$$

Proof. We know from 2.1.11 that $G(x) \leq G$. Thus, $G/G(x)$ is a collection of left cosets. Consider the function $\beta : O(x) \to G/G(x),\ gx \mapsto gG(x)$. To show β is well-defined: take some $gx, hx \in O(x)$, then

$$\begin{aligned} gx = hx \iff h^{-1}g \cdot x = x \iff h^{-1}g \in G(x) &\iff gG(x) = hG(x) \\ &\iff \beta(gx) = \beta(hx). \end{aligned}$$

Following the above argument in the reverse direction yields that β is injective. To show β is surjective, take some $gG(x) \in G/G(x)$. We want to find an element which maps to this under β. Clearly gx satisfies this. Conclude that β is a bijection and therefore domain and codomain have equal cardinality. We can then apply Lagrange to yield the result.

$$\begin{aligned} |O(x)| &= |G/G(x)| \\ \text{(by 2.1.6)} \quad \frac{|G|}{|G(x)|} &= |O(x)| \quad \text{as required.} \end{aligned}$$ □

Example 3.1.11 (Soccer Ball Symmetries)
Let B be the group of rotational symmetries of a soccer ball as illustrated. Each rotation of B can be seen as the permutation of the 12 black pentagons. Hence, the cardinality of B can be calculated by studying any pentagon.


Fig. 3.2 Ball consisting of black pentagons and white hexagons.

See that the count for positions a pentagon can be permuted to is 12. This is the cardinality of the orbit of said pentagon. Moreover, see that all 5 rotations of a pentagon are valid. This is the stabilizer of said pentagon. Using the above theorem we have

$$|B| = 12 * 5 = 60.$$

Alternatively, each rotation of B can be seen as the permutation of the 20 white hexagons. See that the count for positions a hexagon can be permuted to is 20. From the above equality we immediately have that the cardinality of the stabilizer of one such hexagon is 3 showing this group only contains half of rotations of a single hexagon. This is due to the others not mapping pentagons to pentagon locations.

3.2 Burnside's Lemma

The following result provides a method to count the number of orbits. It states that for a group G acting on a set X, the number of orbits generated by said action is equal to the average number of fixed points in the elements of G acting on X [13].

Theorem 3.2.1 (Burnside)
Let a group G act on a set X. Then

$$|O(X)| = \frac{1}{|G|}\sum_{g\in G} b_1(g).$$

Proof. Counting the number of pairs (g, x) such that $gx = x$ by iterating through the $g \in G$ and by iterating through the $x \in X$ we find the following relationship. The orbit-stabilizer can be applied to the latter part of the expression.

$$\sum_{g\in G} b_1(g) = |\{(g,x) \in G\times X \mid gx = x\}| = \sum_{x\in X}|G(x)|$$

$$(*)\quad \sum_{x\in X}|G(x)| = \sum_{x\in X}\frac{|G|}{|O(x)|} = |G|\sum_{x\in X}\frac{1}{|O(x)|}. \quad \text{(by 3.1.10)}$$

As the orbits partitions X into disjoint subsets

$$(**)\quad |O(X)| = \sum_{O\in O(X)} 1 = \sum_{O\in O(X)}\left[\sum_{o\in O}\frac{1}{|O|}\right] = \sum_{x\in X}\frac{1}{|O(x)|}.$$

Combining equalities $(*)$ and $(**)$ we obtain what is required

$$|O(X)| = \frac{1}{|G|}\sum_{g\in G} b_1(g). \quad \square$$

Returning to example 3.1.9, we have a group G consisting of two elements: id with 4 fixed points and ϕ with 2 fixed points. Hence, we can use the above theorem to compute what we previously observed: $|O(S)| = \frac{1}{2}(4+2) = 3$.

Example 3.2.2 (The Rook Problem)
Consider an $n \times n$ chessboard. A rook is a piece that may move any number of squares horizontally or vertically on this board. Hence, a rook threatens another if they are placed on the same row or column. Suppose we are dealing with a rook free-for-all. Can n rooks be placed on the board such that none are threatened?

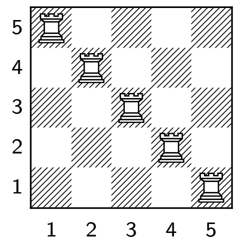


Fig. 3.3 Valid arrangement for $n = 5$.

Certainly, placing them along the n squares of the main diagonal yields a valid arrangement [7]. Furthermore, every valid arrangement is such that each rook is assigned a unique row and unique column. Hence, to find an arrangement we can iterate through the columns and place a rook in a non-occupied row. There are n choices of row in first column , $n-1$ choices of row in the second column and so on. Therefore, there are $n!$ valid configurations in total. These correspond bijectively to the $n!$ permutations of S_n [2]. The next question is more delicate. How many rook configurations exist up to symmetry of the chessboard?

Let N be an n-set. We define the chessboard N^2 as all ordered pairs of N:

$$N^2 = \{(i,j) \mid i,j \in N\}.$$

As the chessboard is square (4-gon) the symmetry group of the board is D_4. This consists of 90, 180 and 270 degree clockwise rotations denoted σ, σ^2, σ^3 along with a vertical reflection ϕ, main-diagonal reflection $\phi\sigma$, horizontal reflection $\phi\sigma^2$ and secondary-diagonal reflection $\phi\sigma^3$. Note that $\sigma^4 = \phi^2 = (\phi\sigma)^2 = (\phi\sigma^2)^2 = (\phi\sigma^3)^2 = id$. The elements of D_4 act on the squares of N^2 as shown below [2].

Identity & Rotations:

$$id(i,j) = (i,j);$$
$$\sigma(i,j) = (j, n+1-i);$$
$$\sigma^2(i,j) = (n+1-i, n+1-j);$$
$$\sigma^3(i,j) = (n+1-j, i).$$

Reflections:

$$\phi(i,j) = (i, n+1-j);$$
$$\phi\sigma(i,j) = (j,i);$$
$$\phi\sigma^2(i,j) = (n+1-i, j);$$
$$\phi\sigma^3(i,j) = (n+1-j, n+1-i).$$

Define the set of valid arrangements as

$$V = \{\Gamma_f \,|\, f \in S_n\} : \Gamma_f = \{(i, f(i)) \mid i \in N\}.$$

The elements of V are sets of n squares (i, j) from N^2. Hence, V is a subset of the power set of N^2, denoted $P(N^2)$. Thus D_4 acts on V. It is the orbit count of the latter action we wish to achieve. Therefore, need to determine $|X_g|$ for each g in D_4 acting on V.

To begin, consider σ and an arrangement $\Gamma_\pi \in X_\sigma$. Then $\sigma\Gamma_\pi = \Gamma_\pi$. Hence, for arbitrary $(i, j) \in \Gamma_\pi$ must have $\sigma(i, j) = (j, n + 1 - i) \in \Gamma_\pi$. Therefore, if the corresponding permutation $\pi \in S_n$ sends i to j (or equivalently, $\pi(i) = j$), must also have $\pi(j) = n + 1 - i$, $\pi(n + 1 - i) = n + 1 - j$ and $\pi(n + 1 - j) = i$. Hence π decomposes into cycles of maximum length 4.

$$(i, j, n + 1 - i, n + 1 - j).$$

Suppose $i = j$. Then $\pi(i) = j = i$ and $\pi(j) = n + 1 - i = i$. That is, $i = j = \frac{n+1}{2}$. This can only occur if n is odd and in this case the central cell $(\frac{n+1}{2}, \frac{n+1}{2})$ must be present in arrangement Γ_π. Now, suppose $i \neq j$ and toward contradiction that 2-cycles can occur. Then

$$\begin{aligned}
&\pi(i) = j;\\
&\pi(j) = n + 1 - i = i;\\
&\pi(n + 1 - i) = n + 1 - j = j.
\end{aligned}$$

This implies $i = \frac{n+1}{2}$ and $j = \frac{n+1}{2}$. Thus $i = j$ contradicting our initial supposition. In a similar fashion, suppose that 3-cycles can occur. This implies $n + 1 = i + j$ and in turn $\pi(j) = n + 1 - i = j$. This is certainly not a 3-cycle.

$$\begin{aligned}
&\pi(i) = j;\\
&\pi(j) = n + 1 - i;\\
&\pi(n + 1 - i) = n + 1 - j = i.
\end{aligned}$$

Conclude that X_σ is non-empty if and only if $n \pmod 4 \in \{0, 1\}$ due to each π decomposing into $\frac{n}{4}$ 4-cycles (n even) or $\frac{n-1}{4}$ 4-cycles and a fixed point (n odd).

Note that, for a 4-set $\{i, j, n+1-i, n+1-j\}$, there are two valid 4-cycles which are mutually inverse:

$$(i, j, n+1-i, n+1-j);$$
$$(j, i, n+1-j, n+1-i).$$

Let $n \pmod 4 = 0$. To count the number of ways N can be split into 4-sets of this form $\{i, j, n+1-i, n+1-j\}$ use the following algorithm. Let i be the least element of N and remove the pair $(i, n+1-i)$. Choose some $j \leq \frac{n}{2}$ remaining in N and remove the pair $(j, n+1-j)$. Repeat until N is empty.

At inception of the algorithm, $(i, n+1-i)$ is determined. Hence, $n-2$ elements remain in N, and there are $\frac{n-2}{2}$ choices for j. After this choice, $(j, n+1-j)$ is determined and $n-4$ elements remain in N. At the next iteration, $(i, n+1-i)$ is determined. Thus, there are $\frac{n-6}{2}$ choices for j. This pattern continues and we achieve count

$$\left(\frac{n-2}{2}\right)\left(\frac{n-6}{2}\right)\left(\frac{n-10}{2}\right)\cdots 1.$$

Let $n \pmod 4 = 1$. Follows that π fixes $\frac{n+1}{2}$. In this case we perform algorithm upon the set $N \setminus \{\frac{n+1}{2}\}$. Taking this alongside our previous observation concerning each set having two valid 4-cycles we conclude:

$$|X_\sigma| = \begin{cases} \displaystyle\prod_{k=0}^{t-1} [n-(2+4k)] & : \ n = 4t; \\ \displaystyle\prod_{k=0}^{t-1} [n-(3+4k)] & : \ n = 4t+1; \\ 0 & : \ \text{otherwise.} \end{cases}$$

Next, consider σ^2 and an arrangement $\Gamma_\pi \in X_{\sigma^2}$. Then $\sigma^2\Gamma_\pi = \Gamma_\pi$. If $\pi(i) = n+1-i$ then $\pi(n+1-i) = i$. That is, π decomposes into cycles of form $(i, n+1-i)$. If $i = n+1-i$ then $i = \frac{n+1}{2}$. This can only occur if n odd. Conclude that if n odd, π decomposes into $\frac{n-1}{2}$ 2-cycles and a fixed point. Alternatively, if n even, π decomposes into $\frac{n}{2}$ 2-cycles.

Let n be even. A straightforward algorithm is used to count the number of ways N can be split into 2-cycles of this form. Choose some $i \in N$ and remove the pair

$(i, n+1-i)$. Next, choose some $i \in N$ the remaining $n-2$ elements of N. Repeat this process until N is empty. In the case n is odd, π fixes $\frac{n+1}{2}$, and algorithm initiated on $N \setminus \{\frac{n+1}{2}\}$. Therefore, have counts for even and odd n respectively.

$$n(n-2)(n-4)\cdots 2;$$
$$(n-1)(n-3)(n-5)\cdots 2.$$

We may conclude:

$$|X_{\sigma^2}| = \begin{cases} \prod_{k=0}^{2t-1} [n-2k] & : \quad n = 4t, 4t+2; \\ \prod_{k=0}^{2t-1} [n-(1+2k)] & : \quad n = 4t+1, 4t+3; \end{cases}$$

Next, consider $\phi\sigma$ and an arrangement $\Gamma_\pi \in X_{\phi\sigma}$. If $\pi(i) = j$ then $\pi(j) = i$. That is, π decomposes into cycles of the form (i, j). If $i = j$ we have a fixed point. Otherwise, we have a two cycle. Hence, each involution in S_n corresponds to an arrangement in $X_{\phi\sigma}$. The theorem below aids in counting these involutions.

Theorem 3.2.3 (Involution Count)
The involution count is the number of involutions of an n-set and is denoted by i_n. It can be calculated using recurrence relation:

$$i_{n+1} = i_n + i_{n-1}. \; (i_1 = i_0 = 1)$$

Proof. Let f be an involution of X. Letting $x = n$, either $f(n) = n$ or $f(n) = y$. The first case gives rise to i_{n-1} involutions. In the second case, there are $(n-1)$ choices for y, thus an involution can be chosen for the remaining $(n-2)$ elements in i_{n-2} ways. By this argument, achieve $i_n = i_{n-1} + (n-1)i_{n-2}$. Hence, for $(n+1)$ we have the above [6].

A table of the first eight non-trivial involution counts i_n:

n	2	3	4	5	6	7	8	9	10
i_n	2	4	10	26	76	232	764	2620	9496

Accordingly, have $|X_{\phi\sigma}| = i_n$. Considering ϕ, the only arrangements that are fixed by this permutation are non-valid. Thus $|X_\phi| = 0$. Finally, every valid arrangement is fixed by id and consequently belongs to X_{id}. Thus $n! = |X_{id}|$.

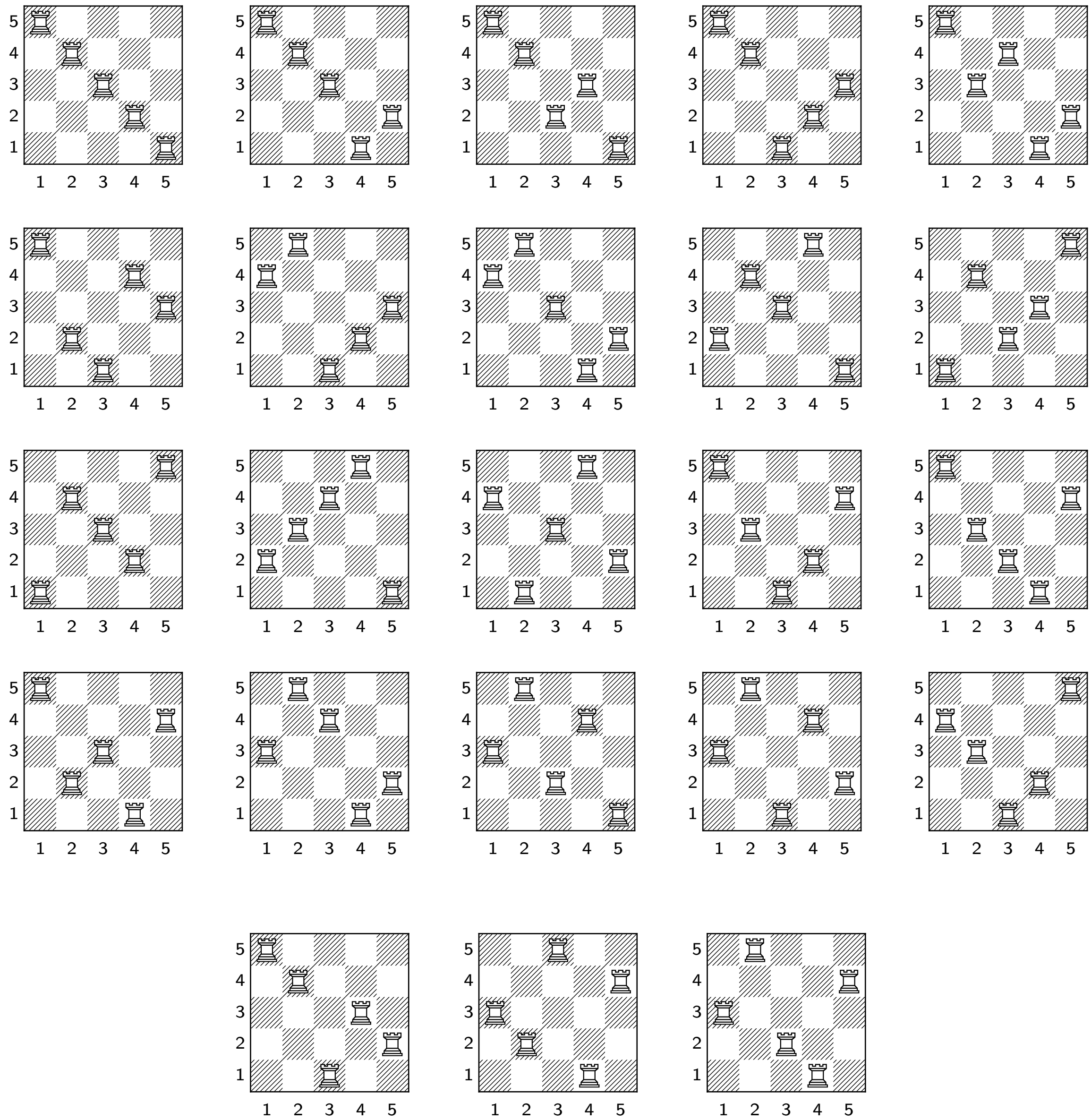
Intuitively, the diagonal reflections are equivalent and have equal number of fixed arrangements. That is $|X_{\phi\sigma}| = |X_{\phi\sigma^3}|$. Similarly, the horizontal reflection is equivalent to the vertical reflection and has no fixed arrangements. Moreover, 90 degree and 270 degree rotation are equivalent as the latter is analogous to rotating 90 degrees counter-clockwise.

Hence, can apply Burnside's Lemma for D_4 acting on V to compute the orbit count, denoted $\mathscr{R}_n$. Let $t = \lfloor \frac{n}{4} \rfloor$.

$$\mathscr{R}_n = \begin{cases} \frac{1}{8}\left(n! + 2i_n + 2\prod_{k=0}^{t-1}[n-(2+4k)] + \prod_{k=0}^{2t-1}[n-2k]\right) & : \quad n = 4t; \\ \frac{1}{8}\left(n! + 2i_n + 2\prod_{k=0}^{t-1}[n-(3+4k)] + \prod_{k=0}^{2t-1}[n-(1+2k)]\right) & : \quad n = 4t+1; \\ \frac{1}{8}\left(n! + 2i_n + \prod_{k=0}^{2t-1}[n-2k]\right) & : \quad n = 4t+2; \\ \frac{1}{8}\left(n! + 2i_n + \prod_{k=0}^{2t-1}[n-(1+2k)]\right) & : \quad n = 4t+3. \end{cases}$$

For $n = 5$ the second expression is used to calculate the number of orbits. This value corresponds to the configurations count we require. The collection of configurations are shown overleaf.

$$\begin{aligned} \mathscr{R}_5 &= \frac{1}{8}\left(5! + 2(26) + 2[5-(3+0)] + [(5-(1+0))(5-(1+2)]\right) \\ &= \frac{1}{8}\left(120 + 2 \cdot 26 + 2 \cdot 2 + 4 \cdot 2\right) \\ &= \frac{1}{8}\left(120 + 52 + 4 + 8\right) = 23. \end{aligned}$$



The growth of the configuration count as n increases is an example of a combinatorial explosion. For further n we have $\mathscr{R}_6 = 115$, $\mathscr{R}_7 = 694$ and $\mathscr{R}_8 = 5282$ [2]. With the aid of Python3 (A.0.1) calculate $\mathscr{R}_{10} = 456,454$ and $\mathscr{R}_{14} = 10,897,964,660$.

4

TRANSFORMING THE UNDERLYING SET

In the first section of this chapter, we allow transformations on the underlying set of S_X and discuss how this induces another action on the right by any permutation group of S_X. These transformations are often presented as colourings of the elements of X. The succeeding sections build upon this to derive various formulations of PET.

4.1 The Induced Action

Let G be a permutation group of S_X, where X is an n-set. Let C be an r-set.

Definition 4.1.1 (Set of Maps)
The *set of maps* from X to C is defined

$$Map(X, C) = \{f : X \to C\}.$$

The natural action of G on X induces an equivalence relation on $Map(X, C)$

$$f_1 \approx_G f_2 \iff \exists g \in G : f_1 = f_2 g.$$

The composition $f_2 g$ is defined explicitly in the remark overleaf. Showing this is an equivalence relation is done similarly as it was for the relation following 3.1.8. The equivalence classes are precisely the orbits of elements in $Map(X, C)$ generated by the induced action of G and create a partition of these elements. These orbits are called patterns and we denote their collection Ω. It is worth noting that G acts on $Map(X, C)$ yet is not a permutation group of this set. Hence, this action is not necessarily faithful and non-identity elements of G may fix every map of $Map(X, C)$.

Remark 4.1.2
The action of G on the left of X induces a right group action of G on $Map(X, C)$. This action is defined $f \cdot g = (fg)(x) = f(gx)$. To see this map is indeed a right group action: Let $f \in Map(X, C)$; $id, g, h \in G$; and $x \in X$.

$$f \cdot gh = f(gh)(x) = f(ghx) = f(gh(x)) = f(g(h(x)) = (fg)(h(x)) = (fg)(h)(x) = fg \cdot h;$$
$$f \cdot id = (f id)(x) = f(idx) = f(id(x)) = f(x) = f.$$

Satisfying 2.1.7[1.] and 2.1.7[2.] respectively.

Definition 4.1.3 (Closure)
A subset $\mathscr{F} \subseteq Map(X, C)$ is *closed* with respect to G if

$$f \in \mathscr{F}, g \in G : fg \in \mathscr{F}.$$

Conditions can be imposed on the mappings by taking a subset. However, this subset must be closed for the permutation group G to hold condition 2.1.7[1.] as a right group action. Let $\mathscr{F} \subseteq Map(X, C)$ be closed and $f \in \mathscr{F}$. Then the equivalence relation will hold on said subset and the pattern fG will also be contained in $\mathscr{F}$. Denote the collection of patterns in $\mathscr{F}$ by $\Omega_{\mathscr{F}}$.

Definition 4.1.4 (Weight Function)
Let $f \in Map(X, C)$. Assign indeterminates v_c for each $c \in C$. The *weight* of this map is defined

$$\omega(f) := \prod_{x \in X} v_{f(x)}.$$

In this form, the weight indicates the number of elements in X mapped by f to each element in C. Note that the weight function produces a product of indeterminates with subscripts $f(x)$ equal to some element in C depending on f. If two indeterminates share a subscript, we write their product as a power.

Proposition 4.1.5 (Weight of Equivalent Maps)
Let G be a permutation group inducing an action on $\mathscr{F} \subseteq Map(X, C)$. Let $f_1, f_2 \in \mathscr{F}$ be such that $f_1 \approx_G f_2$. We have that

$$\omega(f_1) = \omega(f_2).$$

Proof. By definition of the equivalence relation, exists some $g \in G$ such that $f_1 = f_2 g$. Note that g only permutes the set X and the elements iterated over for defining weight of f_1 are simply reordered whilst defining the weight of f_2. Therefore f_1 and f_2 map the same number of elements to each element in C. That is

$$\omega(f_1) = \prod_{x \in X} v_{f_1(x)} = \prod_{x \in X} v_{f_1 g(x)} = \prod_{x \in X} v_{f_2(x)} = \omega(f_2). \quad \square$$

As a pattern is an equivalence class, every map within this class has equal weight. Thus can define weight of pattern M as follows:

$$\omega(M) := \omega(f) \text{ for any } f \in M.$$

Hence, taking the weight of a representative from each pattern allows us to produce a collection of maps that are mutually inequivalent with respect to the permutation group acting.

Let G induce an action on a closed subset $\mathscr{F}$ of $Map(X, C)$. This is our system.

Definition 4.1.6 (Enumerator)
The *enumerator* $\mathbb{E}$ of our system is defined

$$\mathbb{E}(\mathscr{F}; G) := \sum_{M \in \Omega_{\mathscr{F}}} \omega(M).$$

Example 4.1.7 (Face Colouring)
Let an 8-set X represent the facets of an octahedral pendant. The lateral faces of its square pyramids are labelled left to right starting with those above the base. Let G be the group generated by a 90 degree spin about the vertical axis $\sigma = (1, 2, 3, 4)(5, 6, 7, 8)$. As G is generated by a single element, it is cyclic. In fact, G is isomorphic to C_4.

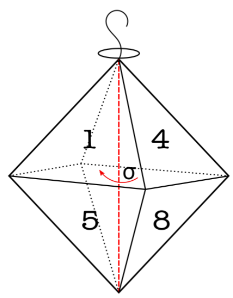


Fig. 4.1 Regular Octahedral Pendant.

Additionally, let $C = \{R, B, G\}$, in which R denotes red, B denotes blue and G denotes green. The set $Map(X, C)$ contains all face colourings of the pendant with these three colours.

Let f_1, f_2 be the colourings:

$$f_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ R & R & R & R & B & G & B & G \end{pmatrix}; \; f_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ G & R & B & R & R & G & R & B \end{pmatrix}.$$

Hence, $\omega(f_1) = \omega(f_2) = v_R^4 v_B^2 v_G^2$. These maps may have equal weight, yet do not share a pattern as there exists no valid element of G relating them.

As the dual of a regular octahedron is a cube [14], a discussion of the faces of an regular octahedron is equivalent to a discussion of the vertices of a cube. Hence, X can instead represent the vertices of a cube.

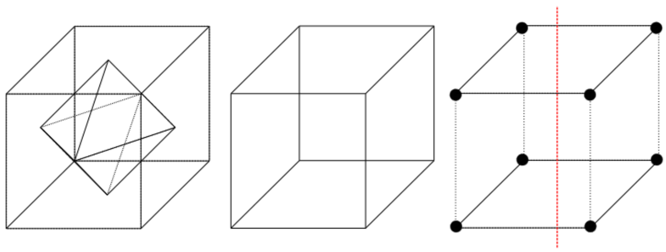


Fig. 4.2 Octahedron and Cube Duality; G Acting on Cube.

4.2 Pólya's Enumeration Theorem

We can now derive the first formulation of our main theorem. It provides a method to enumerate the patterns of our system by studying the collection of maps in $\mathscr{F}$ fixed by each permutation of G.

Theorem 4.2.1 (Pólya v1)
The enumerator of our system is given by

$$\mathbb{E}(\mathscr{F};G) = \frac{1}{|G|} \sum_{g \in G} \left(\sum_{f \in \mathscr{F},\, fg = f} w(f) \right).$$

Proof.

$$\mathbb{E}(\mathscr{F};G) = \sum_{M \in \Omega_{\mathscr{F}}} \omega(M) = \sum_{M \in \Omega_{\mathscr{F}}} \left[\sum_{f \in M} \frac{\omega(f)}{|M|} \right] = \sum_{f \in \mathscr{F}} \frac{\omega(f)}{|fG|}.$$

By 4.1.5, every map within a pattern has weight equal to the weight of said pattern. This gives the second equality. Moreover, the patterns are equivalence classes and can be represented by the maps they contain, therefore we can solely iterate through the set of mappings. This gives the third equality.

For any $f \in \mathscr{F}$, fG is its orbit by definition. Therefore

$$\begin{aligned}
\sum_{f \in \mathscr{F}} \frac{\omega(f)}{|fG|} &= \frac{1}{|G|} \sum_{f \in \mathscr{F}} w(f)|G(f)| \quad \text{(by 3.1.10)} \\
&= \frac{1}{|G|} \sum_{f \in \mathscr{F}} w(f)|\{g \in G : fg = f\}| \quad \text{(by 2.1.9)} \\
&= \frac{1}{|G|} \sum_{g \in G} \left(\sum_{f \in \mathscr{F}, fg = f} w(f) \right) \quad \text{as required.}
\end{aligned}$$

Note that stabilizer $G(f)$ is the set of G-elements that fix f when acting on the right. For last equality, the cardinality of said stabilizer, thus the term of the summation, is non-zero if and only if there exists an element in G fixing f. Hence, we can instead iterate through G, and sum the weight of the maps that are fixed at each iteration. □

Returning to 4.1.7, a vertex colouring of the cube is fixed under a rotation of G if and only if the vertex colouring of the squares perpendicular to the vertical axis are fixed under that rotation. Let $\{1, 2, 3, 4\} = A \subseteq X$ represent vertices of one such square and $\mathscr{F} = Map(A, C)$. We can find the enumerator of this subsystem using the previous result.

$$\mathbb{E} = \frac{1}{|G|} \left(\left[\sum_{f \in \mathscr{F}, fid = f} \omega(f) \right] + \left[\sum_{f \in \mathscr{F}, f\sigma = f} \omega(f) \right] + \left[\sum_{f \in \mathscr{F}, f\sigma^2 = f} \omega(f) \right] + \left[\sum_{f \in \mathscr{F}, f\sigma^3 = f} \omega(f) \right] \right).$$

As every colouring is fixed by the identity, the first term sums the weights of every possible three-colouring of a 4-set. For a colouring to be fixed by σ it must be monochromatic. The same is true for σ^3. For a colouring to be fixed by σ^2 the diagonally opposite vertices must share a colour. The fixed colourings can be found in the appendix (A.0.3).

By the previously discussed isomorphism, $|G| = 4$. Hence we have:

$$\mathbb{E}(A;G) = \frac{1}{4}(\; 4v_R^4 + 4v_B^4 + 4v_G^4 + 4v_R^3v_B + 4v_B^3v_R + 4v_R^3v_G + 4v_G^3v_R + 4v_B^3v_G + 4v_G^3v_B$$
$$+ 8v_R^2v_B^2 + 8v_R^2v_G^2 + 8v_B^2v_G^2 + 12v_R^2v_Bv_G + 12v_B^2v_Rv_G + 12v_G^2v_Rv_B \;).$$

By setting $v_c = 1$ for all $c \in C$ we obtain cardinality of the set of patterns Ω. Conclude that there are 24 configurations of three-colourings of a squares vertices up to rotation.

Following from our previous discussion we can use the data collected from this enumeration to see what vertex colourings of the cube (analogous: face colourings of the regular octahedron) are fixed under an element of G. Let $g = \sigma^2$. By A.0.3, g fixes nine colourings of the squares vertices, and consequently there are ${}^9C_2 = 36$ colourings of the cubes vertices fixed under this rotation. These calculations can be useful in identifying the cube colourings fixed under each $g \in G$.

Definition 4.2.2 (Partition Function)
The *partition function* $\rho : G \to \mathscr{P}(X)$ is defined

$$\rho(g) = \rho(\sigma_1 \cdots \sigma_k) = \{S_1, \cdots, S_k\} = \pi.$$

Moreover, the set of *G-partitions* of X is defined $\mathscr{P}(X, G) = \{\rho(g) : g \in G\}$.

Definition 4.2.3 (Kernel Partition)
Let $f \in Map(X, C)$. Define the following equivalence relation on X

$$x_1 \approx_f x_2 \iff f(x_1) = f(x_2)$$

The set of equivalence classes denoted $ker(f)$ form a partition of X.

Theorem 4.2.4
Let $g \in G$ and $f \in Map(X, C)$. We have the following relation

$$g \in G(f) \iff \rho(g) \preccurlyeq ker(f).$$

Proof. $g \in G(f)$ if and only if $fg = f$. Say g contains cycle (x_1, x_2). Then must have $f(x_1) = f(x_2)$. Therefore, x_1 and x_2 share a block in the partition $ker(f)$. Follows from equivalence relation that for any $g \in G(f) : \rho(g) \preccurlyeq ker(f)$ as required. □

Returning to 4.1.7, have $ker(f_1) = \{\{1,2,3,4\},\{5,7\},\{6,8\}\}$ and $\sigma^2 \in G(f_1)$. Also, $\rho(\sigma^2) = \{\{1,3\},\{2,4\},\{5,7\},\{6,8\}\}$. It is easy to see that $\rho(\sigma^2)$ is refined by $ker(f_1)$.

Definition 4.2.5 (Parts Function)
The *parts function* $\varphi : \mathscr{P}(X,G) \to \mathbb{N}_0$ is defined

$$\varphi(\pi) = |\{g \in G : \rho(g) = \pi\}|.$$

Theorem 4.2.6 (Pólya v2)
Let $K(\pi) = \sum_{f\in\mathscr{F},\pi\preccurlyeq ker(f)} \omega(f)$. *The enumerator of our system is given by*

$$\mathbb{E}(\mathscr{F};G) = \frac{1}{|G|}\sum_{\pi\in\mathscr{P}(X,G)} \varphi(\pi)K(\pi).$$

Proof.

$$\begin{aligned}\mathbb{E}(\mathscr{F};G) &= \frac{1}{|G|}\sum_{f\in\mathscr{F}} w(f)|G(f)| \quad \text{(by 4.2.1)}\\ &= \frac{1}{|G|}\sum_{f\in\mathscr{F}} w(f)|\{g\in G : \rho(g)\preccurlyeq ker(f)\}| \quad \text{(by 4.2.4)}\end{aligned}$$

Note that the cardinality of the stabilizer $G(f)$, and thus the term in the summation, is non-zero if and only if there exists some $g \in G$ such that $\rho(g) \preccurlyeq ker(f)$. Each element of G has some corresponding partition in $\mathscr{P}(X,G)$. Therefore, can achieve count by iterating through $\mathscr{P}(X,G)$, calculating the number of G-elements corresponding to that partition and multiplying this by the sum of the weights of maps in $\mathscr{F}$ with kernels that refine π. In other words

$$\begin{aligned}\mathbb{E}(\mathscr{F};G) &= \frac{1}{|G|}\sum_{\pi\in\mathscr{P}(X,G)} |\{g\in G : \rho(g) = \pi\}|\left[\sum_{f\in\mathscr{F},\pi\preccurlyeq ker(f)} \omega(f)\right]\\ &= \frac{1}{|G|}\sum_{\pi\in\mathscr{P}(X,G)} \varphi(\pi)K(\pi) \quad \text{as required.}\end{aligned}$$

The number of G-elements with partition equivalent to π is precisely $\varphi(\pi)$ as previously defined. Moreover, the summation of weights of f containing a partition π is precisely $K(\pi)$. Notice that $K(\pi) = 0$ if and only if there exists no $\mathscr{F}$-elements containing π and in this case the entire term of the summation is zero. □

It is worth noting that $K(\pi)$ is independent of G and $\varphi(\pi)$ is non-zero if and only if $\pi \in \mathscr{P}(X, G)$. Therefore, determining the enumerator solely depends on the cycle types of the permutations in G.

Example 4.2.7 (Returning to Necklaces)
Let an n-set X represent the set of bead positions of a necklace as in our motivating example 1.0.1. This set can be represented as the vertices of a regular n-gon. Let C be an r-set of colours and $\mathscr{F} = Map(X, C)$ be the set of necklace r-colourings. As only rotations of the necklace are considered, $G = C_n$. The collection of necklace configuration under the action of G correspond bijectively to one representative of each G-pattern of $Map(X, C)$. Denote the number of configurations $h(n, r)$. This is what we wish to compute.

By theorem 3.1.7, G has a unique subgroup $\langle \sigma^{\frac{n}{k}} \rangle$ of order k for every k dividing n. Considering such a k, via the uniqueness property, all permutations in G with order k are elements of the subgroup generated by $\sigma^{\frac{n}{k}}$.

Let $\pi = \rho(\sigma^{\frac{n}{k}})$. We have that π consists of $\frac{n}{k}$ parts of cardinality k. By our previous observations, the $g \in G$ such that $\rho(g) = \pi$ must reside within $\langle \sigma^{\frac{n}{k}} \rangle$. These are elements of the form $\sigma^{\frac{in}{k}}$ with $1 \leq i < k$ and $\gcd(i, k) = 1$. Moreover, they correspond to the generators of the subgroup [1]. Hence, the number of such i can be computed using Euler's totient function $\bar{\varphi}$.

$$\varphi(\pi) = \bar{\varphi}(k) = k \prod_{p|k} \left(1 - \frac{1}{p}\right)$$

where the p dividing k have the added condition of being prime [10]. Applying formulation 4.2.6 we achieve

$$\mathbb{E}(\mathscr{F}, C_n) = \frac{1}{n} \sum_{k|n} \bar{\varphi}(k) K(\pi).$$

Finally, setting $v_c = 1$ for all v_c present in $K(\pi)$ will yield our desired count. As at each iteration of the summation, π has $\frac{n}{k}$ parts each of cardinality k, there are r choices of colour for each part in order for π to be refined by the kernel of said colouring

$$[K(\pi)]_{v_c=1} = |\{f \in Map(X, C) \mid \pi \preccurlyeq ker(f)\}| = r^{\frac{n}{k}}.$$

Hence, our desired count is

$$h(n,r) = \frac{1}{n}\sum_{k|n} \bar{\varphi}(k) r^{\frac{n}{k}}.$$

For $n = 4$ and $r = 3$ we have:

$$h(4,3) = \frac{1}{4}\left(\bar{\varphi}(1)3^4 + \bar{\varphi}(2)3^2 + \bar{\varphi}(4)3^1\right) = \frac{1}{4}\left(1 \cdot 3^4 + 1 \cdot 3^2 + 2 \cdot 3\right) = 24.$$

Matching our configuration count following 4.2.1. Fixing $n = 4$, for further r we have the values of $h(n,r)$ shown in the first table. Similarly, fixing $r = 3$, for further n we have the values of $h(n,r)$ shown in the second table. These calculation were performed with aid of Python3 (A.0.4)

r	4	5	6	7	8	9	10
$h(4,r)$	70	165	336	616	1044	1665	2530

n	5	6	7	8	9	10	11
$h(n,3)$	51	130	315	834	2195	5934	16,107

4.3 The Cycle Indicator

The cycle indicator is a polynomial that encodes information about the cycle types of a permutation group acting on an n-set. Let s_k be the power function 2.3.1 for $k = \{1, \cdots, n\}$ over variables $\{v_1, \cdots, v_r\}$.

Theorem 4.3.1 (Pólya v3)
If $\mathscr{F} = Map(X,C)$ the enumerator of our system is given by

$$\mathbb{E}(\mathscr{F}; G) = \frac{1}{|G|}\sum_{g\in G} s_1^{b_1(g)} s_2^{b_2(g)} \cdots s_n^{b_n(g)}.$$

Proof.

$$\begin{aligned}\mathbb{E}(Map(X,C); G) &= \frac{1}{|G|}\sum_{g\in G}\left(\sum_{fg=f}\omega(f)\right) \quad \text{(by 4.2.1)}\\ &= \frac{1}{|G|}\sum_{g\in G}\left(\sum_{\rho(g)\preccurlyeq ker(f)}\omega(f)\right). \quad \text{(by 4.2.4)}\end{aligned}$$

Let $\tau(g) = t_1^{b_1} \cdots t_n^{b_n}$ and $\rho(g) = A_1, \cdots, A_m$ such that $m = \sum_{i=1}^{n} b_i(g)$. If f is constant on each part A_i (each element of A_i mapped to same element of C) - or in other words, $\rho(g) \preccurlyeq ker(f)$, then the weight of f can be expressed as

$$\omega(f) = \prod_{j=1}^{m} v_{f(A_j)}^{|A_j|} = v_{f(A_1)}^{|A_1|} \cdots v_{f(A_m)}^{|A_m|}.$$

As we consider the entire set $Map(X, C)$, we sum the weight function for every possible colouring of the parts A_i. Therefore, we sum the weight function for all possible combinations of m elements from our set C. We call one such combination an m-tuple, denoted $(k_1, \cdots, k_m)$.

$$\sum_{\rho(g) \preccurlyeq ker(f)} \omega(f) = \sum_{(k_1, \cdots, k_m)} \omega(f) = \sum_{(k_1, \cdots, k_m)} \prod_{j=1}^{m} v_{k_j}^{|A_j|}.$$

Finally, distributing across the summation and expanding the product

$$\begin{aligned}
\sum_{(k_1, \cdots, k_m)} \prod_{j=1}^{m} v_{k_j}^{|A_j|} &= \prod_{j=1}^{m} \sum_{k=1}^{r} v_k^{|A_j|} \\
&= \left(\sum_{k=1}^{r} v_k \right)^{b_1(g)} \left(\sum_{k=1}^{r} v_k^2 \right)^{b_2(g)} \cdots \left(\sum_{k=1}^{r} v_k^n \right)^{b_n(g)} \\
&= (s_1)^{b_1(g)} (s_2)^{b_1(g)} \cdots (s_k)^{b_n(g)} \quad \text{as required.} \; \square
\end{aligned}$$

Definition 4.3.2 (Cycle Indicator)
The *cycle indicator* of a permutation group G acting on an n-set, denoted $Z(G)$, is a polynomial in the variables $p_1, \cdots, p_n$ defined

$$Z(G; p_1, \cdots, p_n) := \frac{1}{|G|} \sum_{g \in G} p_1^{b_1(g)} \cdots p_n^{b_n(g)}.$$

The following formulation arises by applying the prior definition to 4.3.1.

Theorem 4.3.3 (Pólya v4)
If $\mathscr{F} = Map(X, C)$ the enumerator of our system is given by

$$\mathbb{E}(\mathscr{F}; G) = Z(G; s_1, \cdots, s_n). \; \square$$

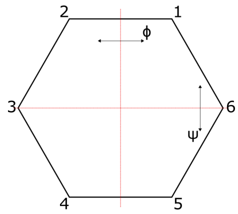


Fig. 4.3 G and H acting on X

Example 4.3.4 (Patterns of a Hexagon)
Let a 6-set X represent the vertices of a hexagon positioned as illustrated. Let permutation groups G and H be those generated by a vertical reflection ϕ and horizontal reflection ψ respectively. For $C = \{R, B, G\}$, let $\mathscr{F} = Map(X, C)$. We wish to determine $E(\mathscr{F}; G)$ and $E(\mathscr{F}; H)$ using the previous formulation.

Decomposition of G-elements and H-elements:

$$\begin{aligned} id &: (1)(2)(3)(4)(5)(6); \\ \phi &: (1,2)(3,6)(4,5); \\ \psi &: (1,5)(2,4)(3)(6). \end{aligned}$$

Clearly $b_1(id) = 6$, $b_1(\phi) = 0$, $b_2(\phi) = 3$ and $b_1(\psi) = b_2(\psi) = 2$. Moreover, $b_a(id) = b_a(\phi) = b_a(\psi) = 0$ for $a > 2$. As we have computed the cycle types of the permutation groups we can now apply 4.3.3 to determine the corresponding enumerator.

$$\begin{aligned} \mathbb{E}(\mathscr{F}; G) &= \frac{1}{|G|}\left(\left[s_1^{b_1(id)} s_2^{b_2(id)} s_3^{b_3(id)} \cdots\right] + \left[s_1^{b_1(\phi)} s_2^{b_2(\phi)} s_3^{b_3(\phi)} \cdots\right]\right) \\ &= \frac{1}{2}\left(\left[s_1^6 s_2^0 s_3^0 \cdots\right] + \left[s_1^0 s_2^3 s_3^0 \cdots\right]\right) \\ &= \frac{1}{2}\left(\left[(v_R + v_B + v_G)^6\right] + \left[(v_R^2 + v_B^2 + v_G^2)^3\right]\right). \end{aligned}$$

$$\begin{aligned} \mathbb{E}(\mathscr{F}; H) &= \frac{1}{|H|}\left(\left[s_1^{b_1(id)} s_2^{b_2(id)} s_3^{b_3(id)} \cdots\right] + \left[s_1^{b_1(\psi)} s_2^{b_2(\psi)} s_3^{b_3(\psi)} \cdots\right]\right) \\ &= \frac{1}{2}\left(\left[s_1^6 s_2^0 s_3^0 \cdots\right] + \left[s_1^2 s_2^2 s_3^0 \cdots\right]\right) \\ &= \frac{1}{2}\left(\left[(v_R + v_B + v_G)^6\right] + \left[(v_R + v_B + v_G)^2 (v_R^2 + v_B^2 + v_G^2)^2\right]\right). \end{aligned}$$

Setting $v_c = 1$ for all $c \in C$ we obtain number of configurations for the respective actions of G and H on X.

$$\mathbb{E}(\mathscr{F};G) = \frac{1}{2}\left(3^6 + 3^3\right) = \frac{1}{2}(729 + 27) = 378;$$
$$\mathbb{E}(\mathscr{F};H) = \frac{1}{2}\left(3^6 + (3^2)(3^2)\right) = \frac{1}{2}(729 + 81) = 405.$$

Although the formulation applied requires $\mathscr{F} = Map(X, C)$ one can still restrict the patterns included in the enumerator by evaluating a subset of the indeterminates. That is, if we evaluate arbitrary $v_c = 0$ and leave the rest undetermined, the enumerator will not include the pattern of any maps including v_c. This is equivalent to performing the enumeration for $\mathscr{F} = Map(X, D)$ where $D = C \setminus \{c\}$.

Lemma 4.3.5
Let $\tau(g) = t_1^{b_1(g)} \cdots t_n^{b_n(g)}$ be the cycle type of some permutation $g \in S_n$. The number of permutations in S_n with this cycle type is given as

$$\frac{n!}{\prod_{i \le n} b_i(g)!\, i^{b_i(g)}}.$$

Proof. Let $\tau(g)$ be as above. There are $n!$ ways to place the n elements in the cycle decomposition corresponding to g. Some of these placements will give rise to a new permutation and others will give rise to the same permutation.

As discussed in 2.2, the starting point of a cycle in a decomposition nor the order in which the cycles are arranged change the permutation. For any cycle of length i there are i starting points that can be selected. Moreover, there are $b_i(g)!$ ways the i-cycles can be filled without changing the permutation.

Therefore, the number of placements that give rise to a new permutation, which is equal to the number of permutations in S_n with this cycle type is given as

$$\frac{n!}{\prod_{i \le n} b_i(g)!\, i^{b_i(g)}} = \frac{n!}{b_1(g)! \cdots b_n(g)!\, 1^{b_1(g)} \cdots n^{b_n(g)}} \quad \square$$

This section concludes with presentation of the cycle indicator of common permutation groups acting naturally on the underlying set that the permutations are defined upon. Let $(b_1, \cdots, b_n)$ denote the n-tuple of cycles length $k \in \{1, \cdots, n\}$ for a permutation in S_n. Note that $\sum_i^n ib_i = n$ for any valid n-tuple.

Proposition 4.3.6 (Cycle Indicator of Common Permutation Groups)
Let S_n, A_n, C_n and D_n act naturally on the underlying n-set. We have

1. $Z(S_n) = \sum_{(b_1,\cdots,bn)} \frac{1}{b_1!\cdots b_n!\,1^{b_1}\cdots n^{b_n}} p_1^{b_1}\cdots p_n^{b_n}$;

2. $Z(A_n) = \sum_{(b_1,\cdots,bn)} \frac{1+(-1)^{n-\sum_{i=1}^n b_i}}{b_1!\cdots b_n!\,1^{b_1}\cdots n^{b_n}} p_1^{b_1}\cdots p_n^{b_n}$;

3. $Z(C_n) = \frac{1}{n}\sum_{k|n} \bar{\varphi}(k) p_k^{\frac{n}{k}}$;

4. $Z(D_n) = \frac{1}{2}Z(C_n) + \begin{cases} \frac{1}{4}p_1^2 p_2^{\frac{(n-2)}{2}} + \frac{1}{4}p_2^{\frac{n}{2}} : & n \text{ even;} \\ \frac{1}{2}p_1 p_2^{\frac{(n-1)}{2}} : & n \text{ odd.} \end{cases}$

Proof. By definition of the cycle indicator 4.3.2 we have

$$Z(S_n) = \frac{1}{n!}\sum_{g\in S_n} p_1^{b_1(g)}\cdots p_n^{b_n(g)}.$$

Iterating over the permutation of S_n is equivalent to iterating over the possible cycle types, and multiplying by the number of permutations with that cycle type. This can be calculated using 4.3.5. Distributing the fraction $\frac{1}{n!}$ across the summation we achieve (1.).

$$Z(S_n) = \sum_{(b_1,\cdots,b_n)} \frac{n!}{n!\,b_1!\cdots b_n!\,1^{b_1}\cdots n^{b_n}} p_1^{b_1}\cdots p_n^{b_n} = \sum_{(b_1,\cdots,b_n)} \frac{1}{b_1!\cdots b_n!\,1^{b_1}\cdots n^{b_n}} p_1^{b_1}\cdots p_n^{b_n}.$$

Let $f : \mathbb{N} \to \{0, 1\}$ be defined $f(x) = \frac{1+(-1)^x}{2}$. There are two cases that can occur:

$$f(x) = \begin{cases} 1 & \text{if } x \text{ even;} \\ 0 & \text{if } x \text{ odd.} \end{cases}$$

By definition of the cycle indicator 4.3.2 we have

$$Z(A_n) = \frac{2}{n!}\sum_{g\in A_n} p_1^{b_1(g)}\cdots p_n^{b_n(g)}.$$

Iterating over the even permutation A_n is equivalent to iterating over the permutations of S_n and setting the term of the summation to zero if the permutation is odd. By 2.2.6, we can check a permutation is odd by considering its cycle type.

This proposition states that a permutation g, with cycle type $\tau(g) = t_1^{b_1(g)} \cdots t_n^{b_n(g)}$ is an even permutation if and only if $n - \sum_{i=1}^{n} b_i$ is even. It follows directly that a permutation is odd if and only if $n - \sum_{i=1}^{n} b_i$ is odd. Let $(b_1, \cdots, b_n)$ be a possible cycle type of S_n. Evaluating f at $n - \sum b_i$ will thus return 0 if this is a cycle type corresponding to odd permutations and return 1 if this is a cycle type corresponding to even permutations.

Hence, multiplying each term of the summation over all possible cycle types (which is equivalent to summing over all the permutations in S_n) with $f(n - \sum_{i=1}^{n} b_i)$ will set the terms corresponding to odd permutations to zero, while leaving the even permutation terms unaffected.

$$\begin{aligned}
Z(A_n) &= \frac{2}{n!} \sum_{(b_1,\cdots,b_n)} f(n - \sum_{i=1}^{n} b_i) \frac{n!}{b_1! \cdots b_n! \, 1^{b_1} \cdots n^{b_n}} p_1^{b_1} \cdots p_n^{b_n} \\
&= \frac{2}{n!} \sum_{(b_1,\cdots,b_n)} \frac{1 + (-1)^{n - \sum_{i=1}^{n} b_i}}{2} \frac{n!}{b_1! \cdots b_n! \, 1^{b_1} \cdots n^{b_n}} p_1^{b_1} \cdots p_n^{b_n}.
\end{aligned}$$

Distributing the fraction $\frac{2}{n!}$ across the summation we achieve (2.).

$$\begin{aligned}
Z(A_n) &= \sum_{(b_1,\cdots,b_n)} \frac{2}{n!} \frac{1 + (-1)^{n - \sum_{i=1}^{n} b_i}}{2} \frac{n!}{b_1! \cdots b_n! \, 1^{b_1} \cdots n^{b_n}} p_1^{b_1} \cdots p_n^{b_n} \\
&= \sum_{(b_1,\cdots,b_n)} \frac{1 + (-1)^{n - \sum_{i=1}^{n} b_i}}{b_1! \cdots b_n! \, 1^{b_1} \cdots n^{b_n}} p_1^{b_1} \cdots p_n^{b_n}.
\end{aligned}$$

By definition of the cycle indicator 4.3.2 we have

$$Z(C_n) = \frac{1}{n} \sum_{g \in C_n} p_1^{b_1(g)} \cdots p_n^{b_n(g)}$$

We know from 3.1.7 and our previous discussion in 4.2.7 that every element of C_n consists of $\frac{n}{k}$ k-cycles for some k dividing n. We can count the number of elements of this form for each k dividing n using Euler's totient function $\bar{\varphi}$.

Therefore, iterating over the elements of C_n is equivalent to iterating over the divisors of n and multiplying by $\bar{\varphi}$ (3.).

$$Z(C_n) = \frac{1}{n} \sum_{k|n} \bar{\varphi}(k) p_k^{\frac{n}{k}}.$$

By definition of the cycle indicator 4.3.2 we have

$$Z(D_n) = \frac{1}{2n} \sum_{g \in D_n} p_1^{b_1(g)} \cdots p_n^{b_n(g)}$$

By 3.1.6, we can treat D_n as the group of rotation and reflection permutations of a regular n-gon. The summation over the permutations of the group can thus be split into summation over rotations added to the summation over reflections. These rotations are precisely the elements of C_n. Therefore

$$\begin{aligned}
Z(D_n) &= \frac{1}{2n} \left[\sum_{g \in C_n} p_1^{b_1(g)} \cdots p_n^{b_n(g)} + \sum_{g \in D_n \setminus C_n} p_1^{b_1(g)} \cdots p_n^{b_n(g)} \right] \\
&= \frac{1}{2n} \left[n \sum_{k|n} \varphi(\bar{k})) p_k^{\frac{n}{k}} + \sum_{g \in D_n \setminus C_n} p_1^{b_1(g)} \cdots p_n^{b_n(g)} \right] \quad \text{(by 3.)} \\
&= \frac{1}{2n} \left[n Z(C_n) + \sum_{g \in D_n \setminus C_n} p_1^{b_1(g)} \cdots p_n^{b_n(g)} \right].
\end{aligned}$$

The reflections of an n-gon depend on if n is even or odd. In the even case, there are two reflection "shapes" that can occur: A reflection that passes through each pair of opposite faces and a reflection that passes through each pair of opposite vertices. There are $\frac{n}{2}$ many of each pair. The cycle type for the first shape is $p_2^{n/2}$ as it permutes $\frac{n}{2}$ pairs of vertices. The cycle type for the second shape is $p_1^2 p_2^{(n-2)/2}$ as it fixes the two vertices it passes through and permutes the remaining $\frac{(n-2)}{2}$ pairs of vertices.

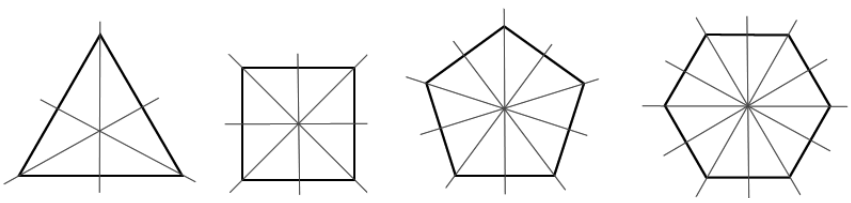


Fig. 4.4 Reflections of various n-gons

In the odd case, a reflection passes through each of the n vertices and it's opposing face. The cycle type for this reflection is $p_1 p_2^{(n-1)/2}$ as it fixes one vertex and permutes the remaining $\frac{(n-1)}{2}$ vertices. These ideas are easily verifiable through visualising regular polygons (see fig 4.4).

Thus, we have the following polynomials for the summation over $D_n \setminus C_n$:

$$\sum_{g \in D_n \setminus C_n} p_1^{b_1(g)} \cdots p_n^{b_n(g)} = \begin{cases} \frac{n}{2} p_1^2 p_2^{\frac{(n-2)}{2}} + \frac{n}{2} p_2^{\frac{n}{2}} : & n \text{ even;} \\ n p_1 p_2^{\frac{(n-1)}{2}} : & n \text{ odd.} \end{cases}$$

Substituting this into our expression for $Z(D_n)$ and rearranging we achieve (4.).

$$\begin{aligned} Z(D_n) &= \frac{1}{2n} n Z(C_n) + \frac{1}{2n} \sum_{g \in D_n \setminus C_n} p_1^{b_1(g)} \cdots p_n^{b_n(g)} \\ &= \frac{1}{2n} n Z(C_n) + \frac{1}{2n} \cdot \begin{cases} \frac{n}{2} p_1^2 p_2^{\frac{(n-2)}{2}} + \frac{n}{2} p_2^{\frac{n}{2}} : & n \text{ even;} \\ n p_1 p_2^{\frac{(n-1)}{2}} : & n \text{ odd.} \end{cases} \\ &= \frac{1}{2} Z(C_n) + \begin{cases} \frac{1}{4} p_1^2 p_2^{\frac{(n-2)}{2}} + \frac{1}{4} p_2^{\frac{n}{2}} : & n \text{ even;} \\ \frac{1}{2} p_1 p_2^{\frac{(n-1)}{2}} : & n \text{ odd.} \end{cases} \qquad \square \end{aligned}$$

To see that the cycle indicator depends upon the group action, let X be an 8-set and consider the permutation $\sigma = (1, 2, 3, 4)(5, 6)(7, 8)$ of X. The permutation group $G = \langle \sigma \rangle$ is isomorphic to the cyclic group C_4.

However, the action α of C_4 on the vertices of a regular 4-gon and the natural action η of G on X are two different actions, and thus have differing cycle indicators. To derive the cycle indicator for action α we apply 4.3.6[3.], yielding

$$Z_\alpha(C_4) = \frac{1}{4} p_1^4 + \frac{1}{4} p_2^2 + \frac{1}{2} p_4.$$

By definition of the cycle indicator 4.3.2, for action η we have

$$Z_\eta(G) = \frac{1}{4} p_1^8 + \frac{1}{4} p_1^4 p_2 + \frac{1}{2} p_2^2 p_4.$$

4.4 Injective and Surjective Transformations

In the first section we discussed restricting the set of maps under consideration to a closed subset. This section includes formulations of PET for two special restriction cases: restriction to injective maps and restrictions to surjective maps. Let a_n be the elementary symmetric function 2.3.2 over variables $\{v_1, \cdots, v_r\}$.

For $Map(X, C)$ to contain surjective maps, the cardinality of X must be at least the cardinality of C. These are the maps for which every element of C is assigned at least one element in X. Denote the set of surjective maps as $Sur(X, C)$.

On the other hand, for $Map(X, C)$ to contain injective maps, the cardinality of X must be at most the cardinality of C. This is due to injective maps requiring each element of X to be assigned a unique element of C. Denote the set of injective maps $Inj(X, C)$. Note that $Inj(X, C)$ is the set of bijective maps if and only if $n = r$.

Proposition 4.4.1 (Injective Pólya)
If $\mathcal{F} = Inj(X, C)$ the enumerator of our system is given by

$$\mathbb{E}(\mathcal{F}; G) = \frac{n!}{|G|} a_n(v_1, \cdots, v_r).$$

Proof. By 4.2.1 we have the first equality. Recall that $G(f)$ is the stabilizer of map f. Note that any the only element of G fixing any injective map is the identity. Hence we get the second equality.

$$\begin{aligned}\mathbb{E}(Inj(X, C); G) &= \frac{1}{|G|} \sum_{f \in \mathcal{F}} \omega(f)|G(f)| \\ &= \frac{1}{|G|} \sum_{f \in \mathcal{F}} \omega(f).\end{aligned}$$

Recalling the definition of the weight function 4.1.4: $\omega(f) = \prod_{x \in X} v_{f(x)}$. The weight function for any $f \in Inj(X, C)$ contains n indeterminates each with a unique subscript from C. Iterating over the maps of $\mathcal{F}$ is equivalent to iterating over the length n indeterminate monomials that are unique up to reordering and multiplying by the number of ways each can be reordered. The list of monomials with n unique subscripts from C can be enumerated by $a_n(v_1, \cdots, v_r)$.

There are $n!$ ways of rearranging the indeterminates of each monomial of this summation and hence $n!$ term can be factored out. The above discussion yields the equality $\sum_{f\in\mathscr{F}} \omega(f) = n!\, a_n(v_1, \cdots, v_r)$. Hence, we have the following expression for the enumerator

$$\mathbb{E}(Inj(X,C);G) = \frac{1}{|G|}\sum_{f\in\mathscr{F}} \omega(f) = \frac{1}{G}\left[n!\, a_n(v_1,\cdots,v_r)\right].$$
$$= \frac{n!}{G} a_n(v_1,\cdots,v_r). \quad \square$$

For the closed subset $Sur(X,C)$ the formula is more intricate and deriving it requires a discussion of the inclusion-exclusion principle. This counting technique and the corresponding formula is presented by Aigner [1]. We quote the result without proof. This formula removes any monomials that do not contain at least one of each indeterminate v_c for all $c \in C$.

Proposition 4.4.2 (Surjective Pólya)
If $\mathscr{F} = Sur(X,C)$ the enumerator of our system is given by

$$\begin{aligned}\mathbb{E}(\mathscr{F};G) = &\frac{1}{|G|}\sum_{g\in G}\Bigg(s_1^{b_1(g)}\cdots s_n^{b_n(g)} - \sum_{i\in C}(s_1 - v_i)^{b_1(g)}\cdots(s_n - v_i^n)^{b_n(g)} \\ &+ \sum_{i\in C\setminus\{j\}}(s_1 - v_i - v_j)^{b_1(g)}\cdots(s_n - v_i^n - v_j^n)^{b_n(g)} \\ &- \sum_{i\in C\setminus\{j,k\}}(s_1 - v_i - v_k)^{b_1(g)}\cdots(s_n - v_i^n - v_k^n)^{b_n(g)} \pm \cdots\Bigg)\end{aligned}$$

where one element is removed from the set C at each inner summation after the first until it is empty. This removed element is then used within the summation it was removed at (see v_j and v_k terms above). $\square$

To see how to this formula may be used lets look at a straightforward example.

Example 4.4.3 (Enumerating the Surjective Colourings of a Triangle)
Let X be a 3-set and C be a 3-set denoted $\{R, B, G\}$. Let $\sigma = (1,2,3)$ and $C_3 = \langle\sigma\rangle$. Permutation group C_3 induces an action on $Map(X,C)$. Letting $\mathscr{F} = Sur(X,C)$, we may compute the enumerator of this system using the proposition above. Note that the cycle type of σ and σ^2 are equal and thus so are their corresponding terms in the formula.

Expanding the summations and applying the definition of the power function 2.3.1 we achieve

$$\begin{aligned}\mathbb{E}(\mathscr{F};G) = \frac{1}{3}\Bigg\{ &\left[s_1^3 - \sum_{i\in\{R,B,G\}} (s_1 - v_i)^3 + \sum_{i\in\{R,B\}} (s_1 - v_i - v_G)^3 - \sum_{i\in\{R\}} (s_1 - v_i - v_B)^3 \right] \\ &+ 2\left[s_3 - \sum_{i\in\{R,B,G\}} (s_3 - v_i^3) + \sum_{i\in\{R,B\}} (s_3 - v_i^3 - v_G^3) - \sum_{i\in\{R\}} (s_3 - v_i^3 - v_B^3) \right] \Bigg\}\end{aligned}$$

$$\begin{aligned}&= \frac{1}{3}\left[(v_R + v_B + v_G)^3 - (v_B + v_G)^3 - (v_R + v_G)^3 - (v_R + v_B)^2 + (v_B)^3 + (v_R)^3 - (v_G)^3\right] \\ &+ \frac{2}{3}\left[(v_R^3 + v_B^3 + v_G^3) - (v_B^3 + v_G^3) - (v_R^3 + v_G^3) - (v_R^3 + v_B^3) + (v_B^3) + (v_R^3) - (v_G^3)\right]\end{aligned}$$

$$= \frac{1}{3}\left[6v_1v_2v_3\right] + \frac{2}{3}[0] = 2v_1v_2v_3.$$

The set $Sur(X,C)$ is analogous to the six ways three colours can be placed on the vertices of an equilateral triangle. These may be represented as the words

$$RBG, GRB, BGR, RGB, BRG, GBR$$

Our enumeration tells us that there are two configurations up to rotation, with pattern representatives RBG and RGB. If we had instead chosen D_3 as our permutation group, these two words would be equivalent under a reflection and our enumerator would have contained a single configuration. That is, all elements of $Sur(X,C)$ are equivalent under the permutations of D_3.

5

APPLICATIONS AND PHENOMENA

This chapter begins with a discussion bridging Burnside's lemma and Pólya's enumeration theorem. After this, applications of Pólya theory to organic chemistry and counting configurations of twisty puzzles are considered.

5.1 Subsets and Two-Colourings

Pólya's enumeration theorem is often said to be a generalisation of Burnside's lemma. The first version of PET (4.2.1) being the closest in formulaic resemblance to Burnside (3.2.1). In our motivating example 1.0.1 we seen an example equivalent to listing the configurations for two-colourings of a 4-set under C_4.

As before, denote this 4-set X and the set of colours $C = \{R, B\}$. A two-colouring can be denoted by a subset S of X. Without loss of generality, assign the elements of S with R and the element of $X \setminus S$ with B.

Suppose $S = \{1, 3, 4\}$, then the corresponding necklace is represented by the word $RBRR$. This necklace can then be permuted by some $\sigma \in C_4$ by applying said permutation to S. In the case g is a 90 degree clockwise rotation:

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

Hence, $\sigma S = BRRR$. These two subsets share an orbit due to the existence of σ. The set of all possible subsets of X is its power set $P(X)$. The orbit count of the action of C_4 on $P(X)$ can be studied by applying Burnside. This calculation is equivalent to determining the enumerator of the action of C_4 on $Map(X, C)$ and evaluating $v_c = 1$ for all $c \in C$ using PET.

We previously observed this phenomenon in the rook problem 3.2.2. In this problem, the set of valid arrangements V is a subset of the power set of the chessboard $P(N^2)$. This set of sets V contain all cardinality five subsets of the squares of the board with the added condition that these squares belong to mutually distinct rows and mutually distinct columns.

Let $\Gamma \in V$. If a square is contained in valid arrangement Γ, a rook is placed on it. Otherwise, the square is left unoccupied. By our previous discussion, a valid arrangement is equivalent to a two-colouring of the chessboard. Moreover, the set V is equivalent to $\mathscr{F} \subseteq Map(X, C)$, where C is a 2-set of colours and $\mathscr{F}$ is the set of two-colourings of the n squares of the board with one colour, respecting the aforementioned added condition, and the remaining $n^2 - n$ with the other colour.

Thus the orbit count of the action of the boards symmetry group D_4 on V calculated in this example is equivalent to determining the enumerator of the action of D_4 on $\mathscr{F}$ and evaluating $v_c = 1$ for all $c \in C$ using PET.

When X is an n-set C is a r-set for $r > 2$, the subsets in $P(X)$ cannot encode information about r states of an element of X. This is the general case that exhibits the true power of Pólya's enumeration theorem. For a finite number of states, a transformation $f : X \to C$ can assign each elements of X to a state in C. Then the configurations of a selection of state assignments under a given group action can be enumerated using PET.

5.2 Enumerating Chemical Compounds

Benzene is a organic chemical compound with molecular formula C_6H_6. It is made up of six carbon atoms and six hydrogen atoms that all lie on a single plane. The carbon atoms are bonded together in a cyclic nature forming a ring. Each hydrogen atom is attached to a unique carbon atom of the ring by a single bond.

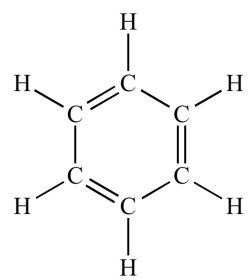


Fig. 5.1 Benzene ring

The hydrogen atoms can be substituted for halogen atoms to yield different compounds. A compound yielded via a substitution is called a benzene derivative. Note that the carbon ring is always present in a benzene derivative, and thus every molecular formula of a benzene derivative will contain C_6. An example of a benzene derivative is dichlorobenzene $C_6H_4Cl_2$, which is the result of substituting two chlorine atoms. This substitution can be done in three ways.

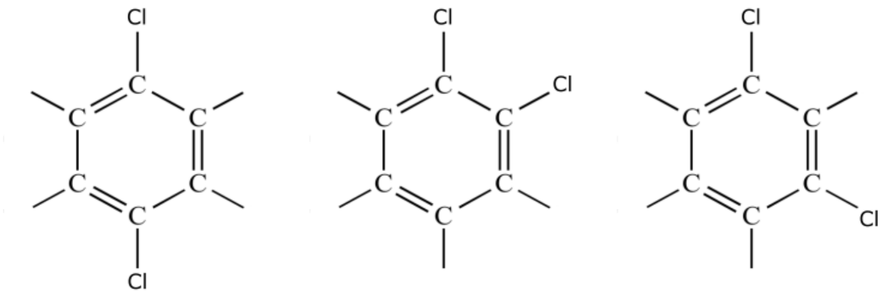


Fig. 5.2 Isomers of dichlorobenzene. Empty label denoting presence of hydrogen atom.

These are referred to as the isomers of dichlorobenzene. Any other arrangement of the three chlorine and four hydrogen atoms is equivalent to one of these under symmetry of the compounds structure (D_6). The isomers of benzene derivatives yielded by substituting chlorine and bromine atoms can be enumerated using PET.

Let a 6-set X represent the positions of the hydrogen atoms in the benzene ring. Let $C = \{H, Cl, Br\}$ be our set of atoms. Then $\mathcal{F} = Map(X, C)$ the set of all possible substitutions of atoms into the positions of X. The enumerator of this system is given by 4.3.3 and 4.3.6[4.]:

$$\begin{aligned}\mathbb{E}(\mathcal{F}; D_6) &= Z(D_6; s_1, \cdots, s_6)\\ &= \frac{1}{2}\Big(\frac{1}{6}\sum_{k|6}\bar{\varphi}(k)s_k^{\frac{6}{k}}\Big) + \frac{1}{4}(s_2^3 + s_1^2 s_2^2).\end{aligned}$$

Expanding this we achieve:

$$\begin{aligned}\mathbb{E} &= \frac{1}{12}(s_1^6 + s_2^3 + 2s_3^2 + 2s_6) + \frac{1}{4}(s_2^3 + s_1^2 s_2^2)\\ &= \frac{1}{12}([v_H + v_{Cl} + v_{Br}]^6 + [v_H^2 + v_{Cl}^2 + v_{Br}^2]^3 + 2[v_H^3 + v_{Cl}^3 + v_{Br}^3]^2 + 2[v_H^6 + v_{Cl}^6 + v_{Br}^6])\\ &\quad + \frac{1}{4}([v_H^2 + v_{Cl}^2 + v_{Br}^2]^3 + [v_H + v_{Cl} + v_{Br}]^2[v_H^2 + v_{Cl}^2 + v_{Br}^2]^2)\\ &= v_H^6 + v_H^5 v_{Cl} + v_H^5 v_{Br} + 3v_H^4 v_{Cl}^2 + 3v_H^4 v_{Cl} v_{Br} + 3v_H^4 v_{Br}^2 + 3v_H^3 v_{Cl}^3 + 6v_H^3 v_{Cl}^2 v_{Br}\\ &\quad + 6v_H^3 v_{Cl} v_{Br}^2 + 3v_H^3 v_{Br}^3 + 3v_H^2 v_{Cl}^4 + 6v_H^2 v_{Cl}^3 v_{Br} + 11v_H^2 v_{Cl}^2 v_{Br}^2 + 6v_H^2 v_{Cl} v_{Br}^3\\ &\quad + 3v_H^2 v_{Br}^4 + v_H v_{Cl}^5 + 3v_H v_{Cl}^4 v_{Br} + 6v_H v_{Cl}^3 v_{Br}^2 + 6v_H v_{Cl}^2 v_{Br}^3 + 3v_H v_{Cl} v_{Br}^4\\ &\quad + v_H v_{Br}^5 + v_{Cl}^6 + v_{Cl}^5 v_{Br} + 3v_{Cl}^4 v_{Br}^2 + 3v_{Cl}^3 v_{Br}^3 + 3v_{Cl}^2 v_{Br}^4 + v_{Cl} v_{Br}^5 + v_{Br}^6.\end{aligned}$$

The indeterminates of each term in the above summation represents a benzene derivative and their coefficients tell us the number of isomers for that derivative. For example, $v_H^3 v_{Cl} v_{Br} \equiv C_6H_3BrCl_2$ represents bromodichlorobenzene which has the following collection of isomers:

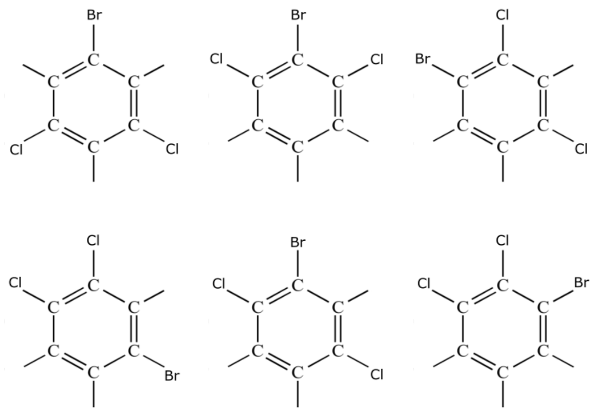


Fig. 5.3 The six isomers of bromodichlorobenzene. Empty label denoting presence of hydrogen atom.

Conclude that there are 27 benzene derivatives that can be made by substituting a combination of chlorine and bromine atoms. This example is equivalent to enumerating all three-colourings of a hexagons vertices under the action of D_6.

5.3 Cube Counting

The Rubik's cube is a puzzle invented in 1974 by Hungarian designer Ernő Rubik. It consists of 54 facets. Twisting one of the cubes faces rotates the 20 facets attached to it around its central facet. Cube enthusiasts frequently use a notation developed by American-British mathematician David Singmaster. A cube move is a 90 degree clockwise rotation of a particular face and the faces are denoted as illustrated. Therefore, the set of cube moves is written as

$$M = \{U, D, F, B, L, R\}.$$

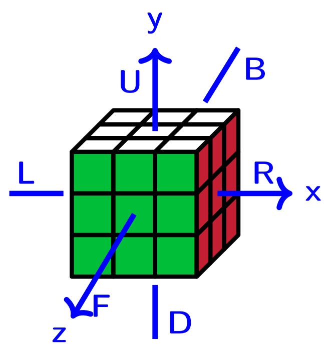
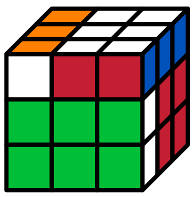


Fig. 5.4 Singmaster Face Notation

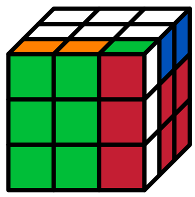
The central facets are fixed by every cube move [11]. Each non-central facet is attached to a sub-cube or "cubie". Cubies come in two forms: corner cubies (with three facets) and edge cubies (with two facets). There are eight corner cubies and twelve edge cubies in this cube. Note that in our diagrams the cube is endowed with the standard solved colouring. This is done to ease understanding of diagrams. However, when we define the cube as a finite-set, the cube is colourless.

Let X be a 54-set representing the facets of the cube. The cube moves of M are precisely permutations of these facets by way of permuting the corresponding cubies. Due to the physical restriction of permuting facets by way of the cubies, certain permutations of the facets cannot occur without disassembling the cube.

Applying any cube move four times yields the identity permutation. The inverse of a cube move is a 90 degree counterclockwise rotation of the appropriate face and can be yielded by performing said cube move three times. Taking the set M along with function composition forms a permutation group of S_X. It is easy to verify through counterexample that M is not commutative:



$= FU \neq UF =$



.

Each element of permutation group M corresponds to a sequence of cube moves. Hence, M is in bijective correspondence with the set of 'legal' facet arrangements of the cube. This leads naturally to the question: What is the cardinality of M?

By labelling the facets of the cube (A.0.5) with elements of the set $\{1, \cdots, 54\}$ and defining each cube move as a permutation (A.0.6) of this set, we can use GAP to generate the group M (A.0.7) and compute $|M| = 43,252,003,274,489,856,000$. This aligns with the cardinality presented in *Adventures in group theory* [5].

This source also provides a derivation of the structure of M but requires a large investment of material. To balance this, we will forgo much of the rigour and motivation for construction of the objects. Joyner shows that M is an example of a semi-direct product of two subgroups N, L of M. These subgroups are defined overleaf. He goes further to show that these subgroups can be represented as the product of common permutation groups. This derivation will be overlooked, but the result is as follows.

$$M = N \rtimes L = (C_3^7 \times C_2^{11}) \rtimes [(A_8 \times A_{12}) \rtimes C_2].$$

Definition 5.3.1 (Inner Semi-Direct Product)
A group G is an inner *semi-direct product* if there exists $H, K \leq G$ satisfying:

1. H is a normal subgroup of G;
2. $G = HK = \{hk \mid h \in H, k \in K\}$;
3. $H \cap K = \{\text{id}\}$.

If these conditions are satisfied, we write $G = H \rtimes K$.

Subgroup N is constructed by taking the normal closure of a subset O of M.

Definition 5.3.2 (Normal Closure)
Let G be a group and let $A \subseteq G$. The *normal closure* of A is the intersection of all normal subgroups of G containing A and is a normal subgroup of G.

$$nc(A) = \bigcap_{N \trianglelefteq G} N.$$

The subset O is defined below. The normal closure N can be thought of as the smallest normal subgroup of M containing O. This is sometimes referred to as the normal subgroup of M generated by O. By definition of normal closure, N is normal.

$$O = \{RUDB^2U^2B^{-1}UBUB^2D^{-1}R^{-1}U^{-1}, BR^{-1}D^2RB^{-1}U^2BR^{-1}D^2RB^{-1}U^2\} \text{ [15]}.$$

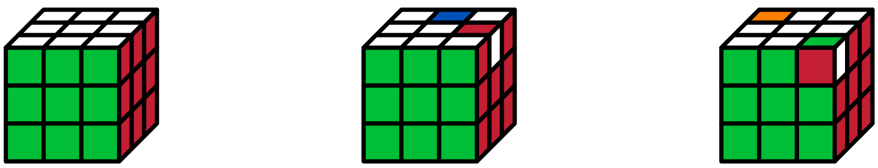


Fig. 5.5 Solved rubik's cube and where it is permuted by O.

The other subgroup L is generated by the subset P of M defined.

$$P = \{U^2, D^2, F, B, R^2, L^2, R^2U^{-1}FB^{-1}R^2F^{-1}BU^{-1}R^2\} \text{ [15]}.$$

Applying a permutation of N composed with a permutation of L is of course a some permutation of M. Constructing these subgroups in GAP (A.0.8), we note that the cardinality of NL is equal to the cardinality of M and their intersection is the identity. We calculate that:

$$|N| = 4,478,976;$$
$$|L| = 9,656,672,256,000;$$
$$|N| \cdot |L| = 43,252,003,274,489,856,000 = |M|.$$

Hence, N and L satisfy the properties of 5.3.1 and $M = N \rtimes L$.

Fig. 5.6 Solved rubik's cube and where it is permuted by last listed element of P.

In theory, we can use our construction of M in GAP to list the cycle types of its permutations and use them to compute the enumerator of r-colourings of the cubes facets up to the action of M using Pólya's enumeration theorem. However, these numbers are too large for any currently accessible computer to handle. Towards compromise, let's look at a smaller puzzle.

The Pocket cube was invented in 1970 by American designer Larry D. Nichols. It consists of 24 facets. Twisting one of the cubes faces rotates the 12 facets attached to it. We can again use Singmaster notation to denote a sequence of twists. The cube moves are again written as

$$M = \{U, D, F, B, L, R\}.$$

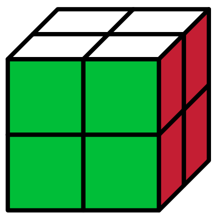


Fig. 5.7 Pocket cube

The pocket cube does not have any facets fixed by M. In the previous discussion the central facets acted as an 'anchor' to keep the Rubik's cube fixed under three-dimensional rotational symmetry. For the pocket cube, some series of cubes moves cause the cube to rotate in three-dimensional space. For example, applying R^3L, causes a cycle of the faces (U, B, D, F) to occur. To fix the pocket cube under the same symmetry, we exclude any cube move that effects the bottom-left cubie of the front face. This is done by restricting the cube moves considered to the set

$$Q = \{U, B, R\}.$$

Let Y be a 24-set representing the facets of the pocket cube. Again, the cube moves of Q are precisely permutation of these facets and $(Q, \circ)$ is a permutation group of S_Y. Moreover, Q is in bijective correspondence with the set of 'legal' facet arrangements of the pocket cube.

By labelling the facets of the pocket cube (A.0.9) with elements of the set $\{1, \cdots, 24\}$ and defining the considered cube moves as permutations (A.0.10) of the set we can use GAP to generate the group Q (A.0.11) and compute $|Q| = 3,674,160$. We can then use GAP to output the cycle decomposition of each permutation in Q. Although permutation group Q is much larger than any other we have considered in an enumeration example, it is still computationally viable.

Our permutation group Q acts naturally on the left of Y. Let C be a set of r colours. We can then define the set of all colourings of the pocket cube with these colours as $Map(Y, C)$. It follows from the previous chapter that Q acts on the right of $Map(Y, C)$.

Letting $\mathscr{F} = Map(Y, C)$, we can compute the enumerator of this system using 4.3.1. Evaluating all of the indeterminates of the enumerator at 1 yields the number of configurations up to cube moves of the pocket cube. This computation is performed using Python3 (A.0.12). The following table shows how the number of configurations increases as r increases.

r	2	3	4	5	6
Count	1088	711,512	215,201,112	28,068,258,612	1,793,681,551,668

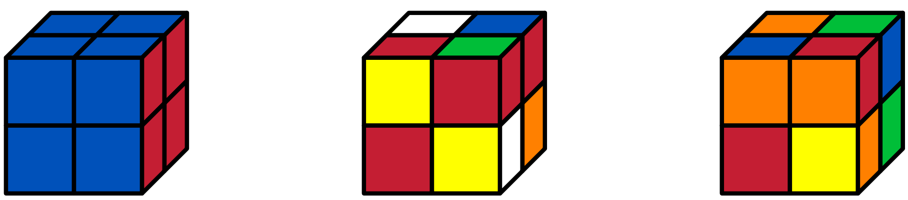


Fig. 5.8 U, F, R faces of various r-coloured pocket cubes.

6

CONCLUSION

In short, the aims outlined in the introduction have been met. The introductory material contained in chapter 2 and chapter 3 have been built upon in chapter 4 to derive the various formulations of PET. A plethora of examples have been included to build motivation. The combinatorial complexity of some of these examples (notably: 3.2.2, 4.2.7 and 5.3) give justification for the use of these enumeration methods in place of brute-force approaches. However, the time-complexity of the algorithmic implementations of PET listed in the appendix can certainly be improved and offer a direction for further research.

There are many avenues for further research regarding Pólya theory. One being the advancements made by Dutch mathematician Nicolaas de Brujin that involve introducing an additional permutation group on the codomain of the underlying set transformations [9]. Another being application to a variety of problems in graph theory such as enumerating rooted trees. American graph theorist Frank Hararay provides a detailed account of this application in his book *Graph theory* [4].

Two results in this thesis (2.2.6 and 4.4.2) have been included without proof. With more time, I would have liked to provide the proofs for these results as well. Moreover, the discussion of the Rubik's cube could have been more detailed, and I would have liked to derive the cube group represented as a product of common permutation groups on page 45.

REFERENCES

[1] M. Aigner. *Combinatorial Theory*. Springer, Dec 1996.

[2] M. Aigner. *A Course in Enumeration*. Springer Science Business Media, Jun 2007.

[3] D.S. Dummit and R.M Foote. *Abstract algebra*. John Wiley Sons, Danvers, 2004.

[4] Frank Harary. *Graph theory*. Cambridge, Mass. Perseus Books, 2015.

[5] D. Joyner. *Adventures in group theory : Rubik's cube, Merlin's machine, and other mathematical toys*. Johns Hopkins University Press, Baltimore, 2008.

[6] D.E. Knuth. *The art of computer programming. 3 Sorting and searching*. Addison-Wesley, 2014.

[7] J.S. Madachy. *Madachy's Mathematical Recreations*. New York: Dover Publications, 1979.

[8] G. Polya and R.C. Read. *Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds*. Springer Science Business Media, Dec 2012.

[9] R. C. Read. Pólya's theorem and its progeny. *Mathematics Magazine*, 60(5):275, Dec 1987.

[10] K.H. Rosen. *Elementary number theory and its applications*. Addison-Wesley, Boston, 2011.

[11] D. Singmaster. *Notes on Rubik's Magic Cube*. London: The Author, 1981.

[12] R.P. Stanley. *Enumerative combinatorics. Vol. 1*. Cambridge University Press, Cambridge, 2012.

[13] R.P. Stanley. *Algebraic Combinatorics*. Springer Science Business Media, Jun 2013.

[14] M.J Wenninger. *Dual models*. Cambridge University Press, Cambridge, 2003.

[15] R Wong. Rubiks Cube Group, 2016.

A

APPENDIX

A.0.1 *Code: rook configuration count (Python3)*

```
import numpy as np;
import math as mth
# board size
n = 14
# board size factorial (number configs fixed by identity)
nn =  mth.factorial(n)
# floor of the board size over 4
t = mth.floor(n/4)
# counters
m = 0
j = 1
k = 0
# number of configs fixed by diagonal
def involution(num):
    if num == 0 or num == 1:
        return 1
    i = [1,1]
    while m < num:
        InvNo = i[j] + m*i[k]
        i.append(InvNo)
        j += 1
        k += 1
        m = m + 1
    return i[-1]
```

```
# number of configs fixed by 180 rotation
def rotate180(num):
    t2 = 2*t
    ii = []
    k = 0
    while k <= t2:
        term = num - 2*k
        ii.append(term)
        k = k + 1
    return ii
# computing burnside
NoInvs = involution(n)
Rot180 = rotate180(n)
prod = mth.prod(Rot180)
configCount = 1/8 * (nn + 2*NoInvs + prod)
print(configCount)
```

A.0.2 *Code:* *generating 4-words from 3-alphabet*

```
import itertools as iter
alphabet = {'R','B','G'}
# create list of words from the alphabet above of length 4
words = [''.join(i) for i in iter.product(alphabet, repeat = 4)]
# print this information
print(len(words),words)
```

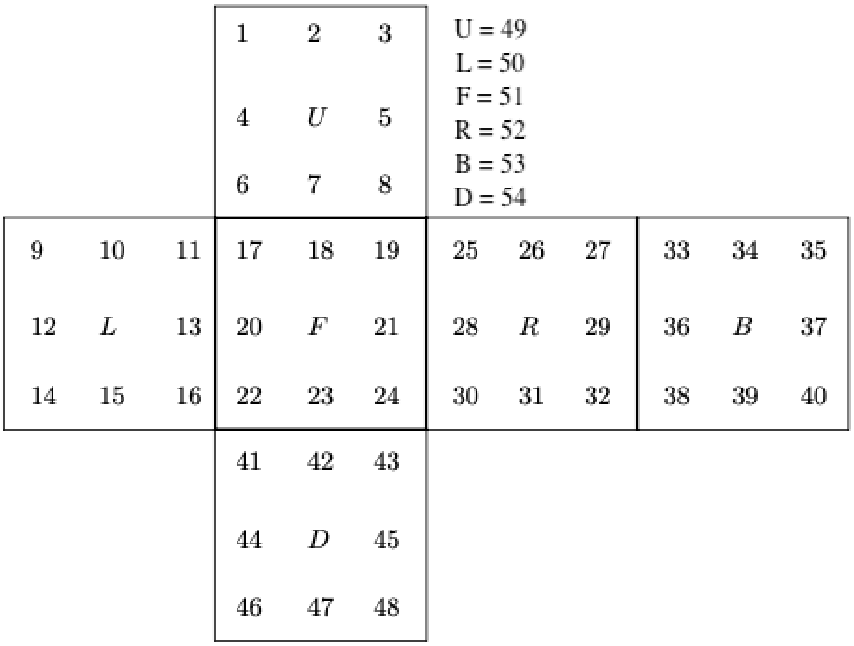
A.0.3 *Table:* *4-words from alphabet $\{R, B, G\}$ fixed by rotations*

id	BBBB, BBBG, BBBR, BBGB, BBGG, BBGR, BBRB, BBRG, BBRR, BGBB, BGBG, BGBR, BGGB, BGGG, BGGR, BGRB, BGRG, BGRR, BRBB, BRBG, BRBR, BRGB, BRGG, BRGR, BRRB, BRRG, BRRR, GBBB, GBBG, GBBR, GBGB, GBGG, GBGR, GBRB, GBRG, GBRR, GGBB, GGBG, GGBR, GGGB, GGGG, GGGR, GGRB, GGRG, GGRR, GRBB, GRBG, GRBR, GRGB, GRGG, GRGR, GRRB, GRRG, GRRR, RBBB, RBBG, RBBR, RBGB, RBGG, RBGR, RBRB, RBRG, RBRR, RGBB, RGBG, RGBR, RGGB, RGGG, RGGR, RGRB, RGRG, RGRR, RRBB, RRBG, RRBR, RRGB, RRGG, RRGR, RRRB, RRRG, RRRR
σ	BBBB, GGGG, RRRR
σ^2	BBBB, GGGG, RRRR, BGBG, GBGB, BRBR, RBRB, GRGR, RGRG
σ^3	BBBB, GGGG, RRRR

A.0.4 **Code:** *necklace configuration count (Python3)*

```
from math import gcd

#no. of bead positions, no. of colours of bead, sum placeholder
n = 3
r = 3
sum = 0
# list of divisors of n to be appended
divisors = []

# euler's totient function
def phi(n):
    amount = 0
    for k in range(1, n + 1):
        if gcd(n, k) == 1:
            amount += 1
    return amount

# appending divisors < n to list
for x in range (1,n):
    if n%x==0:
        divisors.append(x)

# appending n to this list also
divisors.append(n)
# for divisor in list
for k in divisors:
    # calculate as in formula
    sum += phi(k)*r**(n/k)

# divide sum by n as in formula
h = 1/n * sum
# print configuration count
print(h)
```

A.0.5 ***Image:*** *Labelling the 54 facets of Rubik's cube*



A.0.6 ***Table:*** *Cycle decomposition of each cube move with respect to above labelling*

U	$(1,3,8,6)(2,5,7,4)(9,33,25,17)(10,34,26,18)(11,35,27,19)$
D	$(41,43,48,46)(42,45,47,44)(14,22,30,38)(15,23,31,39)(16,24,32,40)$
F	$(17,19,24,22)(18,21,23,20)(6,25,43,16)(7,28,42,13)(8,30,41,11)$
B	$(33,35,40,38)(34,37,39,36)(3,9,46,32)(2,12,47,29)(1,14,48,27)$
L	$(9,11,16,14)(10,13,15,12)(1,17,41,40)(4,20,44,37)(6,22,46,35)$
R	$(25,27,32,30)(26,29,31,28)(3,38,43,19)(5,36,45,21)(8,33,48,24)$

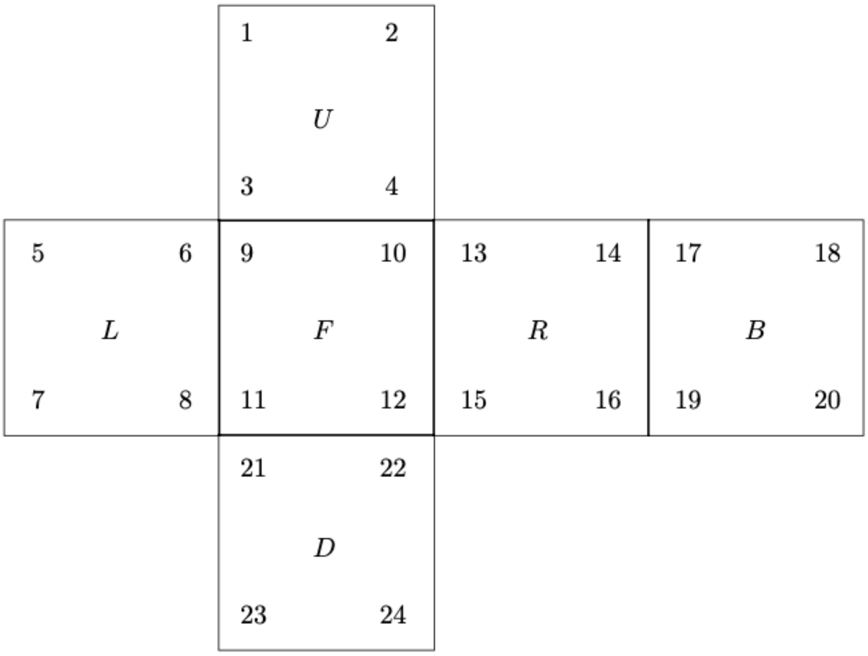
A.0.7 ***Code:*** *computing cardinality of Rubik's cube group (GAP)*

```
# generators of cube group
U := (1,3,8,6)(2,5,7,4)(9,33,25,17)(10,34,26,18)(11,35,27,19);
D := (41,43,48,46)(42,45,47,44)(14,22,30,38)(15,23,31,39)(16,24,32,40);
F := (17,19,24,22)(18,21,23,20)(6,25,43,16)(7,28,42,13)(8,30,41,11);
B := (33,35,40,38)(34,37,39,36)(3,9,46,32)(2,12,47,29)(1,14,48,27);
L := (9,11,16,14)(10,13,15,12)(1,17,41,40)(4,20,44,37)(6,22,46,35);
R := (25,27,32,30)(26,29,31,28)(3,38,43,19)(5,36,45,21)(8,33,48,24);

# creating cube group and printing order
M := Group(U,D,F,B,L,R);
Order(M);
```

A.0.8 **Code:** *creating subgroups of Rubik's cube group (GAP)*

```
# creating permutations used for normal closure N and subgroup L
o1 := R*U*D*B^2*U^2*B^3*U*B*U*B^2*D^3*R^3*U^3;
o2 := B*R^3*D^2*R*B^3*U^2*B*R^3*D^2*R*B^3*U^2;
p1 :=  R^2*U^3*F*B^3*R^2*F^3*B*U^3*R^2;
# creating MO's NCL and MP
O := Subgroup( M, [ o1, o2 ]);
L := Subgroup( M, [ U^2, D^2, F, B, R^2, L^2, p1 ]);
N := NormalClosure( M, O );
# seeing that their product has same order as cube
Order(N)*Order(L);
# seeing that their intersection is identity
Size(Intersection(O,L));
Elements(Intersection(O,L));
```

A.0.9 **Image:** *Labelling the facets of pocket cube*



A.0.10 **Table:** *Cycle decomposition of each cube move with respect to above labelling*

U	$(1,2,4,3)(9,5,17,13)(10,6,18,14)$
D	$(21,22,24,23)(7,11,15,19)(8,12,16,20)$
F	$(9,10,12,11)(3,13,22,8)(4,15,21,6)$
B	$(17,18,20,19)(2,5,23,16)(1,7,24,14)$
L	$(5,6,8,7)(1,9,21,20)(3,11,23,18)$
R	$(13,14,16,15)(2,19,22,10)(4,17,24,12)$

A.0.11 *Code: generating cycle decomposition of pocket cube operations (GAP)*

```
LogTo('/Users/cathaoir/Desktop/gap.txt'); #export terminal log to txt

# generators of pocket cube group
U := (1,2,4,3)(9,5,17,13)(10,6,18,14);
L := (5,6,8,7)(1,9,21,20)(3,11,23,18);
B := (17,18,20,19)(2,5,23,16)(1,7,24,14);

# creating pocket cube group
cube := Group(U,L,B);
# verifying has correct order
Order(cube);
# print cycle decomposition of each element
Elements(cube);
```

A.0.12 *Code: cube configuration count (Python3)*

```
import re

# import element data from GAP and reformatting for use
file = open("/Users/cathaoir/Desktop/gap.txt","r")
FromGap = file.read()
FromGap = FromGap.replace('\n','')
FromGap = FromGap.replace(" ","")

# number of colours
colorCount = 4
list = []
CycleLengths = []

# function that replaces commas inside a pair of brackets
# with a black space
def replace(g):
    return g.group(0).replace(',',' ')

# removes any new lines that exist in the txt file
FromGap = re.sub(r'\(.*?\)', replace, FromGap)
# splits the data into seperate elements
Cycles = FromGap.split(',')
```

```
# for element in pocket cube group
for d in Cycles:

    # create a list of cycles from decomposition
    m = re.findall(r'\(((?:\d+\s*)+)\)', d)

    # for each cycle append its length
    for i in m:
        list.append(len(i.split()))

    # append list of length information to list
    CycleLengths.append(list)
    list = []
# additive identity
sum = 0
# multiplicative identity
t = 1
# for list corresponding to element length data
for item in CycleLengths:

    # 24 is maximum possible number of 1-cycles
    a = 24
    t = 1

    # for integer i from 2 to 24
    for i in range(2,25):
        # how many cycle of length i exist?
        x = item.count(i)
        # number of 1-cycles reduces based on this
        a = a - (i*x)
        # power function terms s_2 to s_24
        t = (colorCount**x) * t
    # power function term s_1
    t = (colorCount**a) * t
    # summing these for each element
    sum += t
# dividing by cardinality of rubik group
# i.e the outcome of cube count under symmetry
print(sum / len(CycleLengths))
```

INDEX