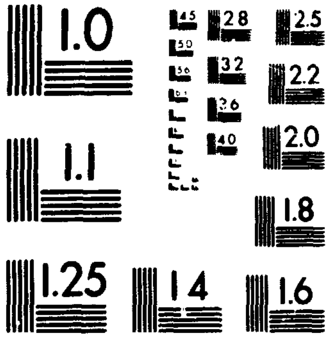
1 of/de 1

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT



Generating Random Elements in a Permutation Group

by

Vincenzo Acciaro

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Computer Science

School of Computer Science
Carleton University
Ottawa, Ontario
July 10, 1991

ISBN 0-315-70961-8

The undersigned recommend to the faculty of Graduate Studies
and Research acceptance of the thesis
Generating Random Elements in a Permutation Group
submitted by Vincenzo Acciaro
in partial fulfilment of the requirements for
the degree of Master of Computer Science

M. D Atkinson

Thesis Supervisor

F Fiala

Chairman, School of Computer Science

Carleton University

July 10, 1991

ABSTRACT

In this thesis we present a new algorithm for testing whether the group **G** generated by a given set of **m** permutations of degree **n** is regular. The worst case execution time of this algorithm is $O(m^2n)$ and the expected execution time is $O(mn(\alpha(n)+e(G)))$where $\alpha(\cdot)$ represents the inverse of Ackerman's function anu $e(G)$ the expected number of elements of **G** which have to be drawn at random before a set of generators is found. The function $e(G)$ is then computed for some common classes of groups. Finally, we discuss the problem of generating uniformly distributed random elements in arbitrary groups by forming random words in the generators, and point out the connections with the representation theory. A solution to this problem would give an upper bound to the probability of error of a probabilistic algorithm for computing a chain of stabilisers, due to J. Leon. A brief survey of the terminology, concepts and basic algorithms dealing with permutation groups is also provided.

For my parents
Adamo and Laura Acciaro

ACKNOWLEDGMENTS

I would like to express my profound gratitude to Dr. M.D. Atkinson, my thesis advisor, for introducing me to such a fascinating subject, and for guiding me through my research with such enthusiasm and expertise. His professionalism is aptly demonstrated by his ability to make the difficult appear simple. I am sure the patience and consideration shown by Dr. Atkinson will continue to inspire me in the future.

I would also like to thank Dr. J.D. Dixon for his suggestions and for our many interesting discussions regarding the subject of this thesis.

Prof. V.L. Plantamura deserves my most grateful recognition for his continued support and invaluable advice.

I am indebted to Dr. N. Santoro for his encouragement and kind assistance.

Finally, I would like to thank Alison M. Turner, William and Elena Anselmi, for their support during the gestation of this thesis, and all the other people who made my stay in Ottawa more pleasant.

TABLE OF CONTENTS

CHAPTER 0. Introduction..1

CHAPTER 1. Background..5

1.1. Action of a group on a set. 5

1.2. Graph associated to a permutation group..5

1.3. Orbits. 6

1.4. Transitivity. 8

1.5. Coset representatives...9

1.6. Obtaining a set of generators for G_α..11

1.7. Semiregular and regular groups...13

1.8. Blocks of imprimitivity. 14

1.9. Graphs associated to a block system...16

1.9.1. A block finding algorithm..17

1.9.2. An improved b!ock finding algorithm. 18

CHAPTER 2. Tests for regularity..21

2.1. Sim's test for regularity..21

2.2. New tests for regularity..23

2.2.1. A probabilistic version..26

2.2.2. A deterministic version. 28

2.2.3. A simplified deterministic version..30

2.3. Expected time complexity for the deterministic version...........................31

CHAPTER 3. Computation of the Eulerian function for some common families of groups. 34

3.1. The group of order one.. 35

3.2. Groups of prime order.........35

3.3. P-groups.........36

3.3.1. Computation of a presentation for G/Φ(G) when G is a p-group given by generators and relations.........40

3.3.2. Application to the dihedral groups of order 2^n.........41

3.3.3. The generalized quaternion groups, Q_n.........42

3.4. Groups of order p^2, p prime.........43

3.4.1. Cyclic groups of order p^2.........43

3.4.2. Elementary abelian groups of order p^2.........44

3.5. Groups of order 2p, p prime greater than two.........45

3.5.1. Cyclic groups of order 2p.........45

3.5.2. Dihedral groups of order 2p.........46

3.6. On the direct product of two groups.........47

3.7. Groups of order pq, p and q primes.........49

3.7.1. Cyclic groups of order pq, p and q primes.........49

3.7.2. Non abelian groups of order pq, p and q primes.........50

3.8. Groups of small order.........51

3.9. The groups PSL(2,k).........55

3.10. On the direct product of isomorphic simple groups.........58

CHAPTER 4. Random walk on a Cayley graph.........59

4.1. Introduction.........59

4.2. Cayley graphs.........61

4.3. Properties of the adjacency matrix of a Cayley graph.........63

4.4. On the rate of convergence to uniform.........65

4.5. Automorphisms of a graph.........67

4.6. Review of fundamental concepts in representation theory.........68

4.7. Regular representations.........70

4.8. Some final considerations. 73

4.9. Methodology to build an example. 74

4.10. A small example : S_4..75

Re'erences. 78

0. Introduction.

The study of symmetry is a recurring theme in Mathematics and Science. The ancient Greeks studied the symmetries of solids in space, as isometries of their vertices - an *isometry* of a set Ω of points in space is a permutation of Ω which maintains the distances unaltered. In the eighteenth and nineteenth centuries the theory of equations advanced through the analysis of symmetries of the set of roots of an equation. Most recently chemists have studied the symmetries of crystals and molecules, and physicists have found symmetry to be important at the subatomic level.

A core observation is that the composition of two symmetries is again a symmetry, and on this property hinges the whole theory of groups.

Early nineteenth century work in group theory was concerned with "concrete groups", in which it is always possible to compute the product of two "operations" - the most common examples of "concrete groups" are groups of matrices and groups of permutations.

Somewhat later in the century the concept of "abstract group" was developed, realizing that the structure of a group does not depend upon the nature of its "operations" - thereafter called elements. In contraposition to "concrete groups" which are "presented" by exhibiting real objects, an "abstract group" is "presented" by giving a set of generators and defining relations.

It is often useful to "go back in time" and consider "concrete groups" instead of abstract ones - the connection between the two lies in representation theory.

A representation is an homomorphism of an abstract group into a concrete one.

For computational purposes, it is useful to represent a group as a group of permutations: by Cayley's theorem every finite group can be represented as a permutation group, but since the degree of this group is equal to its order this representation may be ineffective for any practical purpose. Fortunately, most groups have a permutation representation of small degree.

Very efficient algorithms have been developed in the last two decades dealing with permutation groups specified by a set of generating permutations: on the one hand, we have algorithms which test if the group satisfies particular properties, such as transitivity or primitivity, on the other, we have algorithms which compute the order of the group or decide if a permutation belongs to it. Here, we should recall that for abstract groups the last two problems are simply undecidable.

The *leitmotif* of this thesis, viz. the generation of random elements in a permutation group, while being interesting on its own right, is considered here as being instrumental to the analysis of two (unrelated) algorithms which deal with permutation groups - the analysis of the execution time for a newly developed test for regularity and the analysis of the probability of error for an existing algorithm which builds a chain of stabilisers.

In the first chapter a brief survey of the terminology, concepts and basic algorithms dealing with permutation groups is given. We begin by associating a labelled graph to each permutation group and then continue by reviewing some fundamental algorithms in a graph theoretical fashion.

We show that some properties of the group, like transitivity, depend exclusively on the connectivity of the associated graph, while others, like regularity, depend also on the labelling of the edges.

A comprehensive source of reference for the entire chapter is [Atk90].

In the second chapter we present a new algorithm for testing whether the group **G** generated by a given set of **m** permutations of degree **n** is regular - a transitive permutation group is said to be *regular* if its order and degree are equal.

The worst case execution time of this algorithm is $\mathbf{O(m^2n)}$ and the expected execution time is $\mathbf{O(mn(\alpha(n)+e(G)))}$- here $\alpha(\cdot)$ represents the inverse of Ackerman's function, a very slow growing function, and e(G) the expected number of elements of G which have to be drawn at random before a set of generators is found.

We compare the execution time of this algorithm to that of an existing one, due to C. Sims, presented in lectures given at Oxford in 1973, whose execution time is $O(m^2n)$, and we show the superiority of our algorithm when the set of generators is redundant or not minimal - this may happen, for example, if the generating set is the output of a computer program.

We also consider two variants of the algorithm which are characterized by a short and elegant proof of correctness and ease of implementation, while retaining an execution time which is comparable to the more sophisticated one.

A reference for the entire chapter is [Acc90].

In the third chapter we compute the function e(G), previously introduced, for some common classes of groups, starting from the simplest one, the identity group.

We give a detailed discussion of e(G) when G is a p-group, i.e. a group whose order is the power of a prime p, and we prove that for these groups the quantity e(G) is related exclusively to p and to the minimal number of elements needed to generate them.

Groups which are the direct product of groups of coprime order are also analysed and it is shown how to compute the function e(G) for them.

With these two results at our disposition we can thoroughly analyse the class of nilpotent groups, i.e. those groups which are the direct product of their Sylow subgroups, which includes among others the class of abelian groups.

Several general approaches for computing the function e(G) are given through the chapter and, to show their validity, we employ them to compute e(G) for all the groups of order less than sixteen.

For the entire chapter refer to [Hal36].

In the last chapter we discuss the problem of generating uniformly distributed random elements in arbitrary groups by forming random words in the generators, and point out the connections with the representation theory of groups.

A solution to this problem would give an upper bound to the probability of error of a probabilistic algorithm, due to J. Leon, which solves the membership problem in an arbitrary permutation group and allows the generation of random permutation with a truly uniform distribution.

For the entire chapter refer to [Dia88].

1. Background.

1.1. Action of a group on a set.

Definition 1.1.1 We say that a *group G acts on a set Ω* when there is a function $\Omega \times G \rightarrow \Omega$ which associates to each couple (α, g) an element α^g such that

(i) $(\alpha^g)^h = \alpha^{gh}$ $\forall \alpha \in \Omega, \forall g, h \in G$

(ii) $\alpha^1 = \alpha$ $\forall \alpha \in \Omega$

When such a function is given, we call Ω a G-set. It can be proved that because of this function every element of G induces a permutation on the set Ω, and that the function which associates to each element of G the corresponding permutation of Ω is a homomorphism. When this homomorphism is one to one we call Ω a *faithful G-set*: in this case we can identify G with a subgroup of S_Ω, and we say that G is a *group of permutations of Ω.*

Example 1.1.2 If G is a group and H is a subgroup of G, let Ω be the family of the right cosets of H in G. We can define an action of G on Ω in the following way: if $\alpha = Hk$ is an element of Ω and g is an element of G, α^g is the coset of H obtained by multiplying the coset α on the right by g, i.e. the coset Hkg. Usually the set Ω in this example is written as G//H.

1.2. Graph associated to a permutation group.

Let G be a permutation group acting on a set Ω, generated by a subset H of its elements. Let us suppose that $\Omega = \{1,2,\ldots,n\}$ and $H = \{h_1,h_2,\ldots,h_m\}$. We can associate to the 3-tuple (G,Ω,H) a directed graph $\mathfrak{G}$ whose vertices are the points of Ω and such that there is an edge connecting x to y labelled **h** if and only if $x^{h} = y$, for some generator **h** in H.

Given such a graph $\mathfrak{G}$ and two vertices x and y not necessarily distinct, we can identify a path going from x to y with the ordered sequence $h_{i_1},h_{i_2},\ldots,h_{i_k}$ of the edge labels: it is easy to verify that the element $g = h_{i_1} \cdot h_{i_2} \cdot \ldots \cdot h_{i_k}$ of G moves x to y.

To represent $\mathfrak{G}$ inside a computer we need to store the generators of G: to store a permutation of degree **n** we require O(n) memory cells, and since we have **m** permutations belonging to the generating set a total of O(**m n**) memory cells are required.

In this chapter we will review some fundamental algorithms dealing with permutation groups using to explain them properties of the associated graphs. We will see that some properties of the group, like transitivity, depend exclusively on the connectivity of the associated graph, while other properties, like regularity, depend also on the labels of the edges of the graph.

1.3. Orbits.

Definition 1.3.1 If a group G acts on a set Ω and α∈ Ω , the *orbit of α under the action of G* is the subset of Ω

$$\alpha^G = \{\ \alpha^g \mid g \in G\ \}$$

In other words, the orbit containing α consists of the points into which α is moved under the action of all the elements of G.

It can be proved easily that two orbits are either coincident or disjoint, and therefore the set of orbits constitutes a partition of the set Ω : to this partition it is possible to associate an equivalence relation, where two points of the set Ω are equivalent if they belong to the same orbit.

We have seen that given a permutation group **G** acting on a set Ω, generated by a subset **H** of its elements, we can associate to it a directed graph $\mathfrak{G}$ whose vertices are the points of Ω and such that there is an edge connecting **x** to **y** labelled **h** if and only if $x^h = y$, for some generator **h** in **H**.

To find the orbit containing a point α we merely need to find the set of points which can be reached from α in $\mathfrak{G}$: this can be done by building a spanning tree rooted at α. In fact it is easy to see that if there is a permutation **g** moving α to β, then **g** must be expressed as the product $h_{i_1} \cdot h_{i_2} \cdot \ldots \cdot h_{i_k}$ of generators of **H**, and therefore there is a directed path starting from α and ending in β whose arcs are labelled in order $h_{i_1}, h_{i_2}, \ldots, h_{i_k}$.

A desirable property of this tree is to have a minimal height[1]: this is achieved by using a breadth first traversal[2] of the graph $\mathfrak{G}$.

The above considerations lead to the **orbit finding** algorithm:

```
Algorithm   ORBIT
Input:      the graph $\mathfrak{G}$ associated to $(G,\Omega,H)$ and a vertex $\alpha$ of $\mathfrak{G}$
Output:     a vector VISITED, with
                    VISITED[$\beta$] = yes if the vertex $\beta$ can be reached from $\alpha$
                    VISITED[$\beta$] = no if $\beta$ cannot be reached from $\alpha$
    Q := queue containing a single item $\alpha$
    for each vertex $v \in \mathfrak{G}$ do
        VISITED[v] := no
    endfor
    repeat
```

[1] The reason of this assertion will become clear in section 1.5.

[2] For the definition of breadth first traversal see [Aho74, section 2.4].

```
            extract v from the queue Q
            for each $h \in H$ do
            if      VISITED[$v^h$] = no then
                    VISITED[$v^h$] := yes
                    insert $v^h$ in the queue Q
            endif
        until Q is empty
end
```

It is easy to see that the execution time of this algorithm is **O(m n)**, since the outdegree of each vertex is **m** and **n** is the number of vertices to be visited in the worst case.

1.4. Transitivity.

Definition 1.4.1 We say that a group *G acts transitively on a set Ω* if there is just one orbit:

$$\alpha^G = \Omega \text{ , } \forall \alpha \in \Omega$$

The definition implies that given any two points α and β of Ω there is a permutation $g \in G$ which moves α to β. Being transitive depends on the way in which the group acts on the set, as we are going to show in the following example:

Example 1.4.2 Consider the following permutation groups:

G' : {1, (12)(34), (13)(24), (14)(23)}

G" : {1, (12), (34), (12)(34)}

G' and **G"** are two permutation representations of the same abstract group, the *Klein four group*, but while **G'** is transitive on the set {1,2,3,4}, **G"** is not.

In the light of the discussion about the graphs associated to a permutation group G acting on a set Ω, generated by a subset **H** of its elements, we can say that G acts transitively on Ω if in the associated graph any vertex can be reached from a fixed vertex through a directed path.

The above considerations lead to the **transitivity** algorithm:

```
Algorithm   TRANSITIVE
Input:      the graph $\mathfrak{G}$ associated to $(G,\Omega,H)$
Output:     true if the group G generated by H acts transitively on $\Omega$, false otherwise
        select a vertex $\alpha$ of $\mathfrak{G}$
        apply the orbit algorithm to $\mathfrak{G}$ and $\alpha$
        for each vertex $v \in \mathfrak{G}$ do
                if      VISITED[v] = no then return(false)
        endfor
        return(true)
end
```

It is easy to see that the execution time of this algorithm is dominated by the execution time of the algorithm orbit, and therefore is **O(m n)**.

1.5. Coset representatives.

Definition 1.5.1 If **G** is a permutation group acting on a set Ω then the set of all elements in **G** which fix a point α of Ω is denoted by $\mathbf{G}_\alpha$ and is called the *stabiliser of the point α.*

It can be easily seen that G_α is a subgroup of G: in fact the identity certainly belongs to G_α, if an element x of G fixes α its inverse fixes α too, and finally if x and y are two elements of G which fix α their product also fixes α.

Theorem 1.5.2 (orbit-stabiliser theorem) If a group G acts on a set Ω and $\alpha \in \Omega$, there is a one to one correspondence between the elements belonging to a set of coset representatives of G_α and the points belonging to the orbit containing α. Thus

$$|G:G_\alpha| = |\alpha^G|$$

Proof. See [Dix67, problem 2.12].

We can therefore identify each coset of the stabiliser of α with a point belonging to the orbit containing α. If $\beta \in \alpha^G$ then we write u_β to refer to that coset representative of the set which sends α to β, i.e. such that $\alpha^{u_\beta} = \beta$.

We have seen how to compute the orbit containing a point α by exploiting the graphical analogy. It is easy to extend the **orbit** algorithm to compute a set of coset representatives for G_α: for each vertex β of the graph $\mathfrak{G}$ that is being visited u_β is defined as the product of the edge labels in the path that leads from α to β.[3] In particular, u_α is set equal to 1, and during the breadth first traversal if a vertex γ leads to a vertex δ not yet visited, through an arc labelled h, then we set $u_\delta = u_\gamma h$.

The above considerations lead to the **coset representatives** algorithm:

Algorithm COSET_REPRESENTATIVES

Input: the graph $\mathfrak{G}$ associated to (G,Ω,H) and a vertex α of $\mathfrak{G}$

Output: a set $\{u_i\}$ of representatives of the cosets of G_α

[3] Now it becomes clear the reason why it is preferable to have a tree of minimal height: the time required to compute u_β is proportional to the length of the path from α to β, and a tree of minimal height minimizes the average length of the paths.

```
        Q := queue containing a single item $\alpha$
        $u_\alpha$ := 1
        for each vertex v $\in \mathcal{G}$ do
                VISITED[v] := no
        endfor
        repeat
                extract v from the queue Q
                for each h $\in$ H do
                if      VISITED[$v^h$] = no then
                        VISITED[$v^h$] := yes
                        $u_{v^h} := u_v \cdot h$
                        insert $v^h$ in the queue Q
                endif
        until Q is empty
end
```

It is easy to verify that the execution time of this algorithm is $O(m\,n + n^2)$.

1.6. Obtaining a set of generators for G_α

There is an easy way to obtain a set of generators for the stabiliser of a point α: the method relies upon the following theorem:

Theorem 1.6.1 (Schreier theorem) If G is a group generated by a subset $\mathbf{H=\{h_1,h_2,...h_m\}}$ of its elements, **K** is a subgroup of G and $\mathbf{u_1,u_2,...,u_n}$ a complete set of representatives for the cosets of **K**, then the set of all the products $\mathbf{u_i\, h_j u_k^{-1}}$ lying in **K** is a set of generators for **K**.

Proof. See [Jon90, section 2.3].

How can we apply this theorem to find a complete set of generators for G_α? We have seen before how to build a set of coset representatives for G_α while building a spanning tree rooted at α. Let $\mathbf{u}_\gamma$ stand for the coset representative of G_α mapping α to γ. Let us suppose now that during the traversal of the graph $\mathfrak{G}$ a vertex β leads through an arc labelled $\mathbf{h}_j$ to a vertex δ already visited. The path going from α to β is represented by the product $\mathbf{u}_\beta$ of the edge labels, and the path going from α to δ is represented by the product $\mathbf{u}_\delta$. It is easy to see that the product $\mathbf{u}_\beta \cdot \mathbf{h}_j \cdot \mathbf{u}_\delta^{-1}$ fixes α , i.e. it belongs to G_α, and is of the form required by the Schreier theorem. Moreover, it can be proved that no generator of G_α is missed, i.e. the breadth first traversal of the graph $\mathfrak{G}$ satisfies the conditions of the Schreier theorem.

The above considerations lead to the following algorithm:

```
Algorithm  GENERATORS_OF_$G_\alpha$
Input:     the graph $\mathfrak{G}$ associated to $(G,\Omega,H)$ and a vertex $\alpha$ of $\mathfrak{G}$
Output:    a set M of generators of $G_\alpha$
    M := empty set
    Q := queue containing a single item $\alpha$
    $u_\alpha$ := 1
    for each vertex v $\in \mathfrak{G}$ do
        VISITED[v] := no
    endfor
    repeat
        extract v from the queue Q
        for each h $\in$ H do
        if    VISITED[$v^h$] = no then
              VISITED[$v^h$] := yes
```

```
                $u_{v^h} := u_v \cdot h$
                insert $v^h$ in the queue Q
            else
                add $u_v \cdot h \cdot u_{v^h}^{-1}$ to the set M
            endif
        until Q is empty
end
```

It is easy to verify that the execution time of this algorithm is $O(m\, n^2)$.

1.7. Semiregular and regular groups.

Definition 1.7.1 A group G which acts on a set Ω is called *semiregular* if the stabiliser of each point of Ω is equal to the identity.

Example 1.7.2 Let G be a given group, G_1 and G_2 two groups isomorphic to G. Let $\Omega = G_1 \cup G_2$. Let G act on Ω in the following way: if $\omega \in \Omega$ and $g \in G$ then ω^g is defined as the product in G_1 or G_2, according to the group to which ω belongs. This action is clearly semiregular: in fact $\omega^g = \omega \cdot g = \omega$ implies $g = 1$. This action induces precisely two orbits, each of size $|G|$.

Suppose that $\mathfrak{G}$ is the directed graph associated to a permutation group G acting on a set Ω, generated by a subset H of its elements. It is easy to see that G acts semiregularly on Ω if and only if any two directed paths connecting two vertices x and y determine the same group element as product of their edge labels, for any pair of vertices x and y in $\mathfrak{G}$.

Definition 1.7.3 A group G which acts on a set Ω is said to be *regular* if it is semiregular and transitive.

As a consequence of the definition we have that the order of a regular permutation group is equal to its degree: this condition is equivalent, for a transitive permutation group, to the fact that the stabiliser of each point is the identity.

1.8. Blocks of imprimitivity.

Definition 1.8.1 If Ω is a transitive G-set, we say that a subset Δ of Ω is a *block* of Ω if for any $g \in G$ we have $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \varnothing$

Any G-set Ω always admits the *trivial blocks* $\Delta=\Omega$ and $\Delta=\{\alpha\}$ where α is a point of Ω. A transitive G-set Ω is called *primitive* if it admits only the trivial blocks.

Definition 1.8.2 If Δ is a block of Ω, the set $\{\Delta^g \mid g \in G\}$ is called a *block system* or *system of imprimitivity generated by* Δ.

It can be proved that all the blocks belonging to the same system of imprimitivity have the same size, which must therefore be a divisor of $|\Omega|$ - it follows that group G which acts transitively on a set Ω, with $|\Omega|$ prime, must necessarily be primitive.

Example 1.8.3 If we consider the isometries of a square lamina in 2-dimensional space, i.e. the group $\mathbf{D_8}$ acting on the set of points **{1, 2, 3, 4}** it is possible to construct a non trivial block by taking any two opposite corners of the square. Thus for this G-set we have the following block systems:

$\{\{1\}, \{2\}, \{3\}, \{4\}\}$	generated by $\{1\}$	trivial
$\{\{1, 3\}, \{2, 4\}\}$	generated by $\{1, 3\}$	non trivial
$\{\{1, 2, 3, 4\}\}$	generated by $\{1,2,3,4\}$	trivial

Example 1.8.4 If $H \leq K \leq G$, consider the set $G//H$ of right cosets of H in G on which G acts by right multiplication. The set of right cosets of H in K constitutes a block of the G-set $G//H$.

Example 1.8.5 If Ω is a transitive G-set and N is a normal subgroup of G, any orbit of N is a block of G.

The theorem that follows relates the set of blocks containing a given point α to the set of subgroups of a group G containing the stabiliser of α.

Theorem 1.8.6 (Correspondence Theorem) Let G be a group acting transitively on a set Ω and α a point of Ω. There is a one-to-one correspondence between the set of blocks which contain α

$$D = \{\Delta \mid \alpha \in \Delta\}$$

and the set of subgroups of G containing G_α

$$P = \{K \mid G_\alpha \leq K \leq G\}$$

More precisely:

(i) if $K \in P$ then the orbit α^K is a block

(ii) the correspondence $\Theta: P \rightarrow D$ defined by $\Theta(K) = \alpha^K$ is injective

(iii) the function φ defined for any element $\Delta \in D$ by $\varphi(\Delta) = \{g \in G \mid \alpha^g \in \Delta\}$ has codomain P

(iv) the functions Θ and φ are each the inverse of the other

Proof. See [Mac74, problem 49, p.117] or [Dix67, problem 2.31]

Corollary 1.8.7 A transitive G-set is primitive if and only if the stabiliser of a point is a maximal subgroup of G.

Proof. See [Mac74, problem 51, p.117] or [Dix67, problem 2.31]

To conclude this section we will show that if a group G acts on a set Ω, and α is a point of Ω, then the set of fixed points of G_α form a block. This result will be needed later, when we will discuss a new test for regularity.

Theorem 1.8.8 If G is a group which acts transitively on a set Ω and α a point of Ω, then the normalizer $N(G_\alpha)$ acts transitively on the set Δ of points fixed by G_α.

Proof. $\alpha^g \in \Delta \Leftrightarrow G_\alpha \leq G_{\alpha g} = (G_\alpha)^g \Leftrightarrow G_\alpha = (G_\alpha)^g$ because of the finiteness of G. The last equality is equivalent to $g \in N(G_\alpha)$, and therefore the orbit of α under the action of $N(G_\alpha)$ is precisely Δ.

Corollary 1.8.9 If G is a group which acts transitively on a set Ω, α a point of Ω, then the size of the set Δ of points fixed by G_α is equal to $[N(G_\alpha):G_\alpha]$.

Proof. Simply use the fact that $(N(G_\alpha))_\alpha = G_\alpha$, that $N(G_\alpha)$ acts transitively on Δ and finally the orbit-stabiliser theorem.

Theorem 1.8.10 If G is a group which acts transitively on a set Ω and α a point of Ω, then the set Δ of points fixed by G_α is a block of Ω.

Proof. We have proven above that the normalizer $N(G_\alpha)$ acts transitively on the set Δ of points left fixed by G_α. Now we have $G_\alpha \leq N(G_\alpha) \leq G$ and therefore $\Delta = \alpha^{N(G_\alpha)}$ is a block, by case (i) of the Correspondence Theorem discussed above.

1.9. Graphs associated to a block system.

The purpose of this section is to give a historical background for the block finding algorithm. Let G be a given permutation group which acts transitively on a set Ω. We define an action of G on the cartesian product $\Omega \times \Omega$ as follows:

$$(x\ ,\ y)^g = (x^g\ ,\ y^g) \qquad x,y \in \Omega$$

we will call this action the *diagonal action* according to [Rot88, p.191]. We would like to find the decomposition of $\Omega \times \Omega$ into disjoint orbits under this action. First we notice that the "diagonal" $\{(x , x) \mid x \in \Omega\}$ is always an orbit, because of the transitivity of G on Ω; let us call this orbit Δ_0. For the proofs of tre next theorems and lemmas refer to [Big79, chapter 4].

Lemma 1.9.1 If G acts transitively on Ω, then the number of orbits of $\Omega \times \Omega$ under the diagonal action of G is equal to the rank of G acting on Ω.

We recall here the fact that the *rank* of a group G acting transitively on a set Ω is just the number of orbits of Ω under the action of the stabiliser of an arbitrary point in Ω. To each G-orbit on $\Omega \times \Omega$ we associate a digraph, built as follows:

Definition 1.9.2 Given an orbit Δ_i of $\Omega \times \Omega$, $\Gamma(\Delta_i)$ is the directed graph whose vertices are the points of Ω and whose edges are the ordered pairs of Δ_i.

Theorem 1.9.3 Let G be a group acting transitively on Ω. G acts primitively on Ω if and only if each digraph $\Gamma(\Delta_i)$ is connected, for each orbit Δ_i of $\Omega \times \Omega$.

Lemma 1.9.4 Let G be a group acting transitively on Ω. Let x and y be two distinct points of Ω. The minimal block of imprimitivity containing both x and y corresponds to the set of vertices in the connected component {of a digraph $\Gamma(\Delta_i)$ } containing the edge (x,y).

1.9.1. A block finding algorithm.

The information embedded in Lemma 1.9.4 allows us to find the minimal block of imprimitivity containing two given points x and y of Ω. To do this we first compute the orbit of $\Omega \times \Omega$ containing (x , y) using the algorithm ORBIT, then we build the associated digraph and collect together the points in the connected component containing the edge $(\mathbf{x}, \mathbf{y})$. The problem with this approach is that it leads to an algorithm whose execution time is directly proportional to the average size of an orbit of $\Omega \times \Omega$, i.e. inversely proportional to the rank of $\mathbf{G}$ which we have seen to be equal to the number of orbits of $\Omega \times \Omega$. Consequently this algorithm behaves very well on a regular group, for which the rank is equal to $|\Omega|$, and very badly on a $\mathbf{k}$-transitive group with $\mathbf{k} \geq 2$, for which the rank is always two.

For further information about this approach refer to [Dix71] or [Sim67].

1.9.2. An improved block finding algorithm.

Given a permutation group $\mathbf{G}$ acting on a set Ω, generated by a subset $\mathbf{H}$ of its elements, the algorithm first described in [Atk75], finds the block system generated by the block of minimal size containing two given points α and ε of Ω, in $O(\mathbf{m}\ \mathbf{n}\ \alpha(\mathbf{n}))$ time, where $\alpha(\mathbf{n})$ is a very slow growing function related to the inverse of Ackerman's function.

In this section we will discuss a variant of the algorithm more suited to our purposes. Given a block system Σ and two points α and ε not belonging to the same block, the function EXTEND computes the block system Σ' having the smallest blocks in which α and ε belong to the same block and each block of Σ is contained in a block of Σ'.

The algorithm makes use of the operations FIND and UNION described in [Tar75], which are applied to the current partition Σ of Ω. The operation FIND(θ) returns a distinguished point in the class containing θ which will be used to identify the class. The operation UNION(θ,φ) replaces the partition Σ by a new one in which the classes containing θ and φ are collapsed into a single one.

The classes in Σ are represented by rooted trees defined by a "father" function $\mathbf{f}$: $\mathbf{f}(\theta)$ is the node immediately above θ in the set of trees, or θ itself if θ is a root. This representation allows an efficient implementation of the FIND and UNION operations.

The FIND(θ) operation uses the function **f** to trace a path from θ to the root of its tree, returning the root.

The UNION(θ,φ) operation inserts a branch between the root of the trees containing θ and φ if these trees are different. The weighting and path compression rules described in [Tar75] are used.

The function EXTEND also uses a stack C containing tree branches which are represented by pairs of end points.

```
function    EXTEND($\Sigma,\alpha,\varepsilon$)
input:      a group G acting on a set $\Omega$, generated by a subset H of its elements
            a block system $\Sigma$ of $\Omega$
            two points $\alpha$ and $\varepsilon$ not belonging to the same block
output:     a block system $\Sigma'$ having the smallest blocks in which
            $\alpha$ and $\varepsilon$ belong to the same block and
            each block of $\Sigma$ is contained in a block of $\Sigma'$.
    $\alpha^*$ := FIND($\alpha$) ;
    $\varepsilon^*$ := FIND($\varepsilon$)
    UNION($\alpha^*,\varepsilon^*$)
    C := empty stack
    push ($\alpha^*,\varepsilon^*$) into C
    repeat
        pop an element ($\gamma,\delta$) from C
        for each h $\in$ H do
            $\phi := \gamma^h$ ;          $\Psi := \delta^h$
            $\sigma$ := FIND($\phi$) ;     $\tau$ := FIND($\Psi$)
            if $\sigma \neq \tau$ then
                UNION($\sigma,\tau$)
                push $(\sigma,\tau)$ into C
            endif
        endfor
    until C is empty
return($\Sigma$)
```

Lemma 1.9.2.1 EXTEND($\Sigma,\alpha,\varepsilon$) computes the block system Σ' having the smallest blocks in which α and ε belong to the same block and each block of Σ is contained in a block of Σ'.

Proof. EXTEND($\Sigma,\alpha,\varepsilon$) manipulates a set of trees using the operations UNION and FIND. Let us suppose, temporarily, that path compression is not performed in the FIND operation. Then it is clear that the pairs in the stack C all represent tree edges. From the body of the **for** statement it follows that, at the end of each iteration of the repeat loop, we have the following condition:

> *for every branch (σ,τ) of the set of trees which represent the current partition one of the following holds:*
>
> *1.* $(\sigma,\tau) \in C$, *or*
>
> *2.* σ^h *and* τ^h *lie in the same tree, for every* $h \in H$

But on termination C is empty and so, for all tree branches (σ,τ), $\sigma^{\mathbf{h}}$ and $\tau^{\mathbf{h}}$ lie in the same tree, for every $\mathbf{h} \in \mathbf{H}$. Thus the partition in output is indeed a block system of the action of G on Ω, and because of the initial step α and ε belong to the same block. It is also clear that the output block system is the one containing the smallest blocks, because the algorithm only joins two blocks together when it is forced to do so to fulfil the properties. Finally, note that the block system computed by EXTEND($\Sigma,\alpha,\varepsilon$) is unaffected by how FIND is implemented so that the assumption above that path compression is not used can be removed.

2. Tests for regularity.

2.1. Sim's test for regularity.

Given a set of permutations we would like to determine if the group that they generate is regular or not; moreover, we would like this test to be as efficient as possible. An existing test for regularity is due to Charles Sims and is described in [Atk90]: its execution time is $O(\mathbf{m}^2\,\mathbf{n})$ where $\mathbf{m}$ is the number of generators and $\mathbf{n}$ the degree of the permutation group. To introduce the test we recall the definition of the *centraliser* of a group:

Definition 2.1.1 The *centraliser* of a permutation group G, denoted by C_G is the set of all permutations c of Ω which commute with every element in G.

It is easily seen that C_G is a subgroup of S_Ω. The property that is used for determining the regularity of a group is stated in the following lemma, whose proof can be found in [Atk90]:

Lemma 2.1.2 Let G be a transitive permutation group, H a set of generators for G and α a point of the set Ω on which G acts. Then G is regular if and only if

$$\forall\ \mathbf{h} \in \mathbf{H}\ \exists\ c \in C_G \text{ such that } \alpha^c = \alpha^{\mathbf{h}}$$

The test devised by Sims is quite easy to describe: given a transitive permutation group G generated by a subset H of its elements and a fixed point α of the set Ω, for each generator **h** in turn we look for an element c of C_G such that the images of α under **h** and c are the same: if for some generator **h** we cannot find such an element c of C_G we can assert that G is not regular. How can we find such an element $c \in C_G$? Being a permutation of the points of Ω we know **c** when we know the effect of **c** on each point of Ω. Therefore, for a given generator **h** we fix the initial condition $\alpha^c = \alpha^h$ and then we try to extend **c** to a permutation of Ω lying in the centraliser of **G**: by "extend" we mean determining incrementally the image of each point of Ω under **c**. If we have found the image of some point β under **c** and we want to find the image of a point γ under **c**, where $\gamma = \beta^g$ for some generator **g** of **G**, we use the fact that **c** commutes with **g**, and therefore $\gamma^c = (\beta^g)^c = (\beta^c)^g$. In such a way we can compute the images of the points of Ω under **c** while we compute the orbit of α, a job that we accomplish by a breadth first traversal in the graph representing the action of the generators of **G** on Ω.[4]

```
Algorithm   REGULAR
Input:      a subset H of elements of G which generate G
Output:     true if G is regular on $\Omega$, false otherwise
    select a point $\alpha \in \Omega$
    for each generator $h \in H$ do
        find   a mapping $c:\Omega \rightarrow \Omega$ satisfying the initial condition $\alpha^c = \alpha^h$
               by computing the orbit of $\alpha$
               and for each new point $\gamma = \beta^g$ added to the orbit, for some $g \in H$,
               setting $\gamma^c = (\beta^c)^g$
        if     c is not a bijection
               or
               c does not commute with some generator in H
        then return(false)
    endfor
    return(true)
end
```

[4] This is equivalent to building a spanning tree rooted at α.

Let us now analyze the execution time of the algorithm. We assume for the rest of this chapter that $|H| = \mathbf{m}$ and $|\Omega| = \mathbf{n}$. We also assume that G acts transitively on Ω, and so the size of the orbit containing α is equal to **n**.

The step "**find** a mapping $c:\Omega\rightarrow\Omega$ satisfying ..." takes time equal to the time required to build the orbit containing α, which is $O(\mathbf{m}\ \mathbf{n})$. Testing if c is a bijection takes time $O(\mathbf{n})$ where **n** is the degree of the permutation group. Finding if c does not commute with some generator in H takes time $O(\mathbf{m}\ \mathbf{n})$, since to test if c does not commute with a single generator of G we need $O(\mathbf{n})$ time and the test has to be repeated **m** times. The above steps have to be repeated for each generator in H, giving an overall execution time which is $O(\mathbf{m}^2\,\mathbf{n})$.

2.2. New tests for regularity.

We have seen that a group G which acts transitively on a set Ω is regular if its point stabiliser is the identity subgroup.

This suggests a new technique to test the regularity of a group G: we have seen that it is possible to extend the ORBIT algorithm to compute a set M of generators for G_α: if some generator was not equal to the identity then the group G would not be regular.

The considerations above lead to the following algorithm:

Algorithm REGULAR_SLOW

Input: the graph $\mathcal{G}$ associated to (G,Ω,H) and a vertex α of $\mathcal{G}$

Output: **true** if the group G is regular on Ω, **false** otherwise

```
    Q := queue containing a single item $\alpha$
    $u_\alpha$ := 1
    for each vertex $v \in \mathcal{G}$ do
        VISITED[v] := no
        endfor
        repeat
            extract v from the queue Q
            for each $h \in H$ do
            if      VISITED[$v^h$] = no then
                    VISITED[$v^h$] := yes
                    $u_{vh} := u_v \cdot h$
                    insert $v^h$ in the queue Q
            else
                    if $u_v \cdot h \cdot u_{vh}^{-1} \neq 1$ then return(false) endif
            endif
        until Q is empty
        return(true)
end
```

It is easy to verify that the execution time of this algorithm is $O(m\ n^2)$, since the test $u_v \cdot h \cdot u_{vh}^{-1} \neq 1$ takes $O(n)$ time, and it must be repeated $O(m\ n)$ times. We can conclude that this algorithm behaves worse than Sim's algorithm, whose execution time is $O(m^2\ n)$, since usually m is much smaller[5] than n. The improved algorithms discussed in this chapter all have a structure similar to that of REGULAR_SLOW.

At the heart of the new algorithms there is an efficient implementation of a function that decides whether the stabiliser of a point α fixes another point β: if α and β belong to the same G-orbit this corresponds to the test $G_\alpha = G_\beta$.

[5] For all practical purposes we could assume $m \approx \log n$

In the next sections we will show how to apply this function to decide if a group is regular, restricting this section to the description of the function, which we call EQUALSTABILISERS.

The key consideration is that for G_α to fix β any set **H** of generators of G_α must fix β, and in particular this assertion holds if **H** is a set of Schreier generators computed by the algorithm explained in section 1.6.

Therefore, if $\mathbf{u}_\mu \cdot \mathbf{k} \cdot \mathbf{u}_\delta^{-1}$ is a Schreier generator of G_α we must have

$$\beta^{\mathbf{u}_\mu \cdot \mathbf{k} \cdot \mathbf{u}_\delta^{-1}} = \beta.$$

but the latter equality is equivalent to the following one

$$\beta^{\mathbf{u}_\mu \cdot \mathbf{k}} = \beta^{\mathbf{u}_\delta}.$$

It is now easy to see how the function EQUALSTABILISERS is derived from the algorithm that computes a set of Schreier generators of G_α:

- Instead of labelling the vertex γ of the graph $\mathfrak{G}$ with the coset representative $\mathbf{u}_\gamma$ we label it with $\beta^{\mathbf{u}_\gamma}$.
- If during the breadth first traversal of the graph an arc labelled **h** takes the vertex γ to a vertex η not yet visited, then we label η with $(\beta^{\mathbf{u}_\gamma})^{\mathbf{h}}$, that is with the image of the label of the vertex γ by the permutation **h**: this labelling clearly can be done in constant time.
- If on the contrary the vertex γ is mapped, through an arc labelled **h**, to a vertex η already visited, then we have to verify the equality of the label of the vertex η with the image of the label of the vertex γ by the permutation **h**: this too can be done in constant time.

The following algorithm implements the function:

Function EQUALSTABILISERS(α,β)

Input: the graph $\mathfrak{G}$ associated to (G,Ω,H) and two vertices α and β of $\mathfrak{G}$

```
Output:        true if $G_\alpha$ fixes $\beta$, false otherwise
        Q := queue containing a single item $\alpha$
        LABEL[$\alpha$] := $\beta$
        for each vertex $v \in \mathfrak{G}$ do
                VISITED[v] := no
        endfor
        repeat
                extract v from the queue Q
                for each $h \in H$ do
                if      VISITED[$v^h$] = no then
                        VISITED[$v^h$] := yes
                        LABEL[$v^h$] := LABEL[v]$^h$
                        insert $v^h$ in the queue Q
                else
                        if LABEL[$v^h$] $\neq$ LABEL[v]$^h$ then return(false) endif
                endif
        until Q is empty
        return(true)
end
```

It is easy to see that the test $G_\alpha = G_\beta$ takes time $O(m\ n)$, corresponding to the time needed to visit the graph $\mathfrak{G}$.

2.2.1. A probabilistic version.

A single test $G_\alpha = G_\beta$ is not enough to guarantee the regularity of the group. Consider for example the isometries of a square lamina in 2-dimensional space, i.e. the group $\mathbf{D_8}$ acting on the set of corners $\{1, 2, 3, 4\}$: it is easily seen that the stabilisers of two opposite corners are equal, although the action is not regular, being $|\mathbf{D_8}|=8 \neq$ **#corners**.

The problem arises because of the fixed points of G_α: in fact the equality $\beta^{u_\mu \cdot k \cdot u_\delta^{-1}} = \beta$ is always satisfied when β is a fixed point of G_α, yet $G_\alpha \neq 1$.

The simplest test for regularity which applies the ideas described in the previous section is a probabilistic one in which we select some points β at random from the set Ω and verify the equality of the stabilisers of α and β, for a fixed α:

```
Algorithm   PROBABILISTIC_REGULAR
Input:      the graph $\mathfrak{G}$ associated to $(G,\Omega,H)$, a vertex $\alpha$ of $\mathfrak{G}$ and a parameter k
Output:     probably_true if $G_\alpha$ fixes k randomly selected points
            false otherwise
        counter := 0
        repeat
            select $\beta$ at random from $\Omega$
            if EQUALSTABILISERS($\alpha$,$\beta$) = false
                return(false)
            endif
        counter := counter + 1
        until counter = k
        return(probably_true)
end
```

Lemma 2.2.1.1 The probability that the algorithm outputs "**probably_true**" when a non-regular group is given in input is $\left(\frac{n - [\, N(G_\alpha):G_\alpha \,]}{n}\right)^k$.

Proof. We proved in corollary 1.8.9 that if Ω is a transitive G-set and α is a point of Ω, then the size of the set Δ of points fixed by G_α is $[N(G_\alpha):G_\alpha]$. Since the degree of our permutation group is **n**, the probability that a point β is not a fixed point of G_α is $\frac{n - [N(G_\alpha):G_\alpha]}{n}$.

2.2.2. A deterministic version.

We are going to show next how to guarantee a deterministic behaviour by selecting more carefully the points β from the set Ω.

For this purpose we exploit the fact that the set of points fixed by G_α forms a block: it is here that the *improved block finding algorithm* comes into play, thereby explaining the detailed presentation given in chapter 1.

The first time that we run the test we select the point β at random. If the outcome of the test EQUALSTABILISERS(α,β) is "**true**" we compute the minimal block of Ω containing both α and β, then we select a new point β' among the points of Ω not belonging to this block and repeat the test. If the outcome of the test is again "**true**" we extend the previous block to a bigger one containing β', select a new point β'' not belonging to the new block and repeat the test. We proceed in this way until the outcome of the test is "**false**" or there is no point left to be selected. The latter case is verified when the set of points fixed by G_α is the whole Ω, which implies that $G_\alpha = \{1\}$, i.e. the group is regular.

In the algorithm that follows Σ_α will denote the current block of Ω containing the point α.

Algorithm DETERMINISTIC_REGULAR

Input: the graph $\mathfrak{G}$ associated to (G,Ω,H) and a vertex α of $\mathfrak{G}$

Output: **true** if G acts regularly on Ω, **false** otherwise

```
    initialise $\Sigma$ to that partition of $\Omega$ whose classes are singletons
    repeat
        select $\beta$ from $\Omega - \Sigma_\alpha$
        if EQUALSTABILISERS($\alpha,\beta$) = false
            return(false)
        else
            EXTEND($\Sigma, \alpha, \beta$)
        endif
    until $\Omega = \Sigma_\alpha$
    return(true)
end
```

Let **t** denote the total number of times that the function EQUALSTABILISERS is called: since each new block has a size which is a multiple of the previous one, **t** is at most equal to the total number of prime divisors of $|\Omega|$.

Lemma 2.2.2.1 The execution time of the algorithm DETERMINISTIC_REGULAR is bounded above by $O(\mathbf{t\ m\ n} + \mathbf{m\ n}\ \alpha(\mathbf{n}))$.

Proof. During all the executions of EXTEND(Σ,α,β) at most $\mathbf{n} - \mathbf{1}$ UNION operations can be performed, since each one decreases the number of parts of Σ by 1. Thus a total of $\mathbf{n} - \mathbf{1}$ pairs are pushed into the stack C and so at most $\mathbf{2\ (n - 1)\ m}$ FIND operations are performed in all. Thus, according to [Tar75] the total time spent in all calls of EXTEND is $O(\mathbf{m\ n}\ \alpha(\mathbf{n}))$ where $\alpha(\mathbf{n})$ is a very small growing function related to the inverse of Ackerman's function. The other contribution comes from the time spent in the EQUALSTABILISERS function, and we have seen that EQUALSTABILISERS takes time $O(\mathbf{m\ n})$.

2.2.3. A simplified deterministic version.

To conclude this exposition we describe a deterministic variant of the previous algorithms which does not require the blocks framework at all.

To prove that G_α is trivial we now try to confirm the equality $G_\alpha = G_{\alpha^h}$ for all the generators **h** of G: it is easily seen this condition implies the equality $G_\alpha = G_{\alpha^g}$ for all the elements **g** of G, i.e. $G_\alpha = \{1\}$. Thus, instead of selecting each new β from $\Omega - \Sigma_\alpha$, we select it from $\{\alpha^h \mid h \in H\}$: this ensures that the number of times that the function EQUALSTABILISERS is called is always less than or equal to the number **m** of generators, giving an execution time which is in the worst case $O(m^2 n)$.

```
Algorithm   NO_BLOCKS_REGULAR
Input:      the graph $\mathbb{G}$ associated to $(G,\Omega,H)$ and a vertex $\alpha$ of $\mathbb{G}$
Output:     true if G acts regularly on $\Omega$, false otherwise
    $\Sigma := \{\alpha^h \mid h \in H\}$
    repeat
        delete some $\beta$ from $\Sigma$
        if EQUALSTABILISERS($\alpha,\beta$) = false
            return(false)
        endif
    until $\Sigma$ is empty
    return(true)
end
```

In the next section we will analyze more accurately the execution time of the algorithm DETERMINISTIC_REGULAR and we will show that the number t of times that the test EQUALSTABILISERS has to be repeated is usually very close to the minimal number of elements required to generate G: therefore the execution time of the algorithms

DETERMINISTIC_REGULAR and NO_BLOCKS_REGULAR is similar when the given set of generators **H** is a minimal or nearly minimal generating set. However, when the number of given generators is much larger than the minimal number required, DETERMINISTIC_REGULAR is a superior algorithm.

2.3. Expected time complexity for the deterministic version.

Let us suppose that the group **G** to be tested for regularity is regular, since our algorithm usually runs considerably faster for non regular groups. We would like to compute *the expected number of elements of* **G** *which have to be drawn at random before a set of generators is found*, since this quantity is related to the expected execution time of the algorithm DETERMINISTIC_REGULAR, as we are going to show. We start with some definitions, taken from [Hal36]:

Definition 2.3.1 An *n-basis of a group G* is defined as an ordered set $(x_1,x_2,...,x_n)$ of **n** elements[6] of **G** which generates **G**:

$$<(x_1,x_2,...,x_n)> = G$$

Definition 2.3.2 The number of distinct **n**-basis of **G** is denoted by $\phi_n(G)$ and is call:d *the* n^{th} *Eulerian function of G.*

Two important cases must be noticed:

(i) if **G** cannot be generated by **n** elements then $\phi_n(G) = 0$

(ii) if **G** is cyclic of order **m** then $\phi_n(G) = \phi(m)$, where $\phi(m)$ is the ordinary Eulerian function of an integer.

[6] The condi.ion $x_i \neq x_j$ if $i \neq j$ is not required.

Obviously an **n**-tuple $(g_1,g_2,...,g_n)$ of elements of G either generates G, that is $(g_1, g_2,...,g_n)$ constitutes an **n**-basis of G, or it generates a proper subgroup H of G, in which case it constitutes an **n**-basis of H. The total number of **n**-tuples $(g_1,g_2,...,g_n)$ of elements of G is $|G|^n$. We therefore have the *fundamental identity*:

$$|G|^d = \sum_{H \leq G} \phi_d(H) \tag{2.1}$$

Definition 2.3.3 Let $\lambda_d(G)$ denote *the probability that a d-tuple $(g_1,g_2,...,g_n)$ of elements of G chosen at random generates G.*

It is easy to see that

$$\lambda_d(G) = \frac{\phi_d(G)}{|G|^d} \tag{2.2}$$

Now we can define e(G) as *the expected number of elements of G which have to be drawn at random before a set of generators is found.* The probability that a sequence $g_1,g_2,...,g_{d-1},g_d$ of elements of G generates G and $g_1,g_2,...,g_{d-1}$ does not is $\lambda_d(G) - \lambda_{d-1}(G)$. Therefore

$$e(G) = \sum_{d=1}^{\infty} d\ (\lambda_d(G) - \lambda_{d-1}(G)) \tag{2.3}$$

Theorem 2.3.4 The expected execution time of the algorithm DETERMINISTIC_REGULAR when applied to a regular group G is $O(m\ n\ (\alpha(n) + e(G)))$.

Proof. If G is regular, its elements $h_1,h_2, ...,h_d$ generate G if and only if the smallest block containing $\alpha^{h_1},\alpha^{h_2},...,\alpha^{h_d}$ is Ω. Thus the expected number of elements of Ω that would have to be generated before the smallest block containing them is Ω itself is e(G). Since each new point β is chosen outside the block containing the previously chosen points, *the expected times that the algorithm DETERMINISTIC_REGULAR has to* be called is at most $\mathbf{e(G)}$. Since each execution of the algorithm DETERMINISTIC_REGULAR requires $\mathbf{O(m\ n)}$ time and the total time required to extend an initial block to the full set Ω is $\mathbf{O(m\ n\ \alpha(n))}$ the overall execution time is $\mathbf{O(m\ n\ \alpha(n) + m\ n\ e(G))}$.

The quantity $\mathbf{e(G)}$ is usually very difficult to compute. Kantor and Lubotzky have proved in [Kan90] that

$$\lim_{|G| \to \infty} \lambda_2(G) = 1$$

for all classical simple groups $\mathbf{G}$.

In the next chapter we will compute $\mathbf{e(G)}$ for some common families of groups.

3. Computation of the Eulerian function for some common families of groups.

In this section and in those to follow, we will show how to compute the Eulerian n^{th} function for some common classes of groups, starting from the very simple ones, the cyclic groups of prime order. Besides the Eulerian n^{th} function we will compute the expected number of elements which have to be drawn at random from the group to obtain a complete set of generators.

To compute the Eulerian n^{th} function of a group G we will try to solve for $\phi_n(G)$ the **fundamental identity** which we introduced in the last chapter

$$|G|^n = \sum_{H \leq G} \phi_n(H) \qquad (3.1)$$

The basic combinatorial identity that we will use through the computations is the following:

$$\sum_{d=1}^{\infty} \frac{d}{x^{d-1}} = \left(\frac{x}{x-1}\right)^2 \qquad (3.2)$$

where x is a real number stictly greater than one.

For the very first examples we will show the details of the computations. For the next classes of groups we will follow the same scheme.

3.1. The group of order one.

Consider first the group of order one since it is the base of our inductive construction: it is easily seen that for this group we have

$$\phi_d(\{\mathbf{1}\}) = 1$$

since all the **d**-tuples of elements from **{1}** are of the form (1,1,...,1).

It follows easily that

$$\lambda_d(\{\mathbf{1}\}) = 1$$

and that

$$e(\{\mathbf{1}\}) = 0$$

3.2. Groups of prime order.

The only subgroup of a group of prime order **p** is the trivial one. The fundamental identity (3.1) becomes

$$p^d = \phi_d(\{1\}) + \phi_d(G) = 1 + \phi_d(G)$$

from which it follows that

$$\phi_d(G) = p^d - 1$$

and then[7]

$$\lambda_d = \frac{p^d - 1}{p^d}$$

Let us now compute the expected number of elements which have to be drawn at random before a set of generators is found.

[7] An intuitive proof is the following: among all the possible d-tuples of elements of G, the only one that does not generate G is the tuple (1,1,...,1). Sinze there are p^d possible tuples, p^d- 1 of them generate G, hence the result.

$$
\begin{aligned}
e(G) &= \sum_{d=1}^{\infty} d\,(\lambda_d(G) - \lambda_{d-1}(G)) = \sum_{d=1}^{\infty} d\left(\frac{p^d-1}{p^d} - \frac{p^{d-1}-1}{p^{d-1}}\right) = \sum_{d=1}^{\infty} d\,\frac{p-1}{p^d} \\
&= (p-1)\sum_{d=1}^{\infty}\frac{d}{p^d} \qquad = \frac{p-1}{p}\sum_{d=1}^{\infty}\frac{d}{p^{d-1}} \qquad = \frac{p-1}{p}\cdot\frac{p^2}{(p-1)^2} \\
&= \frac{p}{p-1} \qquad = 1 + \frac{1}{p-1}
\end{aligned}
$$

The result $e(G) = 1 + \frac{1}{p-1}$ agrees with our intuition.

3.3. P-groups.

In this section we will address the problem of computing the probability of generating an arbitrary **p**-group. First we will show how to reduce this problem to that of computing the probability of generating an elementary abelian **p**-group. For this purpose we start with a consideration valid for arbitrary groups:

Lemma 3.3.1 Let **G** be an arbitrary group and **Φ(G)** its Frattini subgroup. Then **r** elements $x_1,\ldots,x_r$ of **G** generate **G** if and only if their images[8] in **G/Φ(G)** generate **G/Φ(G)**.

Proof. The proof of this lemma is a standard one and can be found for example in [Dix67, problem 8.7].

Lemma 3.3.2 If **G** is a **p**-group with minimal number of generators **d** then **G/Φ(G)** is an elementary abelian group of order p^d.

Proof. Again, the proof of this lemma is a standard one and can be found for example in [Dix67, problem 8.26].

[8] That is, the cosets $\Phi(G)x_1,\ldots,\Phi(G)x_r$.

We can now use the information contained in the two lemmas above to compute the probability of generating an arbitrary **p**-group with minimal number of generators **d.** To start, we recall the fact that an elementary abelian group of order $\mathbf{p^d}$ can be considered as a vector space of dimension **d** over **GF(p)**. In the light of this equivalence we introduce a new quantity:

$\mu_{r,s}$ is defined as *the probability that a sequence of* **r** *elements* $x_1,\dots,x_r$ *drawn from a space of dimension* **d** *generates a subspace of dimension* **s**.

Lemma 3.3.3

(i) $\mu_{r,0} = p^{-rd}$

(ii) $\mu_{r,s} = 0$ if $r < s$

(iii) $\mu_{r,s} = \mu_{r-1,s}\frac{p^s}{p^d} + \mu_{r-1,s-1}\left(1 - \frac{p^{s-1}}{p^d}\right)$

Proof.

(i) **r** elements chosen at random span a subspace of dimension zero in a space of dimension **d** over **GF(p)** if and only if they are all equal to the null vector. But the probability that this event occurs is equal to $(\frac{1}{p^d})^r$.

(ii) it is obvious that **r** elements cannot generate a space of dimension s if **r** < s.

(iii) $\mu_{r,s}$ = Prob(dim $<x_1 ,\dots, x_r> = s$ and dim $<x_1 ,\dots, x_{r-1}> = s$) + Prob(dim $<x_1 ,\dots, x_r> = s$ and dim $<x_1 ,\dots, x_{r-1}> = s-1$) = Prob(dim $<x_1 ,\dots, x_{r-1}> = s$ and $x_r \in <x_1 ,\dots, x_{r-1}>$) + Prob(dim $<x_1 ,\dots, x_{r-1}> = s-1$ and $x_r \notin <x_1 ,\dots, x_{r-1}>$) =
$\mu_{r-1,s}\frac{p^s}{p^d} + \mu_{r-1,s-1}\left(1 - \frac{p^{s-1}}{p^d}\right)$

Corollary 3.3.4 $\mu_{s,s} = \prod_{i=d-s+1}^{d}(1 - p^{-i})$

Proof.

$$\begin{aligned}
\mu_{s,s} &= \mu_{s-1,s-1}\left(1 - \frac{p^{s-1}}{p^d}\right) + \mu_{s-1,s}\frac{p^s}{p^d} \\
&= \mu_{s-1,s-1}\left(1 - \frac{p^{s-1}}{p^d}\right) \\
&= \mu_{s-1,s-1}\left(1 - p^{s-1-d}\right) \\
&= \mu_{s-2,s-2}\left(1 - p^{s-2-d}\right)\left(1 - p^{s-1-d}\right) \\
&= (1 - p)(1 - p^{-2})\ldots(1 - p^{s-2-d})(1 - p^{s-1-d})
\end{aligned}$$

as required.

Corollary 3.3.5 $$\mu_{s+k,s} = \frac{\mu_{ss}}{p^{dk}}\prod_{i=1}^{k}\frac{p^{s+i} - 1}{p^{i} - 1}$$

Proof. By induction on **k**. For **k** = 0 the result follows from the last corollary. For **k** > 0 we can write

$$\begin{aligned}
&\mu_{s+k+1,s} = \\
&= \mu_{s+k,s}\frac{p^s}{p^d} + \mu_{s+k,s-1}\left(1 - \frac{p^{s-1}}{p^d}\right) \\
&= \frac{p^s}{p^d}\frac{\mu_{ss}}{p^{dk}}\prod_{i=1}^{k}\frac{p^{s+i} - 1}{p^{i} - 1} + \left(1 - p^{s-1-d}\right)\frac{\mu_{s-1.s-1}}{p^{d(k+1)}}\prod_{i=1}^{k+1}\frac{p^{s-1+i} - 1}{p^{i} - 1} \\
&= \frac{p^s\mu_{ss}}{p^{d(k+1)}}\prod_{i=1}^{k}\frac{p^{s+i} - 1}{p^{i} - 1} + \frac{\mu_{ss}}{p^{d(k+1)}}\prod_{i=1}^{k+1}\frac{p^{s-1+i} - 1}{p^{i} - 1} \\
&= \frac{\mu_{ss}}{p^{d(k+1)}}\prod_{i=1}^{k}\frac{p^{s+i} - 1}{p^{i} - 1}\left(p^{s} + \frac{p^{s} - 1}{p^{k+1} - 1}\right) \\
&= \frac{\mu_{ss}}{p^{d(k+1)}}\prod_{i=1}^{k+1}\frac{p^{s+i} - 1}{p^{i} - 1}
\end{aligned}$$

The lemma that follows is the most important of this section. In fact, it allows us to effectively compute the probability that some elements chosen independently and at random generate a **p**-group with minimal number of generators **d**.

Lemma 3.3.6 If G is a **p**-group with minimal number of generators **d**, then

(i) $$\lambda_d(G) = \prod_{i=1}^{d}(1 - p^{-i})$$

(ii) if $k \geq 0$, then $$\lambda_{d+k}(G) = \lambda_d(G) \cdot \prod_{i=1}^{k} \frac{p^i - p^{-d}}{p^i - 1}$$

Proof. Part (i) of the lemma follows from the fact that $\lambda_d(G) = \mu_{d,d}$ and part (ii) from the fact that $\lambda_{d+k}(G) = \mu_{d+k,d}$.

The formulae given above are quite complicated: the next theorem gives a handy estimate for $\lambda_d(G)$, and also shows that $\lim_{p\to\infty} \lambda_d(G) = 1$.

Theorem 3.3.7 If G is a **p**-group with minimal number of generators **d**, then

$$\frac{p - 1}{p} \geq \lambda_d(G) \geq 1 - p^{-1} - p^{-2}$$

Proof. The upper bound follows from the expression for $\lambda_d(G)$ given in Lemma 3.3.6.

To prove the lower bound, we see that :

$$\lambda_d(G) = \prod_{i=1}^{d}(1 - p^{-i}) \geq \prod_{n=1}^{\infty}(1 - p^{-n})$$

By Euler's formula [Knu73, p.20] :

$$\prod_{n=1}^{\infty}(1 - z^n) = 1 - z - z^2 + z^5 + z^7 - z^{12} - z^{15} + \dots = \sum_{-\infty < j < \infty}(-1)^j z^{(3j^2+j)/2}$$

if we put $z = p^{-1}$ we obtain :

$$\prod_{n=1}^{\infty}(1 - p^{-n}) = 1 - p^{-1} - p^{-2} + p^{-5} + p^{-7} - p^{-12} - p^{-15} + \dots$$

this can be considered as an alternating series, if we collect each even term with the consecutive one; if we take the first three terms by Liebniz's theorem the error will be negative and less than $p^{-5} + p^{-7}$ in absolute value.

Note 3.3.8 The lower bound for $\lambda_d(G)$ given in Theorem 3.3.7 is also valid for $\lambda_{d+k}(G)$, since $\lambda_d(G) = \mu_{d,d} \leq \mu_{d+k,d} = \lambda_{d+k}(G)$.

As an example of the numerical effectiveness of this approximation, for a four generator p-group with p=7, by Lemma 3.3.6 we have $\lambda_4(G) = \frac{236390400}{282475249} \approx 0.8368$, while Theorem 3.3.7 gives $0.8367 \approx \frac{41}{49} \leq \lambda_4(G) \leq \frac{6}{7} \approx 0.8571$.

3.3.1. Computation of a presentation for G/Φ(G) when G is a p-group given by generators and relations.

Let us suppose that G is a p-group, for which a presentation is given. We would like to compute the quotient group of G with respect to its Frattini subgroup. The following theorems will prove very useful.

Theorem 3.3.1.1 If N is the minimal normal subgroup with the property that G/N is elementary abelian, then $N = \Phi(G)$

Proof. Let M be maximal in G. Then M is normal, since G is a p-group, so G/M is elementary abelian, and then by hypothesis $N \leq M$. This shows that N is contained in the Frattini subgroup of G, since it is contained in all the maximal subgroups of G. Conversely, consider $G/N = A_1/N \times A_2/N \times ... \times A_k/N$, where each A_i/N has order p. Let $B_i/N = \times_{j\neq i} A_j/N$. This group is easily seen to be maximal. Clearly $\bigcap(B_i/N)$ is equal to the identity in G/N, from which it follows that $\bigcap B_i = N$. But then N contains the Frattini subgroup of G.

Theorem 3.3.1.2 A presentation for $G/\Phi(G)$ is obtained by adding the following relations to the given ones:

$[x,y] = 1$	for all generators x and y
$x^p = 1$	for all generators x

Proof. Let K be the minimal normal subgroup containing $[x,y]$ and x^p for all generators x and y. G/K is elementary abelian, since $[xK , yK] = 1$ in G/K and $(xK)^p = x^pK = 1$ in G/K. But then $\Phi(G) \leq K$, since by the previous theorem $\Phi(G)$ is the minimal normal subgroup N of G with the property that G/N is elementary abelian. Conversely, since $G/\Phi(G)$ is elementary abelian, $[x,y] \in \phi(G)$, $x^p \in \Phi(G)$. This shows that $K \leq \Phi(G)$.

3.3.2. Application to the dihedral groups of order 2^n.

Let us consider the dihedral group G of order 2^n. A presentation for G is the following

$$G = \langle a,\ b \mid a^{2^{n-1}} = b^2 = (ab)^2 = 1 \rangle$$

By imposing the relations

$$a^2 = 1 \text{ , } [a,b] = 1$$

we obtain a presentation for $G/\Phi(G)$

$$\langle a , b \mid a^2 = b^2 = [a,b] = 1 \rangle$$

and it is easily seen that this defines a group isomorphic to the Klein four-group V_4. Therefore, for all the groups belonging to this class we obtain

$$\lambda_d(D_{2n}) = \frac{4^d - 3 \cdot 2^d + 2}{4^d}$$

and

$$e(D_{2n}) = \frac{10}{3} = 2 + \frac{4}{3}$$

To compute the d^th^ Eulerian function we multiply the order of the group raised to the d^th^ power by the d^th^ lambda function

$$\phi_d(D_{2n}) = (2^n)^d \cdot \frac{4^d - 3 \cdot 2^d + 2}{4^d}$$

3.3.3. The generalized quaternion groups, Q_n

A generalized quaternion group or dicyclic group Q_n is a group of order 2^n generated by two elements **a** and **b** which satisfy the relations

$$a^{2^{n-2}} = b^2 \ ; \quad aba = b$$

such a group can be considered as a generalization of the quaternion group Q. Being Q_n a 2-group with minimal number of generators 2, its behaviour is the same as the dihedral groups of order 2^n, that is

$$\phi_d(Q_n) = 8^d - 3 \cdot 4^d + 2^{d+1} \ . \qquad \lambda_d(Q_n) = \frac{8^d - 3 \cdot 4^d + 2^{d+1}}{8^d}$$

and

$$e(Q_n) = \frac{10}{3} = 2 + \frac{4}{3}$$

3.4. Groups of order p^2, p prime.

It is well known that a group of order p^2 must be abelian. Furthermore, such a group can be either cyclic or the direct product of two cyclic groups of order **p**. Let us examine in detail the two cases. In the following discussion we will not use the general results about the probability of generating a **p**-group, but instead we will compute the required quantities after looking at the subgroup lattice structure.

3.4.1. Cyclic groups of order p^2.

The subgroup lattice structure of these groups is very simple: other than the trivial subgroups these groups have just one subgroup, which is of order **p**.

The fundamental identity (3.1) becomes

$$|C_{p2}|^d = (p^2)^d = \phi_d(\{1\}) + \phi_d(C_p) + \phi_d(C_{p2}) = 1 + (p^d - 1) + \phi_d(C_{p2})$$

from which it follows that

$$\phi_d(C_{p^2}) = p^d\,(p^d - 1)$$

The quantity $\lambda_d(C_{2p})$ is easily seen to be

$$\lambda_d(C_{p^2}) = \frac{\phi_d(C_{p^2})}{|C_{p^2}|^d} = \frac{p^d - 1}{p^d}$$

Hence

$$e(C_{p^2}) = \sum_{d=1}^{\infty} d\,(\lambda_d(C_{p^2}) - \lambda_{d-1}(C_{p^2}))$$

$$= \sum_{d=1}^{\infty} d\,\left(\frac{p^d - 1}{p^d} - \frac{p^{d-1} - 1}{p^{d-1}}\right) \qquad = \sum_{d=1}^{\infty} d\,\frac{p - 1}{p^d}$$

By using the combinatorial identity (3.2) this expression becomes

$$e(C_{p^2}) = \frac{p}{p - 1} = 1 + \frac{1}{p - 1}$$

3.4.2. Elementary abelian groups of order p^2.

We begin by deriving the subgroup lattice structure. This is easily done: in fact we cannot have any element of order $\mathbf{p^2}$ (otherwise the group would be cyclic) and therefore by Lagrange's theorem the order of all non identity elements must be $\mathbf{p}$. Since we have $\mathbf{p^2}$ elements, $\mathbf{p^2 - 1}$ of which are non identity of order $\mathbf{p}$, and since the intersection of two distinct subgroups of order $\mathbf{p}$ must be the identity, we have a total of $\mathbf{p + 1}$ subgroups of order $\mathbf{p}$.

The fundamental identity (3.1) becomes

$$|C_p \times C_p|^d = (p^2)^d \qquad = \phi_d(\{1\}) + (p + 1)\,\phi_d(C_p) + \phi_d(C_p \times C_p)$$

$$= 1 + (p + 1)(p^d - 1) + \phi_d(C_p \times C_p)$$

from which it follows that

$$\phi_d(C_p \times C_p) = p^{2d} - p^{d+1} - p^d + p$$

and

$$\lambda_d(C_p \times C_p) = \frac{\phi_d(C_p \times C_p)}{|C_p \times C_p|^d} = \frac{p^{2d} - p^{d+1} - p^d + p}{p^{2d}}$$

Hence

$$\begin{aligned} e(C_p \times C_p) &= \sum_{d=1}^{\infty} d \left(\lambda_d(C_p \times C_p) - \lambda_{d-1}(C_p \times C_p) \right) \\ &= \sum_{d=1}^{\infty} d \left(\frac{p^{2d} - p^{d+1} - p^d + p}{p^{2d}} - \frac{p^{2(d-1)} - p^d - p^{d-1} + p}{p^{2(d-1)}} \right) \\ &= \sum_{d=1}^{\infty} d \, \frac{p^{d+2} - p^d - p^3 + p}{p^{2d}} \end{aligned}$$

By using the combinatorial identity (3.2) this expression becomes

$$e(C_p \times C_p) = 2 + \frac{p + 2}{p^2 - 1}$$

3.5. Groups of order 2p, p prime greater than two.

It is known that a group of order **2p**, **p** prime greater than two, must be one of the following:

i) Cyclic C_{2p}

ii) Dihedral D_{2p}

Their subgroup lattice structure is different, and therefore we have to consider the two cases separately.

3.5.1. Cyclic groups of order 2p.

A cyclic group of order **n** has exactly one subgroup of order **d** for each divisor of its order. Therefore, in addition to the trivial subgroups we have a subgroup of order **2** and a subgroup of order **p**.

The fundamental identity (3.1) becomes

$$\begin{aligned} |C_{2p}|^d = (2p)^d &= \phi_d(\{1\}) + \phi_d(C_2) + \phi_d(C_p) + \phi_d(C_{2p}) \\ &= 1 + (2^d - 1) + (p^d - 1) + \phi_d(C_{2p}) \end{aligned}$$

from which it follows that

$$\phi_d(C_{2p}) = (2p)^d - 2^d - p^d + 1$$

and

$$\lambda_d(C_{2p}) = \frac{\phi_d(C_{2p})}{|C_{2p}|^d} = \frac{(2p)^d - 2^d - p^d + 1}{(2p)^d}$$

Hence

$$\begin{aligned} e(C_{2p}) &= \sum_{d=1}^{\infty} d\,(\,\lambda_d(C_{2p}) - \lambda_{d-1}(C_{2p})\,) \\ &= \sum_{d=1}^{\infty} d\,\left(\frac{(2p)^d - 2^d - p^d + 1}{(2p)^d} - \frac{(2p)^{d-1} - 2^{d-1} - p^d + 1}{(2p)^{d-1}}\right) \\ &= \sum_{d=1}^{\infty} d\,\frac{p^d + 2^d\,(p-1) - 2p + 1}{(2p)^d} \end{aligned}$$

After simplifying this expression, by using the combinatorial identity (3.2), we obtain

$$e(C_{2p}) = 1 + \frac{2p^2 - 2p + 1}{2p^2 - 3p + 1} = 2 + \frac{p}{2p^2 - 3p + 1}$$

3.5.2. Dihedral groups of order 2p

It is known that these groups have one subgroup of order **p** and **p** subgroups of order two, other than the trivial subgroups, of course.

The fundamental identity (3.1) becomes

$$\begin{aligned} |D_{2p}|^d &= (2p)^d = \phi_d(\{1\}) + p\,\phi_d(C_2) + \phi_d(C_p) + \phi_d(D_{2p}) \\ &= 1 + p\,(2^d - 1) + (p^d - 1) + \phi_d(D_{2p}) \end{aligned}$$

from which it follows that[9]

$$\phi_d(D_{2p}) = (2p)^d - p\,2^d - p^d + p$$

The quantity $\lambda_d(D_{2p})$ is easily seen to be

$$\lambda_d(D_{2p}) = \frac{\phi_d(D_{2p})}{|D_{2p}|^d} = \frac{(2p)^d - p\,2^d - p^d + p}{(2p)^d}$$

Hence

$$\begin{aligned} e(D_{2p}) &= \sum_{d=1}^{\infty} d\,(\,\lambda_d(D_{2p}) - \lambda_{d-1}(D_{2p})\,) \\ &= \sum_{d=1}^{\infty} d\,\left(\frac{(2p)^d - p\,2^d - p^d + p}{(2p)^d} - \frac{(2p)^{d-1} - p\,2^{d-1} - p^d + p}{(2p)^{d-1}}\right) \\ &= \sum_{d=1}^{\infty} d\,\frac{p^d - 2^d\,p + p + 2^d\,p^2 - 2\,p^2}{(2p)^d} \end{aligned}$$

After simplifying this expression, by using the combinatorial identity (3.2), we obtain :

$$e(D_{2p}) = 2 + \frac{p^2}{2\,p^2 - 3\,p + 1}$$

[9] Note that $\phi_1(D_{2p}) = 0$, since in order to generate a dihedral group at least two elements are required.

3.6. On the direct product of two groups.

Let G and H be two finite groups. If we knew $\phi_n(G)$ and $\phi_n(H)$, what could we say about $\phi_n(G \times H)$? Or otherwise, if we knew $\lambda_n(G)$ and $\lambda_n(H)$, what could we say about $\lambda_n(G \times H)$? Generally speaking, very little, but if the two groups have coprime order then the latter quantity is exactly equal to the product of the two former ones, as we are going to show.

Theorem 3.6.1 Let G and H be two finite groups of coprime order. Let $x_i = (g_i , h_i)$, $g_i \in G$, $h_i \in H$. Then $x_1,...,x_d$ generate $G \times H$ if and only if $g_1,...,g_d$ generate G and $h_1,...,h_d$ generate H.

Proof.

⇒ This is true even without the assumption that |G| and |H| are coprime, since the homomorphism that maps an x_i into the corresponding g_i (resp h_i), i.e. the projection homomorphism, is onto.

⇐ Let $x_i = (g_i, h_i)$, $g_i \in G$, $h_i \in H$. Then $x_i^{k(i)} = (g_i^{k(i)}, h_i^{k(i)})$. We can choose $k(i)$ so that $k(i)$ is the order of g_i, and by hypothesis $k(i)$ is coprime with the order of h_i. But then h_i and $h_i^{k(i)}$ generate the same group, and therefore

$$\langle x_i^{k(i)} \rangle = \langle h_i \rangle$$

It follows that:

$$\langle x_1,...,x_d \rangle \geq \langle x_1^{k(1)},...,x_d^{k(d)} \rangle = \langle h_1,...,h_d \rangle = H$$

Using the same argument it is possible to prove that $\langle x_1,...,x_d \rangle \geq G$. By combining the two inclusions it is shown that $\langle x_1,...,x_d \rangle \geq G \times H$. Since it is obvious that $\langle x_1,...,x_d \rangle \leq G \times H$, the theorem follows.

Application 3.6.2 A finite group is called *nilpotent* if and only if it is the direct product of its Sylow subgroups. The class of nilpotent groups includes among others the class of all abelian groups. From the discussion in the previous sections we know how to effectively compute the functions $\phi_n(G)$ and $\lambda_n(G)$ when G is a p-group. Let G be a nilpotent group which is the direct product of k Sylow subgroups:

$$G = H_1 \times H_2 \times \ldots \times H_k$$

Then we have:

$$\phi_n(G) = \phi_n(H_1) \cdot \phi_n(H_2) \cdot \ldots \cdot \phi_n(H_k)$$

and

$$\lambda_n(G) = \lambda_n(H_1) \cdot \lambda_n(H_2) \cdot \ldots \cdot \lambda_n(H_k)$$

Example 3.6.3 Consider the group C_{36}. We know that $C_{36} \cong C_4 \times C_9$. According to the previous sections we have:

$$\phi_d(C_4) = 2^d (2^d - 1) \qquad \phi_d(C_9) = 3^d (3^d - 1)$$

$$\lambda_d(C_4) = \frac{2^d - 1}{2^d} \qquad \lambda_d(C_9) = \frac{3^d - 1}{3^d}$$

Therefore

$$\phi_d(C_{36}) = 2^d (2^d - 1) 3^d (3^d - 1)$$

and

$$\lambda_d(C_{36}) = \frac{(2^d - 1)(3^d -1)}{2^d\, 3^d}$$

3.7. Groups of order pq, p and q primes.

A group of order pq, with p and q primes, q less than p, must be of one of the two following types:

- cyclic
- non abelian and metacyclic

The first case may happen for any values of **p** and **q**, since there is a cyclic group of any given order. The second case can happen only if **q** divides **p - 1**, i.e. a group of order **pq**, with **p** and **q** primes, **q** less than **p**, and **q** not dividing **p - 1** must be necessarily cyclic. The structure of these groups is very simple and therefore their Eulerian functions may be calculated.

3.7.1. Cyclic groups of order pq, p and q primes.

A cyclic subgroup of order **n** is known to have one and only one subgroup of order **d** for each divisor **d** of **n**, therefore C_{pq} possesses one subgroup of order **p** and one of order **q**, in addition to the trivial subgroup.

To compute its Eulerian function we use the general result for the direct product of two groups of coprime order. We know from the past sections that for a cyclic group of prime order **p** we have

$$\phi_d(C_p) = p^d - 1 \qquad \text{and} \qquad \lambda_d(C_p) = \frac{p^d - 1}{p^d}$$

Therefore for a cyclic group of order **pq** we obtain

$$\phi_d(C_{pq}) = (p^d - 1)(q^d - 1) \qquad \text{and} \qquad \lambda_d(C_{pq}) = \frac{(p^d - 1)(q^d - 1)}{p^d\, q^d}$$

It is easy now to compute $e(C_{pq})$.

$$e(C_{pq}) = \sum_{d=1}^{\infty} d \left(\frac{(p^d - 1)(q^d - 1)}{p^d\, q^d} - \frac{(p^{d-1} - 1)(q^{d-1} - 1)}{p^{d-1} q^{d-1}} \right)$$

$$= \sum_{d=1}^{\infty} d\, \frac{(q - 1)\, p^d + (p - 1)\, q^d - p\, q + 1}{(p\, q)^d}$$

After simplification, by using the combinatorial identity (3.2), this expression becomes

$$e(C_{pq}) = 1 + \frac{1}{p - 1} + \frac{1}{q - 1} - \frac{1}{p\, q - 1}$$

3.7.2. Non abelian groups of order pq, p and q primes.

If $\mathbf{M_{pq}}$ is a non abelian group of order **pq**, with $\mathbf{q < p}$, then in addition to the trivial subgroup, it has a normal subgroup of order **p** and **p** subgroups of order **q**.

The fundamental equation (3.1) becomes now

$$\begin{aligned} |M_{pq}|^d &= (pq)^d = \phi_d(\{1\}) + \phi_d(C_p) + p\,\phi_d(C_q) + \phi_d(M_{pq}) \\ &= 1 + (p^d - 1) + p\,(q^q - 1) + \phi_d(M) \end{aligned}$$

from which it follows that

$$\phi_d(M_{pq}) = (p\,q)^d - p^d - p\,q^d + p$$

and

$$\lambda_d(M_{pq}) = \frac{(p\,q)^d - p^d - p\,q^d + p}{(p\,q)^d}$$

Therefore

$$\begin{aligned} e(M_{pq}) &= \sum_{d=1}^{\infty} d\ \left(\frac{(p\,q)^d - p^d - p\,q^d + p}{(p\,q)^d} - \frac{(p\,q)^{d-1} - p^{d-1} - p\,q^{d-1} + p}{(p\,q)^{d-1}} \right) \\ &= \sum_{d=1}^{\infty} d\ \frac{-p^d + p - p\,q^d + p^d\,q - p^2\,q + p^2\,q^d}{(p\,q)^d} \end{aligned}$$

After simplification, by using the combinatorial identity (3.2), this expression becomes

$$e(M_{pq}) = 2 + \frac{1}{q - 1} + \frac{1}{p - 1} - \frac{p}{p\,q - 1}$$

Example 3.7.2.1 For the non cyclic group of order 21 we obtain $e(M_{21}) \approx 2 + 0.31$

3.8. Groups of small order.

In this section we will show that it is possible to compute the quantities ϕ_d, λ_d and e for all the groups of order less than sixteen by using the methods discussed in the previous sections. In fact, the group of order 1 is dealt with in section 3.1, the groups of order **2,3,5,7,11,13** are dealt with in section 3.2, the groups of order **4,9** are dealt with in section 3.4 and the groups of order **6,10,14** are dealt with in section 3.5. We are left now with the groups of order **8** and **12**.

It is known that there are five groups of order eight, and they are: $\mathbf{C_8}$, $\mathbf{C_4 \times C_2}$, the Quaternion group **Q**, $\mathbf{D_8}$ and $\mathbf{C_2 \times C_2 \times C_2}$.

The first group that we consider is $\mathbf{C_8}$: this is a **2**-group with minimal number of generators equal to one. Therefore its behaviour is the same as C_2, that is

$$\lambda_d(C_8) = \lambda_d(C_2) = \frac{2^d - 1}{2^d}, \qquad e(C_8) = e(C_2) = 1 + 1$$

and

$$\phi_d(C_8) = \lambda_d(C_8)\cdot 8^d = \frac{2^d - 1}{2^d}\cdot 8^d = 8^d - 4^d$$

The next three groups, $\mathbf{C_4 \times C_2}$, the quaternion group **Q** and the dihedral group $\mathbf{D_8}$ are **2**-groups with minimal number of generators equal to two. Therefore the behaviour of each of these group is the same as $\mathbf{V_4}$, that is

$$\lambda_d(G) = \frac{4^d - 3\cdot 2^d + 2}{4^d}, \qquad e(G) = 2 + \frac{4}{3}$$

and

$$\phi_d(G) = \lambda_d(G)\cdot 8^d = 8^d - 3\cdot 4^d + 2^{d+1}$$

The last group considered, the elementary abelian group of order eight, is known to have seven subgroups of order two and seven subgroups of order four, isomorphic to the Klein four group, in addition to the trivial subgroup. By applying the fundamental identity (3.1) we obtain

$$8^d = \phi_d(\{1\}) + 7\,\phi_d(C_2) + 7\,\phi_d(V_4) + \phi_d(C_2 \times C_2 \times C_2)$$

from which it follows that

$$\phi_d(C_2 \times C_2 \times C_2) = 8^d - 7{\cdot}4^d + 14{\cdot}2^d - 8$$

By applying the equations (2.2) and (2.3) we obtain

$$e(C_2 \times C_2 \times C_2) = \sum_{d=1}^{\infty} d \, \frac{7{\cdot}4^d - 42{\cdot}2^d + 56}{8^d}$$

after simplification, by using the combinatorial identity (3.2), this expression becomes

$$e(C_2 \times C_2 \times C_2) = \frac{94}{21} \approx 3 + 1.47$$

Before talking about the groups of order twelve, we introduce a theorem which proves to be quite useful for our purposes.

Theorem 3.8.1 If a finitely generated group G is the direct product of two subgroups A and B then the Frattini subgroup of G is isomorphic to the direct proauct of the Frattini subgroup of A and the Frattini subgroup of B.

Proof. See [Dix67, problem 8.22].

Let us consider now the groups of order twelve. It is known that there are five of them, and they are: C_{12}, $C_2 \times C_2 \times C_3$, A_4, D_{12} and the group $T = \langle a,b \mid a^6 = 1,\ b^2 = a^3 = (ab)^2 \rangle$.

Let us consider first the cyclic group of order twelve. Since $C_{12} \cong C_4 \times C_3$, according to Theorem 3.8.1 we have

$$\Phi(C_{12}) = \Phi(C_4 \times C_3) \cong \Phi(C_4) \times \Phi(C_3) \cong C_2 \times \{1\} = C_2$$

and

$$C_{12}/\Phi(C_{12}) \cong C_6.$$

Therefore, by Lemma 3.3.1 the behaviour of C_{12} is the same as C_6, that is

$$\lambda_d(C_{12}) = \frac{6^d - 2^d - 3^d + 1}{6^d} \quad , \qquad e(C_{12}) = 2 + 0.3$$

and

$$\phi_d(C_{12}) = \lambda_d(C_{12}) \cdot 12^d = 2^d\,(6^d - 2^d - 3^d + 1)$$

The next group to consider is $C_2 \times C_2 \times C_3$. This group is nilpotent, since it is the direct product of its Sylow subgroups, which are isomorphic to V_4 and C_3. Therefore, according to the Theorem 3.6.1 we obtain

$$\phi_d(V_4{\times}C_3) = (2^{2d} - 2^{d+1} - 2^d + 2)(3^d - 1) = 12^d - 3{\cdot}6^d - 4^d + 2{\cdot}3^d + 3{\cdot}2^d - 2$$

By applying the equations (2.2) and (2.3) we obtain

$$e(V_4 \times C_3) = \sum_{d=1}^{\infty} d\ \frac{3{\cdot}6^d + 2{\cdot}4^d - 6{\cdot}3^d - 15{\cdot}2^d + 22}{12^d}$$

after simplification, by using the combinatorial identity (3.2), this expression becomes

$$e(V_4 \times C_3) = 6 + \frac{3}{2} - \frac{8}{3} + \frac{24}{11} - \frac{18}{5} \approx 2 + 1.41$$

The third group to consider is the alternating group on four symbols. The subgroup structure of A_4 is well known: besides the trivial subgroups there are four subgroups of order three, one subgroup isomorphic to the Klein four group and three subgroups of order two. The fundamental identity (3.1) becomes now

$$12^d = \phi_d(\{1\}) + 4\,\phi_d(C_3) + 3\,\phi_d(C_2) + \phi_d(V_4) + \phi_d(A_4)$$

from which it follows that

$$\phi_d(A_4) = 12^d - 4{\cdot}3^d - 4^d + 4$$

By applying the equations (2.2) and (2.3) we obtain

$$e(A_4) = \sum_{d=1}^{\infty} d\ \frac{2{\cdot}4^d + 12{\cdot}3^d - 44}{12^d}$$

after simplification, by using the combinatorial identity (3.2), this expression becomes

$$e(A_4) = \frac{3}{2} + \frac{16}{3} - \frac{48}{11} = 2 + 0.46$$

The fourth group to consider is the dihedral group of order twelve: this group is known to have, besides the trivial subgroups: one cyclic subgroup of order six, two dihedral subgroups of order six, one subgroup of order three, seven subgroups of order two and three subgroups of order four isomorphic to the Klein four-group.The fundamental identity (3.1) becomes now

$$12^d = \phi_d(\{1\}) + \phi_d(C_3) + 7\,\phi_d(C_2) + 3\,\phi_d(V_4) + \phi_d(C_6) + \phi_d(D_{12})$$

from which we obtain

$$\phi_d(D_{12}) = 12^d - 3\cdot 6^d + 2\cdot 3^d - 3\cdot 4^d + 9\cdot 2^d - 6$$

By applying the equations (2.2) and (2.3) we obtain

$$e(D_{12}) = \sum_{d=1}^{\infty} d\ \frac{66 + 3\cdot 6^d - 6\cdot 3^d + 6\cdot 4^d - 45\cdot 2^d}{12^d}$$

after simplification, by using the combinatorial identity (3.2), this expression becomes

$$e(D_{12}) = \frac{72}{11} - \frac{8}{3} + \frac{9}{2} - \frac{54}{5} + 6 = \frac{1181}{330} \approx 2 + 1.57$$

The last group of order twelve to consider is the group **T**, isomorphic to the group generated by the two matrices $\begin{pmatrix}0 & i\\ i & 0\end{pmatrix}$ and $\begin{pmatrix}\varepsilon & 0\\ 0 & \varepsilon^2\end{pmatrix}$ where $i = \sqrt{-1}$ and ε is a non real complex cubic root of the unity. Besides the trivial subgroups T has three cyclic subgroups of order four, a cyclic subgroup of order six, a cyclic subgroup of order two and a cyclic subgroup of order three. The fundamental equation becomes now

$$12^d = \phi_d(\{1\}) + 3\,\phi_d(C_4) + \phi_d(C_6) + \phi_d(C_2) + \phi_d(C_3) + \phi_d(T)$$

from which we obtain

$$\phi_d(T) = 12^d - 6^d - 3\cdot 4^d + 3\cdot 2^d$$

By applying the equations (2.2) and (2.3) we obtain

$$e(T) = \sum_{d=1}^{\infty} d \; \frac{6^d + 6 \cdot 4^d - 15 \cdot 2^d}{12^d}$$

after simplification, by using the combinatorial identity (3.2), this expression becomes

$$e(T) = \frac{20}{10} + \frac{45}{10} - \frac{36}{10} = 2 + 0.9$$

3.9. The groups PSL(2,k).

In the discussion that follows all the fields will be finite, which implies that any field considered here will have $\mathbf{p^n}$ elements, for some prime **p** and some positive integer **n**. To introduce the groups **PSL(2,k)** we have to introduce some groups of matrices from which **PSL(2,k)** is obtained. First we define the *general linear group GL(m,k)* as the multiplicative group of all nonsingular **m** × **m** matrices over **GF(k)**. Then we define the *special linear group SL(m,k)* as the multiplicative group of all **m** × **m** unimodular matrices (i.e. having determinant equal to one) over **GF(k)**. Finally we define the *projective unimodular group PSL(m,k)* as the group **SL(m,k)/Z(m,k)** where **Z(m,k)** is the centre of **SL(m,k)**.

The order of **PSL(m,k)** is known to be $\mathbf{(k+1)(k^2-k)}$ if **k** is a power of two, $\mathbf{\frac{(k+1)(k^2-k)}{2}}$ if **k** is a power of an odd prime.

The groups **PSL(m,k)** are known to be simple except for a small number of exceptions. In this section we focus on the case **m=2** and **k=p**, a prime.

Some special cases of these groups are $\mathbf{PSL(2,3) \cong A_4}$ (which is not simple!), $\mathbf{PSL(2,4) \cong A_5 \cong PSL(2,5)}$ and $\mathbf{PSL(2,9) \cong A_6}$.

The structure of the projective unimodular groups has been fully investigated by L.E.Dickson in his treatise *Linear Groups* which dates back to 1900.

P. Hall used the structure of these groups as an example of computation of the Eulerian functions of **PSL(2,p)** in his paper [Hal36], upon which this section is fundamentally based. The **d$^{\text{th}}$** Eulerian function of **PSL(2,p)** is given by the following formula:

$$\phi_d(PSL(2,p)) = (2pqr)^d - 2r\,(pq)^d - pq(2r)^d + pr\,[\,2\,q^d - (2q)^d\,] + 2pqr\cdot S$$

where

$$q = \frac{p-1}{2}\,, \qquad r = \frac{p+1}{2}$$

and S depends on the form of the prime **p**. For **p>11** we have

(i) if $p \equiv \pm 1 \pmod 5$ and $\pm 1 \pmod 8$

$$S = -\frac{60^d}{30} - \frac{24^d}{12} + \frac{12^d}{6} + \frac{10^d}{5} + \frac{8^d}{4} + \frac{2\cdot 6^d}{3} - 2\cdot 3^{d-1} - 5\cdot 2^{d-1} + 2$$

(ii) if $p \equiv \pm 1 \pmod 5$ and $\pm 3 \pmod 8$

$$S = -\frac{60^d}{30} + 12^{d-1} + \frac{10^d}{5} + \frac{6^d}{3} + 4^{d-1} - 3^{d-1} - 3\cdot 2^{d-1} + 1$$

(iii) if $p \equiv \pm 2 \pmod 5$ and $\pm 1 \pmod 8$

$$S = -\frac{24^d}{12} + \frac{8^d}{4} + \frac{6^d}{3} - 2^{d-1}$$

(iv) if $p \equiv \pm 2 \pmod 5$ and $\pm 3 \pmod 8$

$$S = -\,12^{d-1} + 4^{d-1} + 3^{d-1} + 2^{d-1} - 1$$

Example 3.9.1 We computed the closed formula for the expected number of elements of **PSL(2,p)** which have to be drawn at random before a set of generator is found, when **p** belongs to the case (iv). It is:

$$e(PSL(2,p)) = 2 + \frac{1}{p} + \frac{12}{13\cdot(-3+p)} + \frac{4}{33\cdot(-2+p)} + \frac{2}{3\cdot(-1+p)} - \frac{2}{3\cdot(1+p)} - \frac{2}{3\cdot(2+p)} - \frac{1}{-1+p+p^2} + \frac{\frac{24}{11} + \frac{6\cdot p}{11}}{3+2\cdot p+p^2} + \frac{-\frac{72}{13} - \frac{12\cdot p}{13}}{8+3\cdot p+p^2} - \frac{8}{-8-p+p^3}$$

$$- \frac{4}{-4 - p + p^3} \qquad + \frac{2}{-2 - p + p^3}$$

This formula shows that, if $p \equiv \pm 2 \pmod 5$ and $p \equiv \pm 3 \pmod 8$

$$e(PSL(2,p)) = 2 + O(\tfrac{1}{p})$$

The same asymptotic formula for e(PSL(2,p)) holds when **p** belongs to one of the other classes (i), (ii) or (iii).[10]

3.10. On the direct product of isomorphic simple groups.

Let us define $\mathbf{d_n(G)}$ as the *greatest number **d** for which the direct product of **d** groups isomorphic to G can be generated by **n** elements.*

In [Hal36] it is proved that, if G is a simple group of composite order, then

$$\mathbf{d_n(G)} = \frac{\boldsymbol{\phi_n(G)}}{\mathbf{a(G)}}$$

where **a**(G) is *the order of the automorphism group of G.*

Example 3.10.1 For all the groups $\mathbf{A_n}$, n≠2 and n≠6, we have $\mathbf{a(A_n) = n!}$. For A_6 we have $\mathbf{a}(A_6)=1440$. In general, for the groups **PSL(2,p)**, **p** prime, considered above, **a**(G) is twice the order of the group. For A_5 which is isomorphic to **PSL(2,5)** we have $\mathbf{d_2(A_5)} = \frac{\phi_2(A_5)}{a(A_5)} = \mathbf{19}$; this means that A_5^{19} can be generated by two elements, but two elements are not enough to generate A_5^{20}.

[10] This assertion was proved using the computer program *Mathematica*. See [Wol88].

The formula for $\mathbf{d_n}(G)$ given above can also be applied to a characteristically simple group[11] to decide if **n** elements are enough to generate it.

[11] A finite non trivial group without characteiistic subgroups is said to be characteristically simple: such a group is either simple or direct product of isomorphic simple groups [Rot88, theorem 5.20].

4. Random walk on a Cayley graph.

4.1. Introduction.

In this chapter we are going to discuss the problem of generating random elements in a group, with uniform distribution, when a set of generators for the group is known.

To introduce the subject we relate it to another problem, viz. the generation of all the elements in a finite group.

Let G be a group generated by a subset $H = \{\mathbf{h}_1,\mathbf{h}_2,\ldots,\mathbf{h}_m\}$ of its elements. Each element **g** of G can be expressed as a word in the generators $\mathbf{h}_i$, i.e. as a product of the form $\mathbf{h}_{i_1}\cdot\mathbf{h}_{i_2}\cdot\ldots\cdot\mathbf{h}_{i_k}$, with **k** being called the length of the word. This suggests an elementary method to generate all the elements in G: form all the words of length **1**, then all the words of length **2**, and so on... The problem with this approach is that usually an element $g\in G$ has more than one representation as a word in the elements in **H**: this happens because of the relations holding among the generators. Thus each time we form a new word we have to test that the corresponding element has not been produced already. The following example, taken from [Atk90] points out this problem.

Example 4.1.1 In the dihedral group generated by the two permutations:

$$a:\begin{pmatrix}1 & 2 & 3 & \ldots & n\\ n & n-1 & n-2 & \ldots & 1\end{pmatrix} \qquad b:\begin{pmatrix}1 & 2 & 3 & \ldots & n\\ n-1 & n-2 & n-3 & \ldots & n\end{pmatrix}$$

the element **ababab...** of length **n** has no representation as a word of length less than **n**. With the method given above this element will not be discovered until about 2^n words have been formed.

Even if we knew in advance the order of the group, that is when to stop forming new words, it is clear that this method for producing all the elements of a group is not effective. For the same reason, a naive strategy of production of random elements in a group by forming random words in the generators is not generally successful.

A better method for generating all the elements of a group G is based on the following idea : let $G = G^0 > G^1 > G^2 > ... > G^m = \{1\}$ be a descending chain of subgroups of G ending in the identity subgroup and let U_k be a set of representatives for the cosets of G^k in G^{k-1}, for $k=0,1,...,m-1$. It can be proved by induction that every element $g \in G$ can be expressed uniquely as a product of the form $u_{m-1} \cdot u_{m-2} \cdot ... \cdot u_1 \cdot u_0$ where each u_i is in U_i. To produce all the elements of G form all the products of the form $u_{m-1} \cdot u_{m-2} \cdot ... \cdot u_1 \cdot u_0$. To produce a random element of G choose each u_i at random from U_i, and then compute the product $u_{m-1} \cdot u_{m-2} \cdot ... \cdot u_1 \cdot u_0$.

A practical application of these ideas was first proposed by C. Sims in [Sim70] and is known as the *stabiliser chain representation* of a permutation group: let G be a permutation group acting on a set $\Omega = \{1,2,...,n\}$ and take for G^k the stabiliser of the points $1,2,...,k$ in G. Great effort was done to reduce the space required to store the chain of stabilisers and to compute the representation in acceptable time: see [Sim71] and [Sim75]. Further development of Sim's methods by M. Jerrum produced an algorithm, described in [Jer86], which computes the chain of stabilisers in $O(n^5)$ time using $O(n^2)$ space.

In [Leo80] J. Leon described a *probabilistic algorithm* for computing a chain of stabilisers which is much faster than either Sim's or Jerrum's algorithm and is based on the assumption that we are given an *external process* which generates random elements of G. Being probabilistic Leon's algorithm can give the wrong output without any warning: but if the external process is known to generate the random elements with true uniform distribution, then we could at least give an upper bound on the probability of error. In Leon's description of the algorithm the random elements are produced by simply forming random words in the original generators; we have already observed that this method is not uniform.

In the present chapter we will relate the length of the random words to the distribution of the random elements.

4.2. Cayley graphs.

The Cayley graph $\mathcal{G}$ associated to a group $\mathbf{G} = \{\mathbf{g_1,g_2,\ldots,g_n}\}$ generated by a subset $\mathbf{H} = \{\mathbf{h_1,h_2,\ldots,h_m}\}$ of its elements is a directed graph with vertex set G and edge set $\mathbf{L=\{(g_i,\ g_i{\cdot}h_k) \mid i=1,2\ldots,n\ ;\ k=1,2,\ldots,m\}}$, the edge $\mathbf{(g_i,\ g_i{\cdot}h_k)}$ being labelled $\mathbf{h_k}$. For simplicity of notation we denote by **I** the vertex representing the identity, and we write "the Cayley graph $\mathcal{G}$" for "the Cayley graph $\mathcal{G}$ associated to a group $\mathbf{G} = \{\mathbf{g_1,g_2,\ldots,g_n}\}$ generated by a subset $\mathbf{H} = \{\mathbf{h_1,h_2,\ldots,h_m}\}$ of its elements". In addition, when a generator is involutory we will represent its effect by a single undirected arc.

The Cayley graph $\mathcal{G}$ associated to a group G can be considered as a special case of the graphs considered in section 1.2, if we take $\Omega = \mathbf{G}$ and we define $\mathbf{g^h}$ to be $\mathbf{g{\cdot}h}$, for all $\mathbf{g}\in \mathbf{G}$ and $\mathbf{h}\in \mathbf{H}$.

It is easy to see from the definition that in a Cayley graph the indegree of each vertex is equal to the outdegree which in turn is equal to the number of generators of the group.

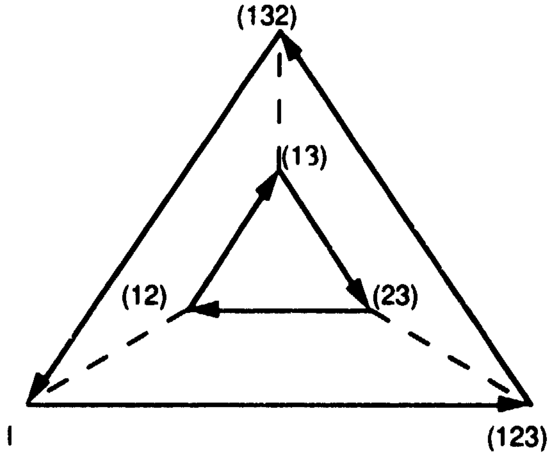
The property of a Cayley graph $\mathcal{G}$ which is most important for us is that *a product of generators corresponds uniquely to a walk in $\mathcal{G}$ starting from **I**.*

Many properties of the group G can be derived by examining $\mathcal{G}$ - we list here some of them whose proof is obvious:

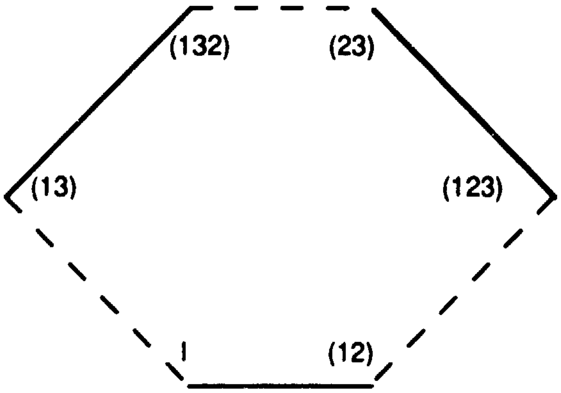
(i) G is *abelian* if all the walks of length two starting from **I** whose arcs are labelled $\mathbf{h_i,h_j}$ and $\mathbf{h_j,h_i}$ lead to the same point, $\forall\ \mathbf{i,j} \in \{1,2,\ldots,m\}$

(ii) a generator **h** is *redundant*[12] if every vertex of **G** is reachable from **I** by a walk which does not include any arc labelled **h**. In other words **h** is redundant if and only if the digraph obtained from **G** by deleting all the arcs labelled **h** is strongly connected.

It should be clear that the Cayley graph **G** of a group G depends on the generating set **H**. As an example, the following two graphs *which are not isomorphic* represent the same group, S_3, being generated by different sets :



Here the group S_3 is generated by the elements (12) and (123). The dashed undirected lines correspond to the involutory generator (12), the directed ones to the generator (123).



Here the group S_3 is generated by the elements (12) and (13). The dashed undirected lines correspond to the involutory generator (13), the other ones to the generator (12).

[12] A generator is redundant if it can be written as the product of other generators.

4.3. Properties of the adjacency matrix of a Cayley graph.

Theorem 4.3.1 If A is the adjacency matrix associated to a graph $\mathcal{G}$ then $(A^t)_{ij}$ is the number of paths of length **t** from the vertex **i** to the vertex **j**.

Proof. See [Chr75, section 1.8.1].

Theorem 4.3.2 If A is the adjacency matrix of a Cayley graph $\mathcal{G}$, then the sum of the elements in any row is equal to the sum of the elements in any column and it is equal to the size of the generating set of the group **G**.

Proof. This follows from the fact that in a Cayley graph the indegree of each vertex is equal to the outdegree which is equal to the number of generators of the group considered.

Corollary 4.3.3 Let A be a square $\mathbf{n} \times \mathbf{n}$ matrix with row and column sums equal to **d**, and entries in $\{0,1\}$. Then $(1,1,\ldots,1)$ is an eigenvector of the matrix with eigenvalue **d**.

Theorem 4.3.4 If A is the adjacency matrix of a Cayley graph $\mathcal{G}$, then $(A^t)_{ii}$, the number of closed paths of length **t** from the vertex **i** to itself, is independent of **i**.

Proof. A closed path of length **t** in a Cayley graph simply represents a product of **t** group elements chosen among the generators which is equal to the identity.

Corollary 4.3.5 $\sum_{i=1}^{n} (A^t)_{ii} = \mathrm{trace}(A^t) = \sum_{i=1}^{n} \lambda_i^t = 0 \pmod{\mathbf{n}}$ for all **t**, where **n** is the degree of A, and λ_i is the i^{th} eigenvalue of A.

Theorem 4.3.6 $\|vA\| \le \mathbf{d}\,\|v\|$ for all vectors v, where $\|\cdot\|$ denotes the euclidean norm.

Proof. If $v = (v_1, v_2, \ldots, v_n)$ then

$$\|vA\|^2 = \sum_{i=1}^{n} \Big(\sum_{j=1}^{n} v_j \; A_{ji} \Big)^2 \qquad = \sum_{i=1}^{n} \Big(\sum_{j=1}^{n} v_j \; A_{ji} \, A_{ji} \Big)^2$$

$$\leq \sum_{i=1}^{n} \sum_{j=1}^{n} v_j^2 \; A_{ji}^2 \sum_{k=1}^{n} A_{ki}^2$$

{ by the Cauchy inequality applied to each term of the outer summation }

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} v_j^2 \quad A_{ji} \sum_{k=1}^{n} A_{ki} \qquad = \sum_{j=1}^{n} v_j^2 \sum_{i=1}^{n} \; A_{ji} \sum_{k=1}^{n} A_{ki}$$

$$= d^2 \, \|v\|^2$$

Theorem 4.3.7 Every eigenvalue λ of $\mathbf{A}$ satisfies $|\lambda| \leq \mathbf{d}$.

Proof. Let v be the eigenvector for λ. Then $\mathbf{vA} = \lambda\mathbf{v}$ and $|\lambda| \cdot \|v\| = \|vA\| \leq \mathbf{d} \, \|v\|$, i.e. $|\lambda| \leq \mathbf{d}$.

Theorem 4.3.8 If **A** is the matrix of a directed graph which has the properties:

(i) there exists a vertex **1** from which every vertex can be reached

(ii) there exists an odd cycle

then $\mathbf{vA} = \lambda\mathbf{v}$ implies that **v** is a multiple of **n**.

Proof. In the proof of Theorem 4.3.6 equality implies equality in each application of the Cauchy inequality. But $\left(\sum_{i=1}^{n} x_i \; y_i \right)^2 = \sum_{i=1}^{n} x_i^2 \cdot \sum_{i=1}^{n} y_i^2$ if and only if **x** and **y** are linearly dependent (i.e. they are parallel). Hence there exist numbers $\mathbf{w_i}$ such that $A_{ji} = w_i \cdot v_j \cdot A_{ji}$. Thus, if there is an edge from **j** to **i** (that is, $A_{ji} = 1$) then $w_i = v_j^{-1}$. It follows that, if there is an even length path from **1** to **i** then $v_i = v_1$. But since an odd cycle exists we can replace paths of odd length by paths of even length and so $v_i = v_1$ for all **i**.

Note 4.3.9 If the Cayley graph has no odd cycle, then it is bipartite. But this implies that **G** has a subgroup **K** of index .wo and all the generators lie in **G-K**. In this case there is no hope of convergence.

Note 4.3.10 The theorems shown above still hold when we normalize the adjacency matrix of the Cayley graph, i.e. when we consider the doubly stochastic matrix obtained by dividing each entry by **d**, according to the next theorem:

Theorem 4.3.11 If λ is an eigenvalue of A and **d** is a real number, then $\frac{\lambda}{d}$ is an eigenvalue of $\frac{1}{d}$ A.

Proof. If v is an eigenvector corresponding to the eigenvalue λ, then $\lambda v = Av$, which implies that $(\frac{1}{d}\lambda) v = (\frac{1}{d} A) v$.

4.4. On the rate of convergence to uniform.

We have seen that a product of generators in a group **G** corresponds uniquely to a walk in the associated Cayley graph $\mathfrak{G}$ starting from the point **I** representing the identity. Thus a product of **m** randomly chosen generators correspond to a random walk of length **m** on the Cayley graph. The question that we would like to answer here is the following: how big should be **m** in order to achieve a uniform convergence, that is in order to visit each vertex with equal (or close to equal) probability?

The analytic tool used to analyse random walks on graphs is the theory of *Markov chains*. Let us briefly review some facts concerning finite Markov chains.

Let us suppose that a system can be at a given time $t = 0,1,2,3,..$ in one of **n** possible states. Let us define p_{ij} as the probability that at any time the system moves from state **i** to state **j**. The matrix $P = (p_{ij})$ is called the *transition matrix* of the Markov chain. The probability that the system will move from state **i** to state **j** in **k** time steps is given by

$(P^k)_{ij}$. Let us also define the initial probability vector π_0 of a Markov chain as the vector whose i^{th} entry represents the probability that the system initially is in the state **i**. In the same way we define the k^{th} step probability vector π^k as the vector whose i^{th} entry represents the probability that the system will be in the state **i** at time **k**. The following relation holds:

$$\pi^k = P^k \pi_0$$

For our purposes we can suppose that $\pi_0 = (1,0,0,\ldots,0)$. Let $e_1, e_2, \ldots, e_n$ be the **n** distinct eigenvectors of the matrix **P**, to which correspond the eigenvalues λ_1, $\lambda_2, \ldots, \lambda_n$ where we assume that $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$. If the eigenvectors were linearly independent, we could express π_0 as $\sum_{i=1}^{n} a_i \cdot e_i$ where the a_i are opportunely chosen coefficients. We obtain now

$$\begin{aligned} \pi^k &= P^k \pi_0 = \sum_{i=1}^{n} a_i \lambda_i^k e_i = \lambda_1^k \left(\sum_{i=1}^{n} a_i \left(\frac{\lambda_i}{\lambda_1} \right)^k e_i \right) \\ &= \lambda_1^k \left(e_1 a_1 + \sum_{i=2}^{n} a_i \left(\frac{\lambda_i}{\lambda_1} \right)^k e_i \right) \end{aligned}$$

The last expression behaves like $\lambda_1^k \cdot e_1 \cdot a_1$ when **k** goes to infinity, that is

$$\lim_{k \to \infty} \pi^k = (a_1, a_1, \ldots, a_1)$$

since as we have seen before, for a doubly stochastic matrix we have $\lambda_1 = 1$ and $e_1 = (1,1,\ldots,1)$.

The important thing to note here is the fact that the rate of convergence depends on the second largest (in modulus) eigenvalue, that is on λ_2.

4.5. Automorphisms of a graph.

An automorphism σ of a graph $\mathcal{G}$ is a permutation of its vertices which preserves adjacency: if $(\mathbf{a},\mathbf{b})$ is an edge of $\mathcal{G}$ then $(\mathbf{a}^{\sigma},\mathbf{b}^{\sigma})$ is also an edge of $\mathcal{G}$. We can associate to an automorphism σ a matrix S defined in the following way:

$S_{ij} = \mathbf{1}$ if $\mathbf{i}^{\sigma} = \mathbf{j}$ for two vertices $\mathbf{i}$ and $\mathbf{j}$

$S_{ij} = \mathbf{0}$ otherwise

The set of all the automorphisms of a graph $\mathcal{G}$ forms a group under the operation of composition.

Theorem 4.5.1 If $\mathcal{G}$ is a graph with adjacency matrix A and σ is an automorphism of $\mathcal{G}$ with matrix S, then $S^T = S^{-1}$.

Proof. $$(SS^T)_{ij} = \sum_{k=1}^{n} s_{ik} \cdot s_{jk}$$

A summand on the right hand side is nonzero if and only if $\mathbf{i}^{\sigma} = \mathbf{k}$ and $\mathbf{j}^{\sigma} = \mathbf{k}$, but this can happen if and only if $\mathbf{i}$ is equal to $\mathbf{j}$ because σ is a bijection. It follows that $(SS^T)_{ij} = \delta_{ij}$, where δ_{ij} is the Kronecker symbol, which is equal to one if $\mathbf{i}$ is equal to $\mathbf{j}$, zero otherwise.

Theorem 4.5.2 If $\mathcal{G}$ is a graph with adjacency matrix A and σ is an automorphism of $\mathcal{G}$ with matrix S, then $SAS^T = A$.

Proof. $$(SAS^T)_{ij} = \sum_{p,q=1}^{n} s_{ip} \cdot a_{pq} \cdot s_{jq}$$

A term in the summation on the right hand side is equal to one if and only if $(\mathbf{p},\mathbf{q})$ is an edge of G, $\mathbf{i}^{\sigma} = \mathbf{p}$ and $\mathbf{j}^{\sigma} = \mathbf{q}$. But $\mathbf{p}$ and $\mathbf{q}$ are unique if they exist and they exist precisely if $\mathbf{i}$ is joined to $\mathbf{j}$, and this proves that $(SAS^T)_{ij} = a_{ij}$

Corollary 4.5.3 If $\mathcal{G}$ is a graph with adjacency matrix A and σ is an automorphism of $\mathcal{G}$ with matrix S, then $SAS^{-1} = A$, i.e. the adjacency matrix of a graph $\mathcal{G}$ commutes with all the matrices associated to the automorphisms of $\mathcal{G}$.

Theorem 4.5.4 The adjacency matrix $\mathbf{A}$ of a Cayley graph $\mathfrak{G}$ commutes with $\mathbf{L(g)}$, the left regular representation of $\mathbf{G}$, for all $\mathbf{g} \in \mathbf{G}$.

Proof. For an arbitrary Cayley graph $\mathfrak{G}$ the left multiplication by an element $\mathbf{g} \in \mathbf{G}$ is an automorphism of $\mathfrak{G}$. In fact, if $(\mathbf{a},\mathbf{b})$ is an edge of $\mathfrak{G}$ labelled $\mathbf{x}$, this means that $\mathbf{ax} = \mathbf{b}$; but then $\mathbf{gax} = \mathbf{gb}$, that is there is an edge labelled $\mathbf{x}$ which connects $\mathbf{ga}$ to $\mathbf{gb}$. Therefore by Corollary 4.5.3 the adjacency matrix of a Cayley graph commutes with all the matrices associated to the automorphisms of $\mathfrak{G}$ induced by left multiplication by any element $\mathbf{g} \in \mathbf{G}$.

4.6. Review of fundamental concepts in representation theory.

To obtain deeper results about the eigenvalues of the adjacency matrix of a Cayley graph, we use some basic facts from the representation theory of groups - in particular, the representation theory explains the large multiplicity which usually characterizes these eigenvalues. We recall some fundamental concepts, drawing freely from [Led77].

If $\mathbf{G}$ is a group, a *matrix representation A of degree* $\boldsymbol{m}$ *over a field K* is an homomorphism $\mathbf{A}:\mathbf{G} \to \mathbf{GL_m(K)}$, the group of all non singular matrices of degree $\mathbf{m}$ over $\mathbf{K}$.

We say that two representations *A and B are equivalent over K*, and we write $\mathbf{A} \sim \mathbf{B}$, if there is a non singular matrix $\mathbf{T}$ with coefficients in $\mathbf{K}$ such that $\mathbf{B(x)} = \mathbf{T^{-1}A(x)T}$ for all $\mathbf{x}$ in $\mathbf{G}$.

A matrix representation $\mathbf{A}$ is said to be *reducible over K* if there is a non singular matrix $\mathbf{T}$ with coefficients in $\mathbf{K}$ such that

$$\mathbf{B(x)} = \mathbf{T^{-1}A(x)T} = \begin{pmatrix} \mathbf{C(x)} & \mathbf{0} \\ \mathbf{E(x)} & \mathbf{D(x)} \end{pmatrix}$$

for all $\mathbf{x}$ in $\mathbf{G}$, irreducible otherwise.

Every matrix representation can be put in lower triangular form, in which the diagonal blocks are irreducible.

If $\mathbf{M} = \begin{pmatrix} \mathbf{M}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 & \mathbf{0} \;.. & \mathbf{0} \\ .. & .. & \dots & .. \\ \mathbf{0} & .. & ..\;\mathbf{0} & \mathbf{M}_r \end{pmatrix}$ we say that the matrix **M** is the *direct sum* of the matrices $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_r$ and we write $\mathbf{M} = \sum_{i=1}^{r} \oplus \; \mathbf{M}_i$.

A matrix represcntation **A** is said to be *completely reducible over* ***K*** if $\mathbf{A} \sim \sum_{i=1}^{r} \oplus \; \mathbf{A}_i$ and each $\mathbf{A}_i$ is irreducible over **K**.

If **G** is a finite group of order **g** and **K** is a field whose characteristic is zero or prime to **g**, then every matrix representation of **G** over **K** is completely reducible over **K** (*Maschke's theorem*).

A matrix representation **A** of a group **G** over an algebraically closed field **K** is irreducible over **K** if and only if the only matrices that commute with all the matrices **A(x)** (for all $\mathbf{x} \in \mathbf{G}$) are the scalar multiples of the unit matrix (*Schur's lemma*).

Given two square matrices $\mathbf{A} = (\mathbf{a}_{ij})$ and **B** of degrees **m** and **n** respectively, we define their *tensor product* or *Kronecker product* $\mathbf{A} \otimes \mathbf{B}$ to be the $\mathbf{mn} \times \mathbf{mn}$ matrix:

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} \mathbf{a}_{11}\mathbf{B} & \mathbf{a}_{12}\mathbf{B} & .. & \mathbf{a}_{mm}\mathbf{B} \\ \mathbf{a}_{21}\mathbf{B} & \mathbf{a}_{22}\mathbf{B} & .. & \mathbf{a}_{2m}\mathbf{B} \\ .. & .. & .. & .. \\ \mathbf{a}_{m1}\mathbf{B} & \mathbf{a}_{m2}\mathbf{B} & .. & \mathbf{a}_{mm}\mathbf{B} \end{pmatrix} = (\mathbf{a}_{ij}\,\mathbf{B})$$

The set $\mathbb{C}(\mathbf{A})$ of all matrices **T** over **K** commuting with **A(x)** (for all $\mathbf{x} \in \mathbf{G}$) is an algebra over **K**, that is a vector space endowed with an associative multiplication: $\mathbb{C}(\mathbf{A})$ is called the *commutant algebra* of A, and it is a subalgebra of the matrix algebra $\mathcal{M}_n(\mathbf{K})$.

If $\mathbf{A} = \sum_{i=1}^{r} \oplus \; (\mathbf{I}_{e_i} \otimes \mathbf{F}_i)$ where $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_r$ are inequivalent irreducible representations of degrees $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_r$ which occur with multiplicities $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_r$ in the diagonal form of A, then the algebra $\mathbf{C}(A)$ has dimension $e_1^2+e_2^2+\ldots+e_r^2$, and r represents the dimension of its centre.

A typical element of $\mathbf{C}(A)$ is a matrix of the form

$$T = \sum_{i=1}^{r} \oplus \, (X_i \otimes I_{f_i})$$

where X_i is an arbitrary $e_i \times e_i$ matrix.

4.7. Regular representations.

Let us consider now the elements of G as the basis of a vector space $G_K = [g_1, g_2, \ldots, g_n]$ over a field K, endowed with an associative multiplication. G_K is an algebra, called the *group algebra* of G over K, its elements being linear combinations of the g_i's with coefficients in K. The elements of G can be regarded as linear combinations of elements of the basis in which all the coefficients are equal to zero, except one (which is equal to one).

We can define the *right regular representation* of G as the group of linear mappings induced by multiplication on the right of elements of G_K by elements of G.

The mapping corresponding to the multiplication on the right by a particular g in G transforms each element g_i of the basis into $g_i \cdot g$ and therefore can be regarded as a permutation of the element of the basis. Such a permutation can be specified by a permutation matrix R(g) of degree n, where $R(g)_{i,j}$ is equal to one if $g_i \cdot g = g_j$, zero otherwise.

The set of matrices $R = \{R(g) \mid g \in G\}$ forms a group, the operation being matrix multiplication. This group is isomorphic to the group of linear mappings defined above, and we will call R the *right regular representation* of G, without creating confusion.

In the same way we can define the *left regular representation* of G, as the group L of all permutation matrices L(g) of degree n, where $L(g)_{j,i}$ is equal to one if $g \cdot g_i = g_j$, zero otherwise.

Theorem 4.7.1 If a group G has c conjugacy classes then there are c distinct irreducible representations over the complex field, and they are all present in the left (resp. right) regular representation of G, each one with multiplicity equal to its degree.

Proof. See [Led77,pag.49]

Thus, in the left regular representation L(x) we have $|G| = \sum_{i=1}^{c} f_i \cdot e_i = \sum_{i=1}^{c} f_i^2$ because the relation $e_i = f_i$ holds for all i = 1,2,...,c.

A typical element T of the commutant algebra of the left regular representation L has the form

$$T = \sum_{i=1}^{c} \oplus (X_i \otimes I_{e_i})$$

where X_i is an arbitrary $e_i \times e_i$ matrix.

Theorem 4.7.2 If X is a square matrix of degree m, the matrix $X \otimes I_n$ has the same eigenvalues of the matrix X, each one repeated n times.

Proof. If $v = \begin{pmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_m \end{pmatrix}$ is an eigenvector of **X**, corresponding to an eigenvalue λ,

and $\begin{pmatrix} 1 \\ 1 \\ .. \\ 1 \end{pmatrix}$ is a vector of **n** ones, consider $\mathbf{w} = v \otimes \begin{pmatrix} 1 \\ 1 \\ .. \\ 1 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_1 \\ \cdot \\ v_1 \\ v_2 \\ v_2 \\ \cdot \\ v_2 \\ \cdot \\ \cdot \\ v_m \\ v_m \\ \cdot \\ v_m \end{pmatrix}$

it is easy to see that **w** is an eigenvector of $\mathbf{X} \otimes \mathbf{I_n}$ to which correspond **n** copies of the eigenvalue λ.

As we said at the beginning of section 4.6 the representation theory of groups is the tool that allows one to explain the multiplicity of the eigenvalues of the adjacency matrix of a Cayley graph; the next corollary makes this assertion more clear.

Corollary 4.7.3 If **T** is an element of the commutant algebra of **L** of the form

$$\mathbf{T} = \sum_{i=1}^{c} \oplus \ (\mathbf{X}_i \otimes \mathbf{I}_{e_i})$$

where $\mathbf{X_i}$ is an arbitrary $e_i \times e_i$ matrix, then if k_i is the multiplicity of the eigenvalue λ_i of **T**, which has multiplicity z_{ij} in the matrix $\mathbf{X_j}$, the following conditions hold:

(i) $\sum_{i=1}^{c} e_i^2 = |G|$

(ii) $k_i = z_{i1} \cdot e_1 + z_{i2} \cdot e_2 + \ldots + z_{ic} \cdot e_c$

(iii) $\sum_{k} z_{ki}' = e_i$

Proof. The validity of the first condition has been shown before. The second condition simply states that the eigenvalue λ_i was present in the matrices $X_1, X_2, \dots, X_c$ with multiplicity resp. z_{i1}, z_{i2}, ..., z_{ic}. The third condition states that the sum of the multiplicities of the distinct eigenvalues in X_i is equal to the degree of X_i, which is equal to e_i because we are dealing with regular representations.

4.8. Some final considerations.

If we are given the *character table* of a group G, it is easy to obtain the degrees of all its irreducible representations: in fact, they correspond to the values that each simple character takes in the identity:

$$e_i = \chi^i(1)$$

This greatly simplifies our work.

In the case of a *finite abelian* group, the number c of conjugacy classes is equal to the order of the group. It follows that each $e_i = f_i = 1$, i.e. all the irreducible representations have degree one. Therefore nothing can be said in this case, by applying the methods discussed above, about the origin of the eigenvalues of the adjacency matrix and their multiplicities.

When we form the random products of generators, if we add to each element in the generating set **H** its inverse, the sum of the elements of each column in the adjacency matrix of the Cayley graph becomes:

$$d = 2\cdot\#(\text{non involutory generators}) + \#(\text{involutory generators})$$

In this case the adjacency matrix A of the Cayley graph is symmetric, and therefore all its eigenvalues λ_i'' lie in the real field. The doubly stochastic matrix A'' obtained from A by dividing each entry by **d** has therefore all its eigenvalues λ_i'' in the real field, and each λ_i'' is equal to $\frac{\lambda_i}{d}$, where λ_i ($i = 1,2,\ldots,|G|$) are the eigenvalues of A.

If we add the identity to the set of generators then we create cycles of odd length in the Cayley graph, which may have the effect of transforming a non convergent Markov chain into a convergent one.

4.9. Methodology to build an example.

Given a finite group G finitely presented an algorithm known as "coset enumeration", first described in [Tod36], allows us to obtain among other information the regular representation of G, which shows the effect of multiplying each element of G by its generators. To build the example discussed in the next section we followed this procedure:

step 1 Obtain a presentation of G.

step 2 Run the coset enumeration algorithm to produce the regular representation of G.

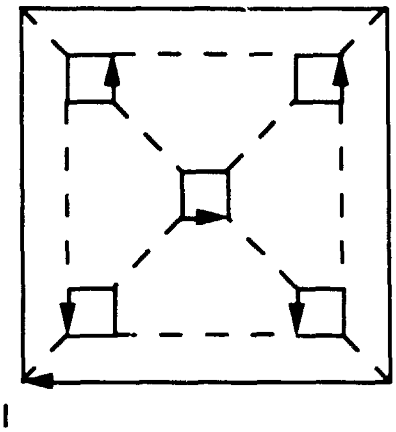
step 3 Build the doubly stochastic matrix A corresponding to the original presentation, plus for each generator its inverse.

step 4 Obtain the eigenvalues, and try to determine which irreducible representation they are associated with.

4.10. A small example : S_4

Presentation:

$$< a, b \mid a^2 = b^4 = (ab)^3 = 1 >$$



Cayley graph:

The dashed undirected lines represent the effect of the involutory generator, the directed ones the effect of the generator of order four.

Eigenvalues rounded to the fifth decimal place:

eigenvalue	-2	-1.56155	-1	0	1	2	2.56155	3
multiplicity	5	3	3	5	1	3	3	1

Degrees of the irreducible representations
derived from the character table :

e_1	e_2	e_3	e_4	e_5
1	1	2	3	3

Number of conjugacy classes:

value of c
5

Origin of the eigenvalues.

The eigenvalue **1**, hav g multiplicity one, must come from the block **[1]** which is of the form $\mathbf{X_1 \otimes I_{e_1}}$.

The same argument applies to the eigenvalue **3**, which must come from the block **[3]** which is of the form $\mathbf{X_2 \otimes I_{e_2}}$.

The eigenvalues **-1.56155** and **2.56155** are the solutions of the equation $\mathbf{x^2 + x - 4 = 0}$, which is irreducible in the integers, thus they must appear always in the same block. Now, all the eigenvalues of multiplicity three must come from the same block: it follows that the eigenvalues **-1.56155** and **2.56155** come from the block of degree nine, which can be $\mathbf{X_4 \otimes I_{e_4}}$ or $\mathbf{X_5 \otimes I_{e_5}}$.

In the table that follows we show the possible[13] dispositions of the eigenvalues in the blocks of the adjacency matrix: each configuration corresponds to a row in the tabl.

To be more concise, we consider the permutations of the blocks of the same size as belonging to the same configuration.

[13] "possible" stands for "compatible with the conditions stated in the present chapter".

The notation λx**k** inside a cell stands for "an eigenvalue λ with multiplicity **k** is associated to this block".

block of deg 1	block of deg 1	block of degree 4	block of degree 9	block of degree 9
1 x 1	3 x 1	0 x 2 -2 x 2	-1.56155 x 3 2.56155 x 3 2 x 3	0 x 3 -2 x 3 -1 x 3
1 x 1	3 x 1	0 x 2 -2 x 2	-1.56155 x 3 2.56155 x 3 0 x 3	-2 x 3 2 x 3 -1 x 3
1 x 1	3 x 1	0 x 2 -2 x 2	-1.56155 x 3 2.56155 x 3 -2 x 3	0 x 3 2 x 3 -1 x 3
1 x 1	3 x 1	0 x 2 -2 x 2	-1.56155 x 3 2.56155 x 3 -1 x 3	0 x 3 -2 x 3 2 x 3

References.

[Acc90] Acciaro, V. and Atkinson, M.D., *A new algorithm for testing the regularity of a permutation group*, School of Computer Science Technical Report SCS-TR-183, Carleton University, Ottawa, November 1990

[Aho74] Aho, A.V, Hopcroft, J.E. and Ullman, J.D., *The design and analysis of computer algorithms*, Addison-Wesley, 1974

[Atk75] Atkinson, M.D., *An algorithm for finding the blocks of a permutation group*, Math.Comp.29 (1975), 911-913

[Atk90] Atkinson, M.D., *A survey of algorithms for handling permutation groups*, School of Computer Science Technical Report SCS-TR-164, Carleton University, Ottawa, January 1990

[Big79] Biggs, N.L. and White, A.T., *Permutation groups and combinatorial structures*, Cambridge University Press, 1979

[Chr75] Christofides, N., *Graph theory, an algorithmic approach*, Academic Press, 1975

[Dia88] Diaconis, P., *Group representations in probability and statistics*, 1988

[Dic00] Dickson, L.E., *Linear groups*, Leipzig, 1900

[Dix67] Dixon, J.D., *Problems in group theory*, Blaisdell Publishing Company, 1967

[Dix71] Dixon, J.D., *Permutation representations and the subgroup lattice*, Proc.2nd Symposium on Symbolic and Algebraic Manipulation, 23-28, ACM, New York, 1971

[Hal36] Hall, P., *The Eulerian functions of a group*, Quart. J. Math., Ox. Series 7 (1936) 134-151

[Jer86] Jerrum, M., *A compact representation of permutation groups*, J.Algorithms 7 (1986), 60-78

[Jon90] Johnson, D.L., *Presentations of groups*, London Mathematical Society Student Texts 15, Cambridge University Press, 1990

[Kan90] Kantor, W.M. and Lubotzky, A., *The probability of generating a finite classical group*, Geometriae Dedicata 36 (1990), 67-87

[Knu73] Knuth, D.E., *The art of computer programming*, Addison Wesley Publishing Company, Inc., 1973

[Led77] Ledermann, W., *Introduction to group characters*, Cambridge University Press, 1987

[Leo80] Leon, J.S., *On an algorithm for finding a base and a strong generating set for a group given by a generating permutations*, Math.Comp.35 (1980), 941-974

[Mac74] Machi', A., *Introduzione alla teoria dei gruppi*, Giangiacomo Feltrinelli Editore, Milano, 1974

[Rot88] Rotman, J.J., *An introduction to the theory of groups*, Wm. C. Brown Publishers, 1988

[Sim67] Sims, C.C., *Graphs and finite permutation groups*, Math.Zeitschr.95, (1967), 76-86

[Sim70] Sims, C.C., *Computational methods in the study of permutation groups*, Computational Methods in Abstract Algebra, 169-183, Pergamon Press, Elmsford, N.Y., 1970

[Sim71] Sims, C.C., *Computation with permutation groups*, Proc.2nd Symposium on Symbolic and Algebraic Manipulation, 23-28, ACM, New York, 1971

[Sim75] Sims, C.C., *Some algorithms based on coset enumeration*, Manuscript, Rutgers University, 1975

[Tar75] Tarjan, R.E., *On the efficiency of a good but not linear set merging algorithm*, J.ACM 22, 1975, 215-265

[Tod36] Todd, J.A. and Coxeter, H.S.M., *A practical method for enumerating cosets in a finite abstract group*, Proc.Edinburgh Math.Soc. (2) 5 (1936), 26-34

[Wie64] Wielandt, H., *Finite Permutation Groups*, Academic Press, New York-London, 1964

[Wol88] Wolfram, S., *Mathematica, A System for Doing Mathematics by Computer*, Addison-Wesley, 1988

END

05·06·92

FIN