Constructive Representation Theory and Fast Discrete Signal Transforms*

Markus Püschel

Technical Report Drexel-MCS-1999-1
Dept. of Mathematics and Computer Science
Drexel University
Philadelphia, PA 19104
June 1999

*The present Tech Report is a translation of the authors dissertation "Konstruktive Darstellungstheorie und Algorithmengenerierung" in Computer Science which was written at the Institute of Algorithms and Cognitive Systems, University of Karlsruhe, Germany in 1998.

Abstract

Many fast discrete signal transforms are given as a decomposition of the corresponding matrix into a product of sparse matrices. In this thesis an algorithm is presented which generates such decompositions automatically using methods of representation theory of finite groups.

The procedure has its roots in the thesis of Torsten Minkwitz and consists essentially of two steps. First, the symmetry of the matrix is determined, which is a pair of monomial representations under which the matrix is invariant. Second, the representations are decomposed stepwise, giving rise to factorized decomposition matrices which determine the factorization of the matrix. Intuitively speaking, the symmetry catches redundancy contained in the matrix and the decomposition of the representations turns the redundancy into a factorization of the matrix. The main contribution of this thesis is an algorithm for the decomposition of a large class of monomial representations including the computation of a factorized decomposition matrix. To solve this problem, a constructive approach to representation theory is developed where representations are considered up to equality instead of equivalence. In this sense, refinements of well-known theorems (e.g. Mackey's Subgroup Theorem, Clifford's Theorem) are developed and applied to the special cases of permutation and monomial representations. Formulas are derived allowing the explicit construction of decomposition matrices for monomial representations in many cases.

AREP, a GAP share package for constructive representation theory, has been created in collaboration with Sebastian Egner and used to implement all algorithms contained in this thesis. Using AREP it was possible to generate fast algorithms for many signal transforms including the Fourier transform, trigonometric transforms, Hartley transform, and Haar transform. This reveals a strong relationship between discrete signal transforms and representation theory, opening a new area of research to explore this connection.

Preface

The present work is a translation of my dissertation in Computer Science to English (Unfortunately, german universities do not encourage you to write an english thesis). I decided to make a faithful translation, though this was sometimes hard to keep once confronted with (fortunately few) minor mistakes. In addition, half a year after completing the original, newer results would allow a more comprehensive presentation of the subject. At least I want to mention some awkward points in this work. In Chapter 2 an algorithm for a large class of monomial representations is presented, but which class is it? Unfortunately this cannot be told in a simple sentence. I tried to derive as many methods as possible yielding a somehow dismembered class of decomposable representations. A recently developed additional method, however, now allows the decomposition of any monomial representation of any solvable group. This is also part of the repeatedly mentioned package AREP. Furthermore I never mention how generalized Fourier transforms fit into the picture. The answer is easy: A generalized Fourier transform (of a finite group) can be viewed as a decomposition matrix of a regular representation (in the strict sense of Chapter 2). This matrix then of course has a perm-irred symmetry, where the permutation representation is even regular. On the other side, every such matrix describes a generalized fast Fourier transform. Hence Algorithm 2.2 allows the generation of fast Fourier transforms for every solvable group. Another weak point is the definition of the matrix SOR_n as the sparsest matrix splitting off the one-representation contained in a permutation representation. Unfortunately the result is not unitary anymore. Taking, e.g. the Haar transform (cf. Section 4.6) instead, retains a unitary transform and also takes only $O(n)$ operations.

Finally I want to mention some slight mistakes. In Theorem 1.29 it must read: Then μ is equivalent by a monomial matrix to an outer ..., also the proof is erraneous. Nevertheless, the statement holds. In Theorem 1.60 the matrix $A_1 \otimes A_2$ is a decomposition matrix only up to a permutation, which of course can easily be computed. In the decomposition of the $\mathrm{DCT\text{-}III}_8$ on page 106, a permutation as the last factor is missing. Also, the Haar transform has a perm-irred symmetry.

The quality of the language is less than what you would expect from a thesis originally written in English. The goal was to create a readable translation rather than spending the large time for perfection.

I thank Prof. Jeremy Johnson for (being my first reader and) proofreading of the most important parts.

Markus Püschel, Pittsburgh, March 1999

Constructive Representation Theory and Generation of Algorithms

For attainment of the University Degree of a Doctor of Science
at the Faculty of Computer Science at the
University of Karlsruhe (College of Technology)

presented

Dissertation

of

Markus Püschel

from Augsburg

Day of oral examination:	6. May 1998
First Referee:	Prof. Dr. Thomas Beth
Second Referee:	Prof. Dr. Heinrich-Wolfgang Leopoldt

13. July 1998

Acknowledgements

In the first place I am indebted to Professor Dr. Beth who always supported my work and laid the foundation for this work already man years ago with his idea of connecting signal processing and representation theory. I want to express my cordial gratitude to Professor Dr. Leopoldt for being the second referee. Among my colleagues I want to thank in the first place Dr. Sebastian Egner for the friendly collaboration which was a lot of pleasure to me. Furthermore I am indebted to Martin Rötteler for proofreading and suggestions for improvement as well as Volker Baumgarte, Detlef Zerfowski, and Markus Grassl for helping me with diverse TEX-problems. In addition I want to thank all my colleagues at the Institute of Algorithms and Cognitive Systems for the open, friendly atmosphere which contributed essentially to the success of this work. Finally, I am indebted to Werner Veit, who put the finishing touches to this dissertation.

Contents

Introduction

Efficient algorithms are of central importance in signal processing. The fast Fourier transform alone was the subject of a vast number of publications in the past thirty years. Many fast signal transforms, like the fast Fourier transform, are realized by a decomposition of the corresponding matrix. The object of this dissertation is the automatic generation of such decompositions.

Background and History

The fast Fourier transform (FFT) is among the most important algorithms in computer science. It is equivalent to an efficient evaluation of the discrete Fourier transform (DFT) which mathematically is nothing but the multiplication of a vector of length n with the matrix

$$\mathrm{DFT}_n = [\omega_n^{ij} \mid i, j \in \{0, \ldots, n-1\}],$$

where ω_n denotes a primitive nth root of unity. Applications of the FFT lie in correlation analysis, polynomial interpolation, the efficient computation of convolutions and, above all, in signal processing. In fact, the publication of the FFT of Cooley/Tukey (1965), [19] was a crucial step towards modern signal processing. They showed that, under certain restrictions on the number n, the computation of the DFT requires only $O(n \log n)$ arithmetic operations compared to $O(n^2)$ required by direct computation.

Therefore it is amazing that an historical examination by Heidemann/Johnson/Burrus (1985), [34] revealed that this very algorithm had been found already more than 150 years ago and by none other than Carl Friedrich Gauß. To this Huang 1971 remarked satirically (cf. [35]) that it probably was the 1001st algorithm of Gauß. According to present knowledge he used the FFT to interpolate orbits of asteroids and wrote down the algorithm in 1805. It has never been published and can be found only in his collected works under the title "Theoria Interpolationis Methodo Nova Tractata", Gauß (1866), [31].

Rader (1968), [53] and Bluestein (1970), [10] strengthened the results of Cooley and Tukey by removing the restrictions on the input size n. With this, the problem of developing fast algorithms for computing the Fourier tranform virtually was solved.

The foundation for a deeper point of view and a considerable generalization of the DFT was presented in the habilitation thesis of Beth (1984), [6]. Already in Apple/Wintz (1970), [2] and Pichler (1975), [52] it had been realized that the DFT_n is precisely the matrix which decomposes the regular representation

$$\phi : \ x \mapsto (1, 2, \ldots, n)$$

of a cyclic group $\mathsf{Z}_n = \langle x \mid x^n = 1 \rangle$ of order n into its irreducible components, i.e.

$$\phi^{\mathrm{DFT}_n} = \bigoplus_{i=1}^{n} (x \mapsto \omega_n^i).$$

With this knowledge a natural connection arises between the Fourier transform and the representation theory of finite groups. Beth realized that the FFT can be explained through a stepwise decomposition of ϕ along a chain of subgroups of Z_n and generalized the DFT to arbitrary finite groups. First results concerning the cost of the FFT of solvable groups can be found in Beth (1987), [7].

Since the work of Beth, fast Fourier transforms for certain classes of non-abelian groups were constructed and their complexity was investigated. Important work in this direction was done by Clausen (1988/89), [13, 14], where among other things an FFT for the symmetric group S_n is constructed, Diaconis/Rockmore (1990), [23] and Rockmore (1994), [55]. A good introduction to the area of generalized Fourier transforms can be found in the book of Clausen/Baum (1993), [16]. The current status of research is presented in the survey article by Maslen/Rockmore (1995), [43]. Applications of generalized FFT's are still comparatively few and mainly lie in the area of signal processing and statistics. An overview of this is provided by Rockmore (1995), [56]. All this work is based on the decomposition of the regular representation.

An entirely new approach and generality was developed by Minkwitz in the framework of his dissertation (1993), [45] (see also [46, 48]). The central point in his work is the definition of *symmetry* of a matrix M. If ϕ, ψ are representations of the same group G then M has the symmetry (ϕ, ψ) if

$$\phi(g) \cdot M = M \cdot \psi(g) \text{ for all } g \in G.$$

Minkwitz showed that the matrix M can be decomposed into a product of sparse matrices if ϕ is a permutation representation and ψ is either a direct sum of irreducible representations (this applies to all generalized Fourier transforms) or also a permutation representation. Thus the restriction to regular representations was dropped and Minkwitz developed methods to stepwise decompose an arbitrary permutation representation of a solvable group. This *stepwise* decomposition leads to a decomposition matrix which is a product of sparse matrices. If M has the symmetry (ϕ, ψ) and A_ϕ, A_ψ are the corresponding decomposition matrices, respectively, then the matrix

$$B = A_\phi^{-1} \cdot M \cdot A_\psi$$

is block diagonal (and hence sparse) which is a consequence of Schur's Lemma from representation theory. Thus by

$$M = A_\phi \cdot B \cdot A_\psi^{-1}$$

a decomposition of M is determined. Obviously, the essential point here is that A_ϕ and A_ψ can be computed in decomposed form. In this case the decomposition of M above represents a fast algorithm for the multiplication with M. Later, Minkwitz recognized that a generalization to monomial representations ϕ, ψ is possible. The procedure for the decomposition of a matrix M accordingly is as follows:

1. Determine an appropriate symmetry (ϕ, ψ) of M.

2. Compute decomposition matrices A_ϕ, A_ψ of ϕ resp. ψ.

3. Compute the block diagonal matrix $B = A_\phi^{-1} \cdot M \cdot A_\psi$.

With this method, it is possible to decompose a *given* matrix (with symmetry). As an example Minkwitz determined (by hand) the symmetry of the discrete cosine transform (DCT) and then was able to decompose it. Prior to this result the DCT had no representation theoretical interpretation. In addition, the application of the method to the DFT resulted in the well-known Cooley-Tukey and Rader decomposition.

With the work of Minkwitz, the foundation for the automatic decomposition of a matrix with symmetry thus was developed. The difficult problem of finding symmetry of a given matrix was examined in the dissertation of Egner (1997), [26]. He called the types of symmetry discussed perm-irred symmetry, perm-perm symmetry and mon-mon symmetry and developed among other things the first algorithm for finding perm-irred symmetry.

Contribution of this Dissertation

The methods for decomposing permutation representations were only sketched in the publications of Minkwitz. Standard representation theory of finite groups provides virtually no answers to this problem, since there representations are considered only up to equivalence.

For this reason in this dissertation first a *constructive* approach to representation theory is developed, where representations are considered up to equality instead of only up to equivalence as in the standard theory. This means among other things that conjugating matrices for equivalent representations move into the focus of interest. In this sense first well-known theorems dealing with the interaction of induction and other constructions for representations are refined and later applied to permutation and monomial representations. To a certain extent a theory of decomposition matrices is developed. Further examined is also the intertwining space of representations which by Minkwitz' definition of symmetry naturally comes into play: The intertwining space $\mathsf{Int}(\phi, \psi)$ of two representations ϕ and ψ is precisely the set of matrices having (ϕ, ψ) as a symmetry. The majority of the results in this section cannot be found in literature. Using these results an algorithm for the decomposition of a large class of monomial representations is developed and presented in detail.

An efficient implementation of algorithms and data structures for the computation with structured representations and matrices has been developed in collaboration with Sebastian Egner (cf. Appendix). Based on these, all algorithms in this dissertation has been implemented. By combining these programs with programs for finding symmetry it was possible to automatically generate matrix decompositions (for matrices with symmetry).

Finally this program was applied to the several well-known classical signal transforms: each could be decomposed substantially. Thus the representation theory of finite groups provides for the considered signal transform not only the theoretical background for the existence of fast algorithms, it also provides a mechanism to generate them.

Organization

Chapter 1 first explains symbols and notation that is used. After that the constructive organization of representation theory is developed as explained above. Main results concern inductions of representations, monomial representations, intertwining spaces, and decomposition matrices. The concrete algorithmic realization is always emphasized and has been carried out (cf. Appendix).

Chapter 2 presents an efficient algorithm, based on the results of Chapter 1, for the decomposition of a large class of monomial representations. A detailed example concludes the chapter.
Chapter 3 deals with the decomposition of matrices with symmetry. First the three types of symmetry considered are defined and then, in each case, it is shown how a decomposition of a matrix can be derived from the symmetry. Finally, an algorithm is developed which is able to find monomial symmetry.
Chapter 4 applies the theory developed to concrete examples from signal processing. The symmetries of well-known classical signal transforms are computed and, using them, decompositions for the transforms are *automatically* obtained and exemplary given. They essentially correspond to the decompositions known from literature and reveal the power of the developed methods.
Appendix is the main part of the manual of the software package AREP for calculation with matrix representations which has been developed in the framework of this dissertation in collaboration with Sebastian Egner. AREP is based on the results of the Chapters 1 and 2.

1

Constructive Representation Theory

Representation theory of finite groups is considered as one of the most beautiful branches of algebra. Many of the most well-known algebraists of the century worked on it and hence it is not surprising that the basics of representation theory are almost entirely clarified.

A (matrix) representation is given by a homomorphism of a finite group G into the group $\mathsf{GL}_n(\mathbb{K})$ of invertible $(n \times n)$-matrices over a field $\mathbb{K}$. The "Maschke condition" which requires that the characteristic of $\mathbb{K}$ does not divide the group order $|G|$, divides representation theory into two essentially different branches: ordinary $(\mathrm{char}(\mathbb{K}) \not| \, |G|)$ and modular $(\mathrm{char}(\mathbb{K}) \mid |G|)$ representation theory. The former nowadays is in most points explored whereas the latter still is subject of actual research.

As references for representation theory we recommend: James/Liebeck (1993), [38] as easiest introduction into this area, Dornhoff (1971), [25] and Serre (1977), [58], to quickly obtain an overview as well as Huppert (1967), [36] and particularly Curtis/Reiner (1962), [21] resp. (1981), [22] for getting deeper into the area. The book of Feit (1982), [29] is written on a high level. Also a very good introduction provides the unpublished script of a lecture on representation theory of Leopoldt (1979), [40].

As introduction into group theory we mention: the comprehensive book of Huppert (1967), [36] and the beautiful book of Marshall Hall jr. (1976), [33]. The Atlas (1985), [18] provides an overview of the finite simple groups. For groups of order less than 1000 (except 512 and 768) the new catalogue of small finite groups of Besche und Eick (1996), [5] is available and can be loaded into the GAP system [57]. Concerning computer algebra of groups Atkinson (1984), [3] is a classic as well as the book of Butler (1991), [12] for computing with permutation groups.

In the books mentioned above representations are considered only up to equivalence, i.e. up to conjugation $\phi \to \phi^A$ by a an invertible matrix A. Accordingly the atomic objects of investigation are equivalence classes of (irreducible) representations. For the purpose of developing the structure theory usually characters of representations are considered rather than representations itself. Characters are invariant under conjugation of the underlying representation and carry most of the information of the corresponding equivalence class. Furthermore, charcacters can be efficiently represented: a character is determined by the values on the conjugacy classes of the group. The computer-based computation with characters contributed essentially to the classification of the finite simple groups.

For many applications, however, calculating with characters is not sufficient. You want to be able to compute explicitly with matrix representations transforming them preserving equality. An example for such an application is the decomposition of a matrix with symmetry. There it is necessary to stepwise decompose a monomial representation carrying along the decomposition matrix (cf. Introduction).

Hence the development of programs for computation with matrix representations requires an extension of representation theory: well-known theorems dealing with equivalence of two representations must be refined up to equality by explicit specification of a conjugating matrix, a transversal, etc. E.g. it is known that every transitive monomial representation μ of a group G is equivalent to an induction of a representation λ of degree 1 of a subgroup $H \leq G$. How do you find H and λ? How to choose a transversal and a conjugating matrix to establish equality? Questions of this kind will be answered in this chapter.

In the framework of this dissertation the software package AREP for calculation with matrix representations has been developed in collaboration with Sebastian Egner. AREP is realized in the language GAP (Groups, Algorithms and Programming, [57]) and implements a term algebra for the computation with representations and matrices as well as all algorithms presented in this dissertation. AREP will be available on the GAP server from mid 98 as a GAP share package.

The central objects in this package are the recursive data types `ARep` and `AMat`. An `ARep` is a GAP record representing a representation The record contains a number of fields which uniquely characterize a representation, e.g. degree, characteristic, and the represented group always have to be present. Now there are a number of elementary constructors allowing to create an `ARep`, e.g. by specification of the images on a set of generators of the group (`ARepByImages`). Furthermore, there are constructors building a higher structured `ARep` from given `ARep`s (e.g. `DirectSumARep`). The idea here is not to immediately evaluate such a construction, like the direct sum, but to build an `ARep` representing it, i.e. an `ARep` with a field `summands` containing the list of summands.

If you want to convert to a matrix representation the appropriate function has to be used. On the other side there are functions converting an unstructured, e.g. monomial `ARep`, into a highly structured `ARep`, e.g. a conjugated induction of a representation of degree 1, which is mathematical *identical* to the original one. Permutation and monomial representations has been given special attention in the package since they are efficiently to store and to compute with and they are of crucial importance for the decomposition of matrices with symmetry.

The data type `AMat` has been created according to the same principle as a GAP record representing a matrix. Again, there are elementary constructors, e.g. `AMatPerm` takes a permutation, a degree, and a characteristic and builds an `AMat` representing a permutation matrix. Higher constructors (product, direct sum, tensor product, etc.) are not evaluated until the corresponding function is invoked. Thus it is possible to build structured matrices which are easier to handle than the (mathematical identical) represented matrices. Furthermore, structured `AMat`s are efficient to store. A manual for AREP can be found in the Appendix.

In the following the theory and the algorithms are presented which form the basis of the functions contained in AREP. The results are partially built up from Minkwitz (1993), [45] and Clausen/Baum (1993), [16]. In the focus of interest are inductions and monomial representations. The algorithm for the decomposition of monomial representations is based on these results and will be presented in the next chapter.

The chapter is organized as follows: In Section 1.1 the terms and symbols used in this dissertation are presented, partially in tabular form. In Section 1.2 a number of theorems is presented describing the interaction of induction with other constructors for representations (restriction, tensor product, etc.). The theorems are known but will be presented constructively refined, i.e. it always holds "$=$" instead of "$\cong$". Especially concerned with monomial representations is Section 1.3 where among other things algorithms for the decomposition of a transitive monomial representation into an induction resp. a conjugated outer tensor product can be found. The Extension Theorem of Minkwitz is subject of Section 1.4. In Section 1.5 the intertwining space of representations is investigated with special attention to the monomial case. The chapter is concluded by Section 1.6 where the foundations for the decomposition of monomial representations are laid. This section culminates in a theorem allowing the stepwise calculation of decomposition matrices along a compositon series.

We exclusively consider representations of finite groups satisfying the Maschke condition. Whenever possible the theorems are formulated in terms of matrix representations avoiding the diction of modules.

1.1 Notation and Constructions

A representation ϕ of degree n is a homomorphism of a group G into the group $\mathsf{GL}_n(\mathbb{K})$ of invertible $(n \times n)$-matrices over a field $\mathbb{K}$. The group G shall be finite and the characteristic of $\mathbb{K}$ shall be no divisor of the group order $|G|$ (Maschke condition). Then, every representation ϕ can be decomposed with an invertible matrix A into a direct sum of irreducible representations ϕ_i (Theorem of Maschke). Every ϕ_i is called an irreducible component of ϕ:

$$\phi^A = \bigoplus_{i=1}^{r} (\underbrace{\phi_i \oplus \ldots \oplus \phi_i}_{n_i}), \text{ where } \phi_i \text{ irreducible, } \phi_i \not\cong \phi_j \text{ for } i \neq j.$$

A representation is called permutation representation if all images are permutation matrices. A representation is called monomial if all images are monomial matrices. A matrix is called monomial if it contains exactly one entry $\neq 0$ in every row and column. Monomial matrices are always invertible. A representation is called unitary if all images are unitary matrices.

If ϕ is a representation over $\mathbb{K}$ of degree n, then G operates on the vector space $V = \mathbb{K}^n$ via ϕ by $v \cdot g = v \cdot \phi(g),\ v \in V,\ g \in G$ making V a right $\mathbb{K}[G]$-module. We will call V the "representation space" of ϕ. Is ϕ decomposed as shown above then we will refer to the representation space of one ϕ_i as an irreducible component of V and to the representation space of $(\phi_i \oplus \ldots \oplus \phi_i)$ as a homogeneous component of V. Hence the decomposition of a representation corresponds to the decomposition of the underlying representation space into irreducible resp. homogeneous components.

We use the following conventions for notation:

Matrices:

$A, B, M, P, \ldots$	matrices
$[a_{i,j} \mid i \in \{1, \ldots, n\},$ $j \in \{1, \ldots, m\}]$	or simply $[a_{i,j}]_{i,j}$, matrix with entries $a_{i,j}$
$\mathrm{diag}(x_k \mid k \in \{1, \ldots, n\})$	diagonal matrix with entries x_k
σ, τ	permutations
$[\sigma, n]$	$[\delta_{i^\sigma j} \mid i, j \in \{1, \ldots, n\}]$, $(n \times n)$ permutation matrix corresponding to permutation σ with convention $[\tau, n] \cdot [a_{i,j}]_{i,j} \cdot [\sigma, m] = [a_{i^\tau, j^{\sigma^{-1}}}]_{i,j}$
$[\sigma, (x_1, \ldots, x_n)]$	$[\sigma, n] \cdot \mathrm{diag}(x_k \mid k \in \{1, \ldots, n\})$, monomial $(n \times n)$-matrix
ω_n	primitive nth root of unity
i	$\sqrt{-1}$

A^*	adjunct of the matrix A (transposed and complex comjugated)
DFT_n	$[\omega_n^{ij} \mid i \in \{0,\ldots,n-1\}, j \in \{0,\ldots,n-1\}]$
$\mathbf{1}_n$	identity matrix of degree n
$\mathbf{0}_n$	all-zero matrix of degree n

Groups:

$G, H, N, \ldots$	groups
$H \backslash G$	set of right cosets of H in G
$H \times N$	direct product of H and N
$H \ltimes N$	semidirect product of normal subgroup N with subgroup H
$g, h, x, y, s, t, \ldots$	group elements
$\langle x, y, \ldots \rangle$	group or vector space generated by $x, y, \ldots$
T, S	transversals (= systems of right coset representatives)
E	trivial group
Z_n	cyclic group of order n
D_{2n}	dihedral group with $2n$ elements
A_n	alternating group on n points
S_n	symmetric group on n points
Q_8	quaternion group

Representations:

$\phi, \psi, \rho, \ldots$	representations
μ	monomial representation
π	permutation representation
ϕ_G, μ_G, π_G	representations of the group G
1_G	one-representation of degree 1 of G
$\deg(\phi)$	degree of ϕ
χ_ϕ	character of ϕ

Sets and Lists:

$\{t_1, \ldots, t_n\}$	set of elements $t_1, \ldots, t_n$
$(t_1, \ldots, t_n)$	list of elements $t_1, \ldots, t_n$
$\cup$	union of sets or concatenation of lists

Furthermore, we denote with

$$A \oplus B \;=\; \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & B \end{bmatrix}$$

the *direct sum* of the matrices $A = [a_{i,j} \mid i, j \in \{0, \ldots, n-1\}]$ and $B = [b_{i,j} \mid i, j \in \{0, \ldots, m-1\}]$ and the *Kronecker product* (or tensor product) with

$$A \otimes B \;=\; \Big[a_{i \operatorname{div} m,\, j \operatorname{div} m} \cdot b_{i \operatorname{mod} m,\, j \operatorname{mod} m} \mid i, j \in \{0, \ldots, nm-1\}\Big].$$

In the Kronecker product $A \otimes B$ the coarse structure is determined by A and the fine structure by B. An example:

$$\mathbf{1}_2 \otimes \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \mathbf{1}_3 = \begin{bmatrix}
1 & \cdot & \cdot & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 1 & \cdot & \cdot & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 1 & \cdot & \cdot & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
3 & \cdot & \cdot & 4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 3 & \cdot & \cdot & 4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 3 & \cdot & \cdot & 4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 2 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 2 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 2 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 3 & \cdot & \cdot & 4 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 3 & \cdot & \cdot & 4 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 3 & \cdot & \cdot & 4
\end{bmatrix}.$$

A matrix M is called *block permuted* (with respect to the blocks $B_1, \ldots, B_n$), if

$$M = P_1 \cdot (B_1 \oplus \ldots \oplus B_n) \cdot P_2 \text{ with permutation matrices } P_1, P_2.$$

We will use the following constructions for representations:

- $\phi_G^A = g \mapsto A^{-1} \cdot \phi_G(g) \cdot A$ is a conjugated (by $A \in \mathsf{GL}_n(\mathbb{K})$) representation of G. We also write $\phi_G \xrightarrow{A} \phi_G^A$.

- $\phi_G \oplus \psi_G = g \mapsto \phi_G(g) \oplus \psi_G(g)$ is the direct sum of the representations ϕ_G, ψ_G of the same group G.

- $\phi_G \otimes \psi_G = g \mapsto \phi_G(g) \otimes \psi_G(g)$ is the inner tensor product of the representations ϕ_G, ψ_G of the same group G and again is a representation of G.

- $\phi_G \# \psi_H = (g, h) \mapsto \phi_G(g) \# \psi_H(h)$ is the outer tensor product of the representations ϕ_G of G and ψ_H of H. The outer tensor product is a representation of the direct product $G \times H$.

- $\lambda_G \cdot \phi_G = g \mapsto \lambda_G(g) \cdot \phi_G(g)$ is the linear multiple of the representation ϕ_G with the representation λ_G of degree 1. It is a special case of an inner tensor product.

- $\phi_G \downarrow H = h \mapsto \phi_G(h)$ is the restriction of the representation ϕ_G of G to the subgroup H.

- $\phi_H \uparrow_T G$ denotes the induction of the representation ϕ_H to G with T where H is a subgroup of G and $T = (t_1, \ldots, t_n)$ a transversal (system of right coset representatives) of $H\backslash G$ with length $n = (G : H)$. Since the equivalence class of the induction is independent of the choice of transversal we will omit it when calculating only up to equivalence. The degree of the induction is $\deg(\phi_H) \cdot n$. It is defined as

$$\begin{aligned} \phi_H \uparrow_T G &= g \mapsto \left[\dot{\phi}_H(t_i \cdot g \cdot t_j^{-1}) \,\middle|\, i,j \in \{1,\ldots,n\}\right], \quad \text{with} \\ & \dot{\phi}_H(x) = \begin{cases} \phi_H(x), & x \in H \\ \mathbf{0}_{\deg(\phi_H)}, & \text{else} \end{cases}. \end{aligned}$$

  If ϕ_H is of degree 1, then the induction is monomial. If even $\phi_H = 1_H$, then the induction is a permutation representation.

- A regular representation of a group is a special case of a permutation representation given by $1_{\mathsf{E}} \uparrow G$, where E denotes the trivial group.

- The extension of a representation ϕ of H to a supergroup G of H is denoted by $\overline{\phi}$. In contrast to the induction the extension of a representation of a subgroup does not exist in general.

- $\phi_H^t = g \mapsto \phi_H(t \cdot g \cdot t^{-1})$ is the inner conjugate of a representation ϕ_H of H by an element t of a supergroup G of H. ϕ_H^t is a representation of the conjugated subgroup $H^t = t^{-1}Ht$. If in particular H is normal in G then the inner conjugate of any representation of H again is a representation of H, however, in general not equivalent to the original one. The definition implies the following rule:

$$\left(\phi_H^t\right)^s = \phi_H^{ts} = g \mapsto \phi_H(tsgs^{-1}t^{-1}),$$

  i.e. g *first* is conjugated by the inverse of the *outer* exponent.

Tabular 1.5 gives an overview of the interaction of the presented constructions. We denote by $(\cdot)^A$ the conjugation of a representation with a matrix A and with $(\cdot)^t$ the inner conjugation by a group element. The tabular can be read as follows. If first the operation in a row is performed and then the operation in a column then the corresponding entry in the tabular is: nothing, if no general transformation rule exists, a cross, if such a rule exists but trivial, the number of the theorem, if the rule is non-trivial. E.g.

the expression $(\phi_1 \oplus \phi_2)^A$ cannot be transformed for arbitrary representations ϕ_1, ϕ_2 and matrix A (of appropriate size) hence the entry of row 2, column 1 is empty. The associativity of $\oplus, \otimes$ can be found on the main diagonal at position 2, 3 respectively. The (trivial) rule $(\phi \downarrow H)^A = \phi^A \downarrow H$ leads to a cross at positions $(6, 1)$ and $(1, 6)$. The Theorem of Mackey allowing to decompose the restriction of an induction can be found in row 5, column 6.

	$(\cdot)^A$	$\oplus$	$\otimes$	$\#$	$\uparrow$	$\downarrow$	$(\cdot)^t$
$(\cdot)^A$	$\times$	$\times$	$\times$	$\times$	1.6	$\times$	$\times$
$\oplus$		$\times$	$\times$	$\times$	1.5	$\times$	$\times$
$\otimes$			$\times$			$\times$	$\times$
$\#$				$\times$	1.7		$\times$
$\uparrow$			1.17	1.7	1.4	1.12	1.11
$\downarrow$	$\times$			$\times$	1.15	$\times$	$\times$
$(\cdot)^t$	$\times$			$\times$	1.10	$\times$	$\times$

Table 1.5: Interaction of constructions

It is apparent that all non-trivial transformation rules are linked with the induction. This might be explained by the fact that the induction is the only non-trivial constructor for representations. The theorems mentioned all can be found in the following section.

1.2 Induction

In this section we provide the essential theorems allowing to calculate with arbitrary inductions of representations.

Change of Transversal It is known that the equivalence class of an induction of a representation ϕ of $H \leq G$ is independent of the choice of transversal, i.e.

$$\phi \uparrow_T G \cong \phi \uparrow_{T'} G.$$

We want to determine the matrix corresponding to the pair (T, T') which establishes equality. Let ϕ be a representation of H, $n = (G : H)$ and $T = (t_1, \ldots, t_n)$ an arbitrary transversal of $H \backslash G$. First we consider two particular cases of change of transversal. Permutation of T with $\sigma \in \mathsf{S}_n$ leads to another transversal

$$T' = T^\sigma = (t_{1^{\sigma^{-1}}}, \ldots, t_{n^{\sigma^{-1}}})$$

and the induction with T' can be calculated as

$$\begin{aligned}(\phi \uparrow_{T'} G)(x) &= \left[\dot\phi(t'_i x {t'_j}^{-1})\right]_{i,j}\\ &= \left[\dot\phi(t_{i^{\sigma^{-1}}} x t^{-1}_{j^{\sigma^{-1}}})\right]_{i,j}\\ &= \left([\sigma^{-1}, n] \otimes \mathbf{1}_{\deg(\phi)}\right) \cdot \left[\dot\phi(t_i x t_j^{-1})\right]_{i,j} \cdot \left([\sigma, n] \otimes \mathbf{1}_{\deg(\phi)}\right)\\ &= (\phi \uparrow_T G)(x)^{[\sigma,n]\otimes \mathbf{1}_{\deg(\phi)}}.\end{aligned}$$

Change of coset representatives leads to the transversal

$$T' = (h_1 t_1, \ldots, h_n t_n),\ \ h_i \in H,$$

having the following effect on the induction:

$$\begin{aligned}(\phi \uparrow_{T'} G)(x) &= \left[\dot\phi(t'_i x {t'_j}^{-1})\right]_{i,j}\\ &= \left[\dot\phi(h_i t_i x t_j^{-1} h_j^{-1})\right]_{i,j}\\ &= \left[\dot\phi(t_i x t_j^{-1})\right]^D_{i,j},\end{aligned}$$

where $D = \bigoplus_{i=1}^n \phi(h_i^{-1})$ is a block-diagonal matrix with blocks of size $\deg(\phi)$. The general case of change of transversal can be composed from these particular cases.

1.1 Theorem *Let $H \le G$ be a subgroup and ϕ a representation of H and let $T = (t_1, \ldots, t_n)$ and $T' = (t'_1, \ldots, t'_n)$ be two transversals of $H\backslash G$. Assume that σ is the permutation in S_n mapping the cosets $(Ht_1, \ldots, Ht_n)$ on the cosets $(Ht'_1, \ldots, Ht'_n)$. Then*

$$\phi \uparrow_{T'} G = (\phi \uparrow_T G)^M \ \textit{with}\ M = ([\sigma, n] \otimes \mathbf{1}_{\deg(\phi)}) \cdot \bigoplus_{i=1}^{n} \phi\left(t_{i^{\sigma^{-1}}} \cdot {t'_i}^{-1}\right).$$

We call M the matrix corresponding to the change of transversal $T \to T'$.

Proof We have $\phi \uparrow_{T^\sigma} G = (\phi \uparrow_T G)^{[\sigma,n]\otimes\mathbf{1}_{\deg(\phi)}}$ according to the calculation above. The transition from T^σ to T' only is a change of coset representatives and hence $\phi \uparrow_{T'} G = (\phi \uparrow_{T^\sigma} G)^D$ with $D = \bigoplus_{i=1}^n \phi(t_{i^{\sigma^{-1}}} \cdot {t'_i}^{-1})$ and the result follows. ∎

The algorithm for the change of transversal follows this proof.

1.2 Algorithm (Change of Transversal) Given is an induction $\phi \uparrow_T G$ of a representation ϕ of a subgroup $H \le G$ and a second transversal T' of $H\backslash G$. The matrix M in Theorem 1.1 corresponding to the change of transversal $T \to T'$ shall be computed.

1. Determine the permutation σ mapping the list of cosets $(Ht_1, \dots, Ht_n)$ onto $(Ht'_1, \dots, Ht'_n)$.
2. Evaluate ϕ at the elements $t_{i^{\sigma^{-1}}} \cdot {t'}_i^{-1}$, $i = 1 \dots n$.

Then $M = ([\sigma, n] \otimes \mathbf{1}_{\deg(\phi)}) \cdot \bigoplus_{i=1}^{n} \phi\left(t_{i^{\sigma^{-1}}} \cdot {t'}_i^{-1}\right)$. ∎

1.3 Corollary Algorithm 1.2 needs $O(n \log(n))$ comparisons of cosets and n evaluations of ϕ.

Proof To determine σ the lists $(Ht_1, \dots, Ht_n)$ and $(Ht'_1, \dots, Ht'_n)$ are sorted and σ is the quotient of the sorting permutations. The rest is trivial. ∎

The change of transversal is the most important basic routine for the computation with inductions. The following theorems reveal the interaction of the induction with other operations. In all cases equality is achieved by choosing a certain transversal. In oder to perform a transformation the only thing to do is a change of the given transversal to the required one and the representation transforms.

Double Induction Induction is a transitive operation. If ϕ is a representation of H and $H \leq K \leq G$ then (cf. Figure 1.1)

$$\phi \uparrow G \cong (\phi \uparrow K) \uparrow G.$$

Here we establish equality by appropriate choice of the transversal.

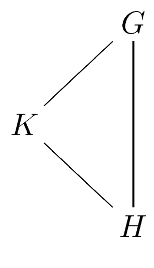

Figure 1.1: Situation for double induction

1.4 Theorem *Let $H \leq K \leq G$ be groups and ϕ a representation of H. Suppose $S = (s_1, \dots, s_m)$ and $T = (t_1, \dots, t_n)$ are transversals of $H \backslash K$ and $K \backslash G$ respectively. Then*

$$\phi \uparrow_{TS} G = (\phi \uparrow_T K) \uparrow_S G,$$

where $TS = (t_1s_1, \ldots, t_ns_1, t_1s_2, \ldots, t_ns_2, \ldots, t_1s_m, \ldots, t_ns_m)$ *denotes the complex product of the transversals* S *and* T.

Proof TS is a transversal of $H\backslash G$ and

$$\begin{aligned}(\phi \uparrow_T K) \uparrow_S G &= \left(x \mapsto \left[\dot\phi(t_ixt_j^{-1})\right]_{i,j}\right) \uparrow_S G \\ &= x \mapsto \left[\dot\phi(t_is_kxs_\ell^{-1}t_j^{-1})\right]_{(k,i),(\ell,j)} \\ &= \phi \uparrow_{TS} G,\end{aligned}$$

where

$$\dot\phi(y) = \begin{cases} \phi(y), & y \in H \\ \mathbf{0}_{\deg(\phi_H)}, & \text{else} \end{cases}.$$

■

The previous theorem is essential for the decomposition of a monomial representation. It allows to decompose an induction into small steps along a chain of subgroups. The only thing to do is a change of transversal according to Theorem 1.1.

Direct Sum Induction is additive, i.e. if ϕ_1 and ϕ_2, are representations of $H \leq G$ then

$$(\phi_1 \oplus \phi_2) \uparrow G \cong (\phi_1 \uparrow G) \oplus (\phi_2 \uparrow G).$$

Equality can be achieved by a permutation matrix.

1.5 Theorem *Let* $H \leq G$ *be a subgroup with representations* ϕ_1 *and* ϕ_2 *of degrees* d_1 *and* d_2 *respectively. Assume* T *is a transversal of* $H\backslash G$ *of length* n*. For brevity let* $d = d_1 + d_2$*. Denote with* σ *the permutation mapping the list (* $\cup =$ *concatenation)*

$$\bigcup_{k=0}^{n-1} (k \cdot d + 1, \ldots, k \cdot d + d_1) \;\cup\; \bigcup_{k=0}^{n-1} (k \cdot d + d_1 + 1, \ldots, (k+1) \cdot d)$$

onto $(1, \ldots, n \cdot d)$*. Then*

$$((\phi_1 \oplus \phi_2) \uparrow_T G)^{[\sigma, n \cdot d]} = (\phi_1 \uparrow_T G) \oplus (\phi_2 \uparrow_T G).$$

Proof The induction is of degree $n \cdot d$. The first concatenation corresponds to the indices of the base vectors of the representation space of $\phi_1 \uparrow G$ in $(\phi_1 \oplus \phi_2) \uparrow G$, analogous for the second concatenation. The corresponding change of bases decomposes the representation. ■

Conjugation Induction of equivalent representations ϕ and $\psi = \phi^A$ leads to equivalent results. The conjugating matrix can be stated immediately.

1.6 Theorem *Let $H \leq G$ be a subgroup of index n with representation ϕ over $\mathbb{K}$ of degree d and T a transversal of $H\backslash G$. Assume $A \in \mathsf{GL}_d(\mathbb{K})$, then*

$$(\phi^A \uparrow_T G) = (\phi \uparrow_T G)^{(\mathbf{1}_n \otimes A)}.$$

Proof For $x \in G$ we have

$$\begin{aligned}(\phi^A \uparrow_T G)(x) &= \left[\dot{\phi^A}(t_i x t_j^{-1})\right]_{i,j} \\ &= \left[A^{-1} \cdot \dot{\phi}(t_i x t_j^{-1}) \cdot A\right]_{i,j} \\ &= (\phi \uparrow_T G)(x)^{(\mathbf{1}_d \otimes A)},\end{aligned}$$

as desired. ∎

If in particular A is a decomposition matrix for ϕ then we can compute a permutation matrix P with Theorem 1.5 such that $(\mathbf{1}_d \otimes A) \cdot P$ decomposes $\phi \uparrow G$ (however, in general not in irreducibles).

Outer Tensor Product Assume G_1, G_2 are groups and $H_1 \leq G_1$, $H_2 \leq G_2$ subgroups with representations ϕ_1, ϕ_2. A well-known theorem (cf. e.g. [21], p. 316) says (cf. Figure 1.2)

$$(\phi_1 \uparrow G_1) \# (\phi_2 \uparrow G_2) \cong (\phi_1 \# \phi_2) \uparrow (G_1 \times G_2).$$

As before we can achieve equality by appropriate choice of transversal. The following theorem is the base for the constructive decomposition of a monomial representation into an outer tensor product.

1.7 Theorem *Let $H_1 \leq G_1$ and $H_2 \leq G_2$ be subgroups with representations ϕ_1 and ϕ_2. Assume $T_1 = (t_1^{(1)}, \ldots, t_n^{(1)})$ and $T_2 = (t_1^{(2)}, \ldots, t_m^{(2)})$ are transversals of $H_1\backslash G_1$ and $H_2\backslash G_2$ respectively. Then*

$$(\phi_1 \uparrow_{T_1} G_1) \# (\phi_2 \uparrow_{T_2} G_2) = (\phi_1 \# \phi_2) \uparrow_{T_1 \times T_2} (G_1 \times G_2),$$

where $T_1 \times T_2 = \left((t_1^{(1)}, t_1^{(2)}), (t_1^{(1)}, t_2^{(2)}), \ldots\right)$ denotes the Cartesian product of the lists T_1 and T_2.

$$\begin{tikzcd}
& G_1 \times G_2 \arrow[dl, dash] \arrow[dr, dash] \arrow[dd, dash] & \\
G_1 \arrow[dd, dash] & & G_2 \arrow[dd, dash] \\
& H_1 \times H_2 \arrow[dl, dash] \arrow[dr, dash] & \\
H_1 & & H_2
\end{tikzcd}$$

Figure 1.2: Situation for the outer tensor product of inductions

Proof $T_1 \times T_2$ is a transversal of $(H_1 \times H_2)\backslash(G_1 \times G_2)$ and

$$\begin{aligned}
& ((\phi_1 \uparrow_{T_1} G_1) \#(\phi_2 \uparrow_{T_2} G_2))\,(x_1, x_2) \\
= \; & \left[\dot{\phi}_1(t_i^{(1)} x_1 t_j^{(1)^{-1}})\right]_{i,j} \otimes \left[\dot{\phi}_2(t_k^{(2)} x_2 t_\ell^{(2)^{-1}})\right]_{k,\ell} \\
= \; & \left[(\phi_1 \,\dot{\#}\, \phi_2)\left((t_i^{(1)} x_1 t_j^{(1)^{-1}}), (t_k^{(2)} x_2 t_\ell^{(2)^{-1}})\right)\right]_{(i,k),(j,\ell)} \\
= \; & ((\phi_1 \# \phi_2) \uparrow_{T_1 \times T_2} (G_1 \times G_2))\,(x_1, x_2),
\end{aligned}$$

as desired. ■

If the products $H_1 \times H_2$ and $G_1 \times G_2$ in Theorem 1.7 are inner direct products then $T_1 \times T_2 = T_1 T_2$ is just the complex product of the transversals. In order to decompose an induction into an outer tensor product, however, it is not sufficient that the represented group is a direct product. In Section 1.3 we will learn a necessary and sufficient criterion for the existence of such a decomposition in the particular case of a monomial representation.

Inner Conjugation In this section we explore the interaction of induction and inner conjugation. First we want to recall the definition of inner conjugation.

1.8 Definition *Let $H \leq G$ be a subgroup, ϕ a representation of H and $s \in G$. Then*

$$\phi^s = x \mapsto \phi(sxs^{-1})$$

is a representation of $H^s = s^{-1}Hs$ and is called the "inner conjugate" of ϕ with s.

Again, we want to emphasize that this definition implies the following computation rule:

$$\left(\phi_H^t\right)^s = \phi^{ts} = g \mapsto \phi_H(tsgs^{-1}t^{-1}),$$

i.e. g *first* is conjugated with the inverse of the *outer* exponent.

1.9 Example We consider $\mathsf{S}_3 = \langle x, y \mid x^3 = y^2 = 1,\ x^y = x^{-1}\rangle$ with normal subgroup $\mathsf{Z}_3 = \langle x\rangle \trianglelefteq \mathsf{S}_3$. Z_3 has the representation $\lambda :\ x \mapsto \omega_3$ (ω_3 is a primitive 3rd root of unity) and

$$\lambda^y :\ x \mapsto \lambda(yxy^{-1}) = \lambda(x^{-1}) = \omega_3^{-1}$$

yielding $\lambda \ncong \lambda^y = \lambda^{-1}$. ■

The induction of a representation ϕ of $H \leq G$ can be expressed as an induction of ϕ^s.

1.10 Theorem *Let $H \leq G$ be a subgroup, $s \in G$, ϕ a representation of H, and T a transversal of $H\backslash G$. Then*

$$\phi \uparrow_T G = \phi^s \uparrow_{s^{-1}T} G.$$

Proof Let $T = (t_1, \ldots, t_n)$ and $x \in G$:

$$\begin{aligned}(\phi \uparrow_T G)(x) &= \left[\dot{\phi}(t_i x t_j^{-1})\right]_{i,j} \\ &= \left[\dot{\phi}^s(s^{-1}t_i x t_j^{-1}s)\right]_{i,j} \\ &= (\phi^s \uparrow_{s^{-1}T} G)(x).\end{aligned}$$

Note that $s^{-1}T$ is a transversal of $H^s\backslash G$. ■

In particular we have $\phi \uparrow G \cong \phi^s \uparrow G$. What happens with the inner conjugate of an induction?

1.11 Theorem *Let $H \leq K \leq G$ be subgroups, ϕ a representation of H, T a transversal of $H\backslash K$, and $s \in G$. Then*

$$(\phi \uparrow_T K)^s = \phi^s \uparrow_{T^s} K^s,$$

where $T^s = (s^{-1}t_1 s, \ldots, s^{-1}t_n s)$ denotes the element-wise conjugate of the transversal $T = (t_1, \ldots, t_n)$ by s.

Proof With $x \in K^s$ we observe

$$\begin{aligned}
(\phi \uparrow_T K)^s(x) &= (\phi \uparrow_T K)(sxs^{-1}) \\
&= \left[\dot{\phi}(t_i sxs^{-1} t_j^{-1})\right]_{i,j} \\
&= \left[\dot{\phi}^s(t_i^s x (t_j^s)^{-1})\right]_{i,j} \\
&= (\phi^s \uparrow_{T^s} K^s)(x)
\end{aligned}$$

as desired. ∎

Restriction The Subgroup Theorem of Mackey allows the partial decomposition of the restriction of an induction to an arbitrary subgroup. Assume $H, K \leq G$ are subgroups and ϕ is a representation of H. Then

$$(\phi \uparrow G) \downarrow K \cong \bigoplus_{s \in S} (\phi^s \downarrow (H^s \cap K)) \uparrow K$$

where S denotes a system of representatives of the double cosets $H \backslash G / K = \{HgK \mid g \in G\}$ (cf. Figure 1.3). Of course, the indices $(K : (H^s \cap K)),\ s \in S$ are in general different. We will give transversals for the inductions in order to establish equality.

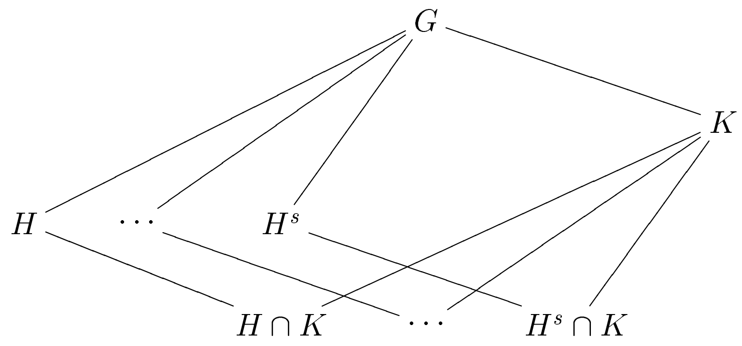

Figure 1.3: Situation for Mackey's Subgroup Theorem

1.12 Theorem (Subgroup Theorem of Mackey) *Assume $H, K \leq G$ are subgroups, ϕ is a representation of H and $S = (s_1, \ldots, s_n)$ is a system of representatives of the double cosets $H \backslash G / K$. Assume $T_i = (t_{i,1}, \ldots, t_{i,r_i})$, $i = 1 \ldots n$ are transversals of $(H^{s_i} \cap K) \backslash K$. Then the concatenation*

$$T = \bigcup_{i=1}^{n} s_i T_i \text{ is a transversal of } H \backslash G$$

and

$$(\phi \uparrow_T G) \downarrow K = \bigoplus_{i=1}^{n} (\phi^{s_i} \downarrow (H^{s_i} \cap K)) \uparrow_{T_i} K.$$

For $s \in G$ the equivalence class of the representation $(\phi^s \downarrow (H^s \cap K)) \uparrow K$ only depends on the double coset HsK.

Proof T is a transversal: T is of right length because of $(G : H) = \sum_{i=1}^{n}(K : (H^{s_i} \cap K))$. Suppose further $x, y \in T, x \neq y$:

Case 1: $\exists i : x, y \in T_i$, hence $x = s_i t_{i,j},\ y = s_i t_{i,k},\ j \neq k$,

$$xy^{-1} \in H \Leftrightarrow s_i t_{i,j} {t_{i,k}}^{-1} s_i^{-1} \in H \Leftrightarrow t_{i,j} {t_{i,k}}^{-1} \in H^{s_i}$$

which contradicts the fact that T_i is a transversal of $(H^{s_i} \cap K)\backslash K$.

Case 2: $x = s_i t_{i,k},\ y = s_j t_{j,\ell},\ i \neq j$,

$$xy^{-1} \in H \Leftrightarrow s_i t_{i,k} {t_{j,\ell}}^{-1} s_j^{-1} \in H \Leftrightarrow Hs_i \underbrace{t_{i,k} {t_{j,\ell}}^{-1}}_{\in K} = Hs_j,$$

a contradiction to the fact that s_i, s_j are elements of different double cosets.

Let $x \in K$:

$$\begin{aligned}
(\phi \uparrow_T G)(x) &= \left[\dot{\phi}(t_i x t_j^{-1}) \;\middle|\; i, j \in \{1, \ldots, \sum_{k=1}^{n} r_k\}\right] \\
&= \bigoplus_{k=1}^{n} \left[\dot{\phi}(s_k t_{k,i} x {t_{k,j}}^{-1} s_k^{-1}) \mid i, j \in \{1, \ldots, r_k\}\right] \\
&= \bigoplus_{k=1}^{n} \left[\dot{\phi^{s_k}}(t_{k,i} x {t_{k,j}}^{-1}) \mid i, j \in \{1, \ldots, r_k\}\right] \\
&= \bigoplus_{k=1}^{n} ((\phi^{s_k} \downarrow (H^{s_k} \cap K)) \uparrow_{T_k} K)(x).
\end{aligned}$$

It remains to show that for arbitrary $s \in G$ the equivalence class of the induction $(\phi^s \downarrow (H^s \cap K)) \uparrow K$ only depends on HsK. For this purpose assume that U is a transversal of $H^s\backslash G$ of length n and $h \in H$. Then

$$\begin{aligned}
\left(\phi^{hs} \downarrow (H^{hs} \cap K)\right) \uparrow_U K &= (\phi^s \downarrow (H^s \cap K))^{\phi(h^{-1})} \uparrow_U K \\
&= ((\phi^s \downarrow (H^s \cap K)) \uparrow_U K)^{\mathbf{1}_n \otimes \phi(h^{-1})}
\end{aligned}$$

according to Theorem 1.6. Now suppose that $k \in K$ and V is a transversal of $H^{sk}\backslash G$. We obtain

$$\begin{aligned} \left(\phi^{sk} \downarrow (H^{sk} \cap K)\right) \uparrow_V K &= (\phi^s \downarrow (H^s \cap K))^k \uparrow_V K \\ &= (\phi^s \downarrow (H^s \cap k)) \uparrow_{kV} K \end{aligned}$$

according to Theorem 1.10. This concludes the proof. ∎

In the particular case that $\phi = 1_H$ is the one-representation, i.e. $\phi \uparrow G$ is a permutation representation, Mackey's theorem yields exactly the decomposition of $(1_H \uparrow G) \downarrow K$ into its transitive constituents. The following two cases will play an important role.

1.13 Corollary If $N \trianglelefteq G$, ϕ a representation of N, and T a transversal of G/N then

$$(\phi \uparrow_T G) \downarrow N = \bigoplus_{t \in T} \phi^t.$$

Proof Follows from $N\backslash G/N = G/N$ and Theorem 1.12. ∎

1.14 Corollary Let $H \leq G$, $N \trianglelefteq G$ with $HN = G$, ϕ a representation of H and T a transversal of $(N \cap H)\backslash N$. Then T is also a transversal of $H\backslash G$ and

$$(\phi \uparrow_T G) \downarrow N = (\phi \downarrow (N \cap H)) \uparrow_T N.$$

Proof Because of $H\backslash G/N = HN\backslash G = G\backslash G$ there is only one double coset with representative 1. The rest follows from Theorem 1.12. ∎

The following equivalence dealing with the induction of a restriction is rarely found in standard books. Assume $H \leq G$ and ϕ is a representation of G, then

$$(\phi \downarrow H) \uparrow G \cong (1_H \uparrow G) \otimes \phi.$$

This will be proven constructively in the following theorem.

1.15 Theorem *Let $H \leq G$ be a subgroup, ϕ a representation of G and $T = (t_1, \ldots, t_n)$ a transversal of $H\backslash G$. Then*

$$((\phi \downarrow H) \uparrow_T G)^D = (1_H \uparrow_T G) \otimes \phi, \text{ with } D = \bigoplus_{t \in T} \phi(t).$$

Proof For $x \in G$ we have

$$\begin{aligned} ((\phi \downarrow H) \uparrow_T G)^D(x) &= \left[\phi(t_i)^{-1}(\phi \dot{\downarrow} H)(t_i x t_j^{-1})\phi(t_j)\right]_{i,j} \\ &= \left[\dot{1}_H(t_i x t_j^{-1}) \cdot \phi(x)\right]_{i,j} \\ &= (1_H \uparrow_T G)(x) \otimes \phi(x). \end{aligned}$$

The second "=" holds because of: if $t_i x t_j^{-1} \in H$, then the block

$$\phi(t_i)^{-1}(\phi \dot{\downarrow} H)(t_i x t_j^{-1})\phi(t_j) = \phi(x),$$

and the all-zero matrix else. ∎

Inner Tensor Product The partial decomposition of the inner tensor product of two inductions is described in the so-called *Tensor Product Theorem* Suppose $H_1, H_2 \leq G$ are subgroups with representations ϕ_1, ϕ_2 and S is a system of representatives of the double cosets $H_1 \backslash G / H_2$ then

$$(\phi_1 \uparrow G) \otimes (\phi_2 \uparrow G) \cong \bigoplus_{s \in S} (\phi_1^s \downarrow (H_1^s \cap H_2) \otimes \phi_2 \downarrow (H_1^s \cap H_2)) \uparrow G.$$

In order to formulate this theorem constructively we introduce a notation which we are going to use only in this paragraph.

1.16 Definition *Let G be a group. We denote by $\overline{G} = \{(g, g) \mid g \in G\}$ the diagonal embedding of G in $G \times G$. For $H \subseteq G$ we analogously denote $\overline{H} = \{(h, h) \mid h \in H\}$. If ϕ is a representation of G then*

$$\overline{\phi}: \ (g, g) \mapsto \phi(g)$$

denotes the corresponding representation of $\overline{G}$.

Note that in the following theorem (and only there) the symbol $\overline{\phi}$ does not denote an extension of ϕ.

1.17 Theorem (Tensor Product Theorem) *Let $H_1, H_2 \leq G$ be subgroups with representations ϕ_1, ϕ_2 and R_i a transversal of $H_i \backslash G$, $i = 1, 2$.*

Assume $S = ((x_1, y_1), \ldots, (x_n, y_n))$ is a system of representatives of the double cosets $(H_1 \times H_2) \backslash (G \times G) / \overline{G}$, then $S' = (x_1 y_1^{-1}, \ldots, x_n y_n^{-1})$ is a system of representatives of the double cosets $H_1 \backslash G / H_2$ and

$$(H_1 \times H_2)^{(x_i, y_i)} \cap \overline{G} = \overline{H_1^{x_i} \cap H_2^{y_i}}.$$

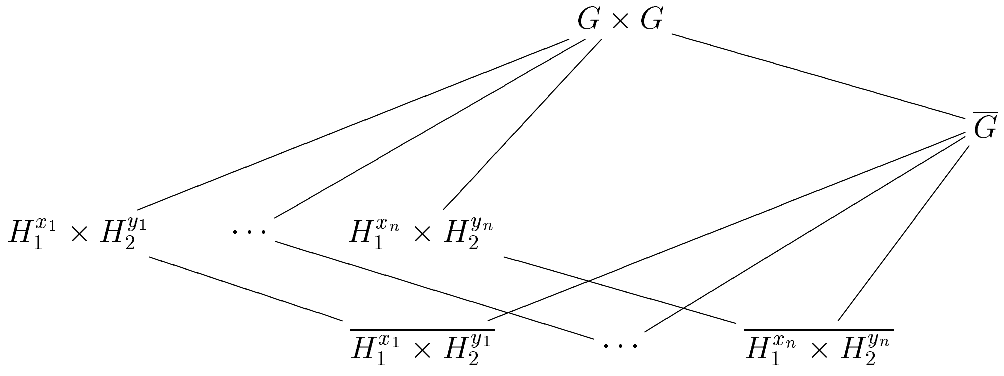

Figure 1.4: Situation for the Tensor Product Theorem

Let further T_i be a transversal of $(H_1^{x_i} \cap H_2^{y_i})\backslash G,\ i = 1 \ldots n$, and M the (block permuted) matrix corresponding to the change of transversals $(R_1, R_2) \to T = \bigcup_{i=1}^n (x_i, y_i) \cdot \overline{T_i}$ of the following representation (cf. Theorem 1.1):

$$\Big((\phi_1 \# \phi_2) \uparrow_{(R_1,R_2)} (G \times G)\Big)^M = (\phi_1 \# \phi_2) \uparrow_T (G \times G).$$

Then (cf. Figure 1.4)

$$((\phi_1 \uparrow_{R_1} G) \otimes (\phi_2 \uparrow_{R_2} G))^M =$$

$$\bigoplus_{i=1}^n ((\phi_1^{x_i} \downarrow (H_1^{x_i} \cap H_2^{y_i})) \otimes (\phi_2^{y_i} \downarrow (H_1^{x_i} \cap H_2^{y_i}))) \uparrow_{T_i} G.$$

The equivalence class of the representation $(\phi_1^{x_i} \downarrow (H_1^{x_i} \cap H_2^{y_i})) \otimes (\phi_2^{y_i} \downarrow (H_1^{x_i} \cap H_2^{y_i}))$ only depends on the double coset $H_1 \cdot x_i y_i^{-1} \cdot H_2$.

Proof First we are going to show the correspondence of the double cosets given as $(H_1 \times H_2)\backslash(G \times G)/\overline{G}$ and the double cosets $H_1\backslash G/H_2$.

$$\begin{array}{ll} & (x_1, x_2), (y_1, y_2) \text{ are in 2 different} \\ & \text{double cosets } (H_1 \times H_2)\backslash(G \times G)/\overline{G} \\ \Leftrightarrow & \forall g \in G: \ (x_1, x_2) \cdot (g, g) \cdot (y_1, y_2)^{-1} \notin (H_1 \times H_2) \\ \Leftrightarrow & \forall g \in G: \ (x_1 g y_1^{-1}, x_2 g y_2^{-1}) \notin (H_1 \times H_2) \\ \Leftrightarrow & x_1^{-1} H_1 y_1 \cap x_2^{-1} H_2 y_2 = \emptyset \\ \Leftrightarrow & H_1 \cap x_1 x_2^{-1} H_2 (y_1 y_2^{-1})^{-1} = \emptyset \\ \Leftrightarrow & x_1 x_2^{-1}, y_1 y_2^{-1} \text{ are in 2 different} \\ & \text{double cosets } H_1\backslash G/H_2. \end{array}$$

The equation

$$(H_1 \times H_2)^{(x_i, y_i)} \cap \overline{G} = \overline{H_1^{x_i} \cap H_2^{y_i}}$$

can easily be verified. Let ψ_1, ψ_2 be representations of G. The inner tensor product of ψ_1 and ψ_2 can be viewed as restriction of the outer tensor product, i.e. the following holds for the diagonal embeddings in $G \times G$:

$$\overline{\psi_1 \otimes \psi_2} = (\psi_1 \# \psi_2) \downarrow \overline{G}.$$

We derive

$$\begin{array}{ll} & \overline{((\phi_1 \uparrow_{R_1} G) \otimes (\phi_2 \uparrow_{R_2} G))}^M \\ = & ((\phi_1 \uparrow_{R_1} G) \# (\phi_2 \uparrow_{R_2} G))^M \downarrow \overline{G} \\ = & \left((\phi_1 \# \phi_2) \uparrow_{(R_1, R_2)} G\right)^M \downarrow \overline{G} \\ & \text{(Theorem 1.7)} \\ = & ((\phi_1 \# \phi_2) \uparrow_T G) \downarrow \overline{G} \\ & \text{(Theorem of Mackey, 1.12)} \\ = & \bigoplus_{i=1}^{n} \left((\phi_1 \# \phi_2)^{(x_i, y_i)} \downarrow ((H_1 \times H_1)^{(x_i, y_i)} \cap \overline{G})\right) \uparrow_{\overline{T_i}} \overline{G} \\ = & \bigoplus_{i=1}^{n} \left((\phi_1 \# \phi_2)^{(x_i, y_i)} \downarrow (\overline{H_1^{x_i} \cap H_2^{y_i}})\right) \uparrow_{\overline{T_i}} \overline{G} \\ = & \bigoplus_{i=1}^{n} \left(\overline{(\phi_1^{x_i} \downarrow (H_1^{x_i} \cap H_2^{y_i}))} \otimes \overline{(\phi_2^{y_i} \downarrow (H_1^{x_i} \cap H_2^{y_i}))}\right) \uparrow_{\overline{T_i}} \overline{G}. \end{array}$$

Omitting the dash on both sides yields the result. The proof that the equivalence class of the representation $(\phi_1^{x_i} \downarrow (H_1^{x_i} \cap H_2^{y_i})) \otimes (\phi_2^{y_i} \downarrow (H_1^{x_i} \cap H_2^{y_i}))$ only depends on the double coset $H_1 \cdot x_i y_i^{-1} \cdot H_2$ is analogous to the corresponding statement in Theorem of Mackey 1.12. ∎

Assume $\phi_1 = 1_{H_1}$ and $\phi_2 = 1_{H_2}$ are one-representations, i.e. $\phi_1 \uparrow G$ and $\phi_2 \uparrow G$ permutation representations. Then the Tensor Product Theorem just describes the decomposition of the inner tensor product into its transitive constituents.

Kernel of an Induction The kernel of an induction can be computed as follows.

1.18 Theorem *Let ϕ be a representation of $H \leq G$. Then* $\ker(\phi \uparrow G)$ *is the normal intersection of* $\mathrm{core}(H)$ *and* $\ker(\phi)$*, i.e. the largest normal subgroup of* $\mathrm{core}(H)$ *and* $\ker(\phi)$ *in G, where*

$$\mathrm{core}(H) = \bigcap_{t \in G} H^t \trianglelefteq G$$

denotes the intersection of all conjugated subgroups of H.

Proof Let $T = (t_1, \ldots, t_n)$ be a transversal of $H \backslash G$ and $x \in \ker(\phi \uparrow_T G)$. Then $txt^{-1} \in H$ for all $t \in T$ and $txt^{-1} \in \ker(\phi)$. The former means $x \in \mathrm{core}(H)$, the latter shows $x \in \ker(\phi)$ and since $\ker(\phi \uparrow_T G) \trianglelefteq G$ we get that x is in the normal intersection of $\mathrm{core}(H)$ and $\ker(\phi)$.

For the reverse let x be an element of the normal intersection of $\mathrm{core}(H)$ and $\ker(\phi)$. Then $txt^{-1} \in H$, $t \in T$ and $txt^{-1} \in \ker(\phi)$ for all $t \in G$, hence in particular for all $t \in T$ hence $x \in \ker(\phi \uparrow_T G)$. ∎

In particular the kernel of a permutation representation $\pi = 1_H \uparrow G$ is given by $\mathrm{core}(H)$.

Unitary Representations The following theorem shows that inductions of unitary representations again are unitary.

1.19 Theorem *Let ϕ be a unitary representation of $H \leq G$ and T a transversal of $H \backslash G$. Then $\phi \uparrow_T G$ also is unitary.*

Proof Denote $n = \deg(\phi)$, $m = (G : H)$ and let $x \in G$. Then $A = (\phi \uparrow_T G)(x) = [\dot{\phi}(t_i x t_j^{-1})]$ is block permuted with blocks of size n and $A^* = [\dot{\phi}(t_i x t_j^{-1})]^* = [\dot{\phi}(t_j x t_i^{-1})^*]$

hence also $A \cdot A^*$. We calculate the block $B_{k,\ell}$ of $A \cdot A^*$ at position $(k, \ell) \in \{1, \ldots, m\} \times \{1, \ldots, m\}$ as

$$B_{k,\ell} = \sum_{i=1}^{m} \dot{\phi}(t_k x t_i^{-1}) \cdot \dot{\phi}(t_\ell x t_i^{-1})^*.$$

The sum equals $\mathbf{0}_n$ if $k \neq \ell$ since in this case one of the factors in the sum always equals $\mathbf{0}_n$. For $k = \ell$ there is always j with $t_k x t_j^{-1} \in H$ and

$$B_{k,k} = \sum_{i=1}^{m} \dot{\phi}(t_k x t_i^{-1}) \cdot \dot{\phi}(t_k x t_i^{-1})^* = \phi(t_k x t_j^{-1}) \cdot \phi(t_k x t_j^{-1})^* = \mathbf{1}_n,$$

since ϕ is unitary. ∎

1.3 Monomial Representations

Monomial representations are a natural generalization of permutation representations. A representation $\phi : G \to \mathsf{GL}_n(\mathbb{K})$ of a group G is called monomial if every image $\phi(g),\ g \in G$ is a monomial matrix, i.e. $\phi(g)$ contains in every row and column exactly one entry $\neq 0$.

Where the set of all permutation matrices in $\mathsf{GL}_n(\mathbb{K})$ is finite (of size $n!$) the same does not hold anymore for monomial matrices (if $|\mathbb{K}| = \infty$) not even for the subset of those of finite order.

There are substantially less references dealing with monomial representations resp. groups than with permutation groups though questions concerning the former generally cannot be answered by reduction to the latter. E.g. monomial representations of degree > 1 can be thoroughly irreducible whereas the same is impossible for permutation representations. Furthermore, there is a class of groups (so-called M-groups) with the property that every representation is equivalent to a monomial one.

As references for monomial groups we merely want to mention Ore (1942), [49] and Crouch (1955), [20] both of which does not contribute to the problems dealt with in this dissertation.

In the following we will present the most important constructive results concerning monomial representations. A monomial representation essentially is a direct sum of induction of representations of degree 1 (cf. Theorem 1.22 and Theorem 1.24), hence the results of the last section can be applied. First we want to generalize some notion of permutation representations to monomial representations. For this purpose we associate with every monomial representation μ a unique permutation representation in the following way.

1.20 Definition *Let μ be a monomial representation. Substituting all entries $\neq 0$ by 1 in the images of μ leads to a permutation representation which will be denoted by $\hat{\mu}$.*

Using $\hat{\mu}$ allows to transfer many conceptions for permutation representations to monomial representations. We want to mention the following references for permutation groups: The beautiful book of Wielandt (1964), [60] provides a very good introduction, some aspects are better presented in the book of Passman (1968), [50]. The book of Dixon/Mortimer (1996), [24] is the most recent standard book on the topic and contains among other things the Theorem of O'Nan/Scott classifying the primitive permutation groups.

1.21 Definition *A monomial representation μ of a group G is called transitive, primitive, n-fold transitive, if the same holds for $\hat{\mu}$. The orbits of μ on $\{1, \dots, \deg(\mu)\}$ are defined as the orbits of $\hat{\mu}$ on this set. The stabilizer* $\mathrm{stab}_\mu(i)$ *of a point i under μ is the stabilizer of i under $\hat{\mu}$.*

Orbit Decomposition Like permutation representations also monomial representations can be decomposed by a permutation into its transitive constituents according to their orbits. To achieve further decomposition it is hence possible to restrict to the transitive case which will be investigated in the next paragraphs.

1.22 Theorem *Let μ be a monomial representation of degree n of a group G. Assume the orbits of μ on $\{1, \dots, n\}$ are given as the lists $O_1, \dots, O_k$. Suppose further σ is the permutation mapping $L = (\ell_1, \dots, \ell_n) = O_1 \cup \dots \cup O_k$ onto $(1, \dots, n)$, i.e. $\ell_i^\sigma = i,\ i = 1 \dots n$. Then*

$$\mu^{[\sigma,n]} = \bigoplus_{i=1}^{k} \mu_i,$$

where μ_i are transitive monomial representations.

Proof trivial. ∎

The algorithm for orbit decomposition hence reads as follows.

1.23 Algorithm (Orbit Decomposition) Given is a monomial representation μ of degree n of a group G. μ shall be decomposed into its transitive constituents.

1. Determine the orbits $O_1, \dots, O_k$ of μ on $\{1, \dots, n\}$.
2. Concatenate the orbits to obtain the list $L = \bigcup_{i=1}^{k} O_i = (\ell_1, \dots, \ell_n)$ and determine the permutation $\sigma \in \mathsf{S}_n$ with $\ell_i^\sigma = i,\ i = 1 \dots n$.

3. Conjugate μ by $[\sigma, n]$ and decompose into a direct sum of transitive representations $\mu_1, \dots, \mu_k$ of degrees $|O_1|, \dots, |O_k|$.

We obtain $\mu^{[\sigma,n]} = \bigoplus_{i=1}^{k} \mu_i$, where the μ_i are transitive. ■

Decomposition into an Induction Every monomial representation is equivalent to an induction of a representation λ of degree 1 of a subgroup H. This will now be proven constructively.

1.24 Theorem *Let μ be a transitive monomial representation of a group G with representation space $V = \langle v_1, \dots, v_n \rangle$, i.e.*

$$v_i \cdot \mu(g) = v_{i^{\hat{\mu}(g)}} \cdot a_i(g).$$

Then there is a diagonal matrix D, a subgroup $H \leq G$ with representation λ of degree 1 and a transversal T such that

$$\mu^D = \lambda \uparrow_T G.$$

Proof Assume $H = \mathrm{stab}_\mu(1)$ denotes the stabilizer of 1 under μ. Since μ is transitive we have $(G : H) = \deg(\mu) = n$. Let $T = (t_1, \dots, t_n)$ be a transversal of $H \backslash G$ with $1^{\hat{\mu}(t_i)} = i$. For $h \in H$ we observe $v_1 \mu(h) = v_1 a_1(h)$ and define by

$$\lambda : \; h \mapsto a_1(h)$$

a representation λ of H of degree 1. The representation space of the induced representation then is given by $V^G = \langle v_1 \otimes t_i \mid i = 1 \dots n \rangle$. Setting $t_i g = h_i t_{i'}, \; h_i \in H$ we deduce

$$\begin{aligned} (v_1 \otimes t_i)(\lambda \uparrow_T G)(g) &= v_1 \lambda(h_i) \otimes t_{i'} \\ &= (v_1 \otimes t_{i'}) a_1(h_i). \end{aligned}$$

We define $w_i = v_1 \mu(t_i) = v_i a_1(t_i)$ for $i = 1 \dots n$ and calculate

$$\begin{aligned} w_i \cdot \mu(g) &= v_1 \cdot \mu(t_i g) \\ &= v_1 \cdot \mu(h_i t_{i'}) \\ &= v_1 \cdot a_1(h_i) \mu(t_{i'}) \\ &= w_{i'} \cdot a_1(h_i). \end{aligned}$$

Hence the change of bases $v_i \to a_1(t_i)v_i,\ i = 1 \dots n$, describes the transformation of μ to $\lambda \uparrow_T G$. The corresponding matrix is

$$D = \mathrm{diag}(a_1(t_i)^{-1} \mid i = 1 \dots n),$$

since G operates from the *right*. ∎

The case of a permutation representation deserves special attention: here μ is even equal to an induction.

1.25 Corollary If under the conditions of Theorem 1.24 μ is even a permutation representation and $H = \mathrm{stab}_\mu(1)$ then

$$\mu = 1_H \uparrow_T G$$

for every transversal $T = (t_1, \dots, t_n)$ of $H\backslash G$ with the property $1^{\mu(t_i)} = i,\ i = 1 \dots n$.

Proof Since all entries in the images of μ are 1 we obtain 1_H as the representation from which μ is induced. The correction matrix D hence degenerates to the identity. ∎

According to the proof of Theorem 1.24 we obtain the following algorithm for the decomposition of a monomial representation into a conjugated induction.

1.26 Algorithm (Decomposition into an Induction) Let μ be an arbitrary transitive monomial representation of a group G. μ shall be decomposed into a conjugated induction of a representation of a subgroup of degree 1.

1. Compute the stabilizer $H = \mathrm{stab}_\mu(1)$ of 1 under μ.
2. Calculate a list $T = (t_1, \dots, t_n)$ with $t_i \in G$ and $1^{\hat{\mu}(t_i)} = i$.
3. Define the representation $\lambda,\ (\deg(\lambda) = 1)$ of H by

   $$\lambda:\ h \mapsto \mu(h)_{1,1},$$

   where $\mu(h)_{1,1}$ denotes the upper left entry in the matrix $\mu(h)$.
4. Compute $a_1(t_i)$ for $i = 1 \dots n$ where $a_1(t_i)$ is defined as the (unique) entry $\neq 0$ in the first row of $\mu(t_i)$.

We obtain

$$\mu^D = \lambda \uparrow_T G,\ \text{with } D = \mathrm{diag}(a_1(t_i)^{-1} \mid i = 1 \dots n).$$

∎

1.27 Example The following monomial representation of the symmetric group $\mathsf{S}_3 = \langle x, y \mid x^3 = y^2 = 1,\ x^y = x^{-1}\rangle$ shall be decomposed into a conjugated induction using Algorithm 1.26:

$$\mu:\ x \mapsto \begin{bmatrix} 0 & 1/2 & 0 \\ 0 & 0 & 1 \\ 2 & 0 & 0 \end{bmatrix},\ y \mapsto \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix},$$

with corresponding permutation representation

$$\hat{\mu}:\ x \mapsto [(1,2,3),3],\ y \mapsto [(2,3),3].$$

1. $H = \mathrm{stab}_\mu(1) = \mathrm{stab}_{\hat{\mu}}(1) = \langle y \rangle$.
2. $T = (1, x, x^2)$.
3. $\lambda:\ y \mapsto \mu(y)_{1,1} = -1$.
4. $a_1(1) = 1,\ a_1(x) = 1/2,\ a_1(x^2) = 1/2$
   (Entries $\neq 0$ in the first row of $\mu(1),\ \mu(x),\ \mu(x^2)$).

The result is

$$\mu^{\mathrm{diag}(1,2,2)} = \lambda \uparrow_T \mathsf{S}_3.$$

■

On the uniqueness of this decomposition we prove the following theorem.

1.28 Theorem *Let λ_i be a representation of $H_i \leq G$ of degree 1 and T_i a transversal of $H_i\backslash G$ for $i = 1, 2$. Suppose*

$$\lambda_1 \uparrow_{T_1} G = \lambda_2 \uparrow_{T_2} G,$$

then H_1 and H_2 are conjugated subgroups and $\lambda_1,\ \lambda_2$ are inner conjugated representations of G.

Proof Let $\mu = \lambda_1 \uparrow_{T_1} G = \lambda_2 \uparrow_{T_2} G$. First we show that H_1 is the stabilizer of a point under μ. Assume $t \in T_1$ is the (unique) element $\in H_1$ at position i. Then

$$x \in \mathrm{stab}_\mu(i) \Leftrightarrow txt^{-1} \in H_1 \Leftrightarrow x \in H_1$$

hence $H_1 = \mathrm{stab}_\mu(i)$. Analogous, also H_2 is the stabilizer of a point under μ. Since μ is transitive it follows that $H_1 = H_2^s$ for a certain $s \in G$. Using Theorem 1.10 we get $\mu = \lambda_2 \uparrow_{T_2} G = \lambda_2^s \uparrow_{s^{-1}T_2} G$ where λ_2^s is a representation of H_1. Again, we consider the transversal elements $u \in T_1$ and $v \in s^{-1}T_2$ which both are in H_1, w.l.o.g. at the common position j. Then for $x \in H_1$

$$\lambda_1(uxu^{-1}) = \lambda_2^s(vxv^{-1}) \Leftrightarrow \lambda_1(x) = \lambda_2^s(x)$$

hence λ_1 and λ_2 are inner conjugates. ∎

An extension of Theorem 1.28 will be proven later (Theorem 1.53). In order to compute efficiently with monomial representations the first step is to decompose them into a conjugated induction. In this form they can easier be handled, e.g. the results from the previous section allow a number of manipulations.

Decomposition into an Outer Tensor Product In this paragraph we will prove a necessary and sufficient criterion which determines whether a monomial representation decomposes into a conjugated, outer tensor product. The criterion has been found by Minkwitz and will be presented here with a shorter proof.

1.29 Theorem *Let μ be a transitive monomial representation of a group G and let $G = N_1 \times N_2$ be the direct product of N_1 and N_2. The stabilizer* $\mathrm{stab}_\mu(1)$ *of 1 under μ shall be denoted S. Then μ is equivalent to an outer tensor product of two representations μ_1 of N_1 and μ_2 of N_2,*

$$\mu \cong \mu_1 \# \mu_2,$$

if and only if

$$|S| = |S \cap N_1| \cdot |S \cap N_2|.$$

Assume $\mu^D = \lambda \uparrow_T G$ with a representation λ of degree 1 of S (cf. Theorem 1.24) then

$$\mu^{DM} = ((\lambda \downarrow S \cap N_1) \uparrow_{T_1} N_1) \# ((\lambda \downarrow S \cap N_2) \uparrow_{T_2} N_2),$$

where T_i is a transversal of $(S \cap N_i)\backslash N_i$ for $i = 1, 2$ and M is a monomial matrix corresponding to the change of transversals $T \to T_1T_2$ (cf. Theorem 1.1).

Proof Suppose $\mu \cong \mu_1 \# \mu_2$ (necessarily μ_1 and μ_2 are monomial) and $S = \mathrm{stab}_\mu(1)$. In this situation we have $S_1 = \mathrm{stab}_{\mu_1}(1) = S \cap N_1$ (since $1^{\mu(x)} = 1$ and $x \in N_1 \Leftrightarrow 1^{\mu_1(x)} = 1$), analogous $S_2 = \mathrm{stab}_{\mu_2}(1) = S \cap N_2$. Comparing the degrees yields

$$\begin{aligned} \deg(\mu) &= \deg(\mu_1) \cdot \deg(\mu_2) \\ \Leftrightarrow \quad |G|/|S| &= |N_1|/|S \cap N_1| \cdot |N_2|/|S \cap N_2| \\ \Leftrightarrow \quad |S| &= |S \cap N_1| \cdot |S \cap N_2|, \end{aligned}$$

as desired.

Assume now $S_i = S \cap N_i,\ i = 1,2$ and $|S| = |S_1| \cdot |S_2|$. Because of $S_i \trianglelefteq S,\ i = 1,2$ and $S_1 \cap S_2 = \{1\}$ we get $S = S_1 \times S_2$ and hence

$$\begin{aligned} \lambda \uparrow_T G &= ((\lambda \downarrow S_1) \# (\lambda \downarrow S_2)) \uparrow_T G \\ &\cong ((\lambda \downarrow S_1) \uparrow_{T_1} N_1) \# ((\lambda \downarrow S_2) \uparrow_{T_2} N_2), \end{aligned}$$

where the first equality holds since λ is an outer tensor product (because it is irreducible, cf. e.g. [25], p. 54). The second equivalence holds because of Theorem 1.7. If M denotes the conjugating matrix corresponding to the change of transversals $T \to T_1 T_2$ then

$$\begin{aligned} \mu^{DM} &= \lambda \uparrow_{T_1 T_2} G \\ &= ((\lambda \downarrow S_1) \uparrow_{T_1} N_1) \# ((\lambda \downarrow S_2) \uparrow_{T_2} N_2) \end{aligned}$$

according to Theorem 1.7. ∎

Before we proceed we will give an example of a permutation representation of a direct product which does *not* decompose into an outer tensor product.

1.30 Example We consider the group $\mathsf{S}_3 = \langle x, y \mid x^3 = y^2 = 1, x^y = 1 \rangle$. A faithful, transitive permutation representation ϕ of $\mathsf{S}_3 \times \mathsf{S}_3$ is given by

$$\begin{aligned} \phi: \ & (x,1) \mapsto [(1,5,4)(2,3,6),6],\ (y,1) \mapsto [(1,2)(3,4)(5,6),6], \\ & (1,x) \mapsto [(1,4,5)(2,3,6),6],\ (1,y) \mapsto [(1,2)(3,5)(4,6),6]. \end{aligned}$$

If ϕ decomposed into an outer tensor product then one of the factors would be of degree < 2. Since the factors must be faithful representations of S_3 this is impossible. ∎

1.31 Corollary If ϕ is a regular representation or a representation of degree 1 then ϕ decomposes into an outer tensor product exactly as the group decomposes into a direct product.

Proof ϕ regular: In this case we have $|S| = 1$ and for arbitrary N_1, N_2 with $G = N_1 \times N_2$ the condition $|S| = |S \cap N_1| \cdot |S \cap N_2|$ is satisfied.
$\deg(\phi) = 1$: In this case it is $S = G$ and with $G = N_1 \times N_2$ the condition from Theorem 1.29 is satisfied. ∎

Note that the previous theorem cannot easily be generalized to arbitrary inductions since we use the fact that a representation of G of degree 1 is irreducible. If furthermore G is a direct product then this representation is *equal* to an outer tensor product. An approach to decompose a representation using the special structure of the images is investigated in Egner/Püschel/Beth (1997), [27] but is inferior to the method presented above.

The proof of the preceding theorem also provides the algorithm for the decomposition into a conjugated outer tensor product.

1.32 Algorithm (Decomposition into an Outer Tensor Product)
Given is a transitive monomial representation μ of G. μ shall be decomposed into a conjugated outer tensor product.

1. Decompose $\mu^D = \lambda \uparrow_T G$ with a representation λ of degree 1 of a subgroup $S \leq G$ (Algorithm 1.26).

2. Determine a decomposition $G = N_1 \times N_2$ of G with $|S| = |S \cap N_1| \cdot |S \cap N_2|$. For this purpose we compute the set of normal subgroups. If there is no such decomposition then μ does not decompose into an outer tensor product.

3. Determine transversals T_i of $(S \cap N_i)\backslash N_i,\ i = 1, 2$ and the matrix M corresponding to the change of transversals $T \to T_1 T_2$.

We obtain
$$\mu^{DM} = ((\lambda \downarrow S \cap N_1) \uparrow_{T_1} N_1) \# ((\lambda \downarrow S \cap N_2) \uparrow_{T_2} N_2)\,.$$

∎

Abelian Groups In this paragraph we will classify the transitive monomial representations of abelian groups. For this purpose we will first recall the relationship between the representations of a group G and those of the factor group G/N.

1.33 Lemma Let $N \trianglelefteq G$ be a normal subgroup. Then the representations of G/N correspond bijectively to those representations of G for which N is contained in the kernel.

Proof Let $\kappa:\ G \to G/N,\ g \mapsto gN$ denote the canonical homomorphism. Assume ϕ is a representation of G/N, then the composition $\phi \circ \kappa$ is a representation of G containing N in the kernel. ($\phi \circ \kappa$ is a homomorphism and $x \in N \Rightarrow (\phi \circ \kappa)(x) = \phi(1 \cdot N) = \mathbf{1}_{\deg(\phi)}$).

If vice-versa ϕ is a representation of G satisfying $N \leq \ker(\phi)$ then $gN \mapsto \phi(g)$ is a (well-defined) representation of G/N. ∎

Sometimes we will identify representations of G/N with the corresponding representation of G.

The representation theory of abelian groups is very simple since all irreducibles have degree 1. This implies that any representation ϕ of a subgroup $H \leq G$ has an extension $\overline{\phi}$ to G. In Algorithm 1.74 we will show how this can be done constructively.

Now we are ready to classify monomial representations of abelian groups.

1.34 Theorem *Let μ be a transitive monomial representation of an abelian group G with decomposition $\mu^D = \lambda \uparrow_T G$ according to Theorem 1.24, where λ is a representation of $N \leq G$ with extension $\overline{\lambda}$ to G. Then*

$$\mu^{DD_1} = \overline{\lambda} \cdot (1_N \uparrow_T G) \textit{ with } D_1 = \mathrm{diag}(\overline{\lambda}(t) \mid t \in T) \textit{ and } \overline{\lambda} \downarrow N = \lambda$$

In particular μ is equivalent (by a diagonal matrix) to the product of a representation of degree 1 and a regular representation of a factor group of G. Thus the irreducible components of μ are pairwise different.

Proof λ can be extended to a representation $\overline{\lambda}$ of G. Using Theorem 1.15 we get

$$\mu^{DD_1} = (\lambda \uparrow_T G)^{D_1} = \left((\overline{\lambda} \downarrow N) \uparrow_T G\right)^{D_1} = (1_N \uparrow_T G) \otimes \overline{\lambda} = \overline{\lambda} \cdot (1_N \uparrow_T G).$$

A regular representation of an abelian group contains pairwise different irreducibles. ∎

According to Corollary 1.31 a regular representation decomposes into an outer tensor product in the same way as the group decomposes into a direct product. Since an abelian group can be decomposed into a direct product using a composition series, this can be done very efficient.

Prime Degree In this paragraph we want to characterize the transitive monomial representations μ of prime degree $\deg(\mu) = p$. Therefore we need the following theorem about permutation groups of prime degree which already has been found by Burnside (cf. [24], p. 91 and 99).

1.35 Theorem (Burnside) *Every transitive permutation group of prime degree p is either two-fold transitive or solvable with a regular, cyclic p-Sylow normal subgroup.*

1.36 Theorem *Let μ be a transitive monomial representation of G of prime degree* $\deg(\mu) =$ *p. Assume μ is not two-fold transitive and has the decomposition $\mu^D = \lambda_H \uparrow_T G$ with a representation λ of $H \leq G$ of degree 1. Then exactly one of the two following cases applies:*

i) μ is irreducible

ii) λ_H has an extension λ_G to G, i.e.

$$\mu^{D \cdot D_1} = \lambda_G \cdot (1_H \uparrow_T G), \text{ with } D_1 = \mathrm{diag}(\lambda_G(t) \mid t \in T).$$

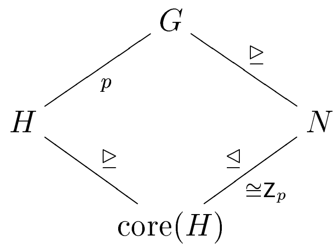

Figure 1.5: Situation in Theorem 1.36

Proof $1_H \uparrow G$ is a faithful permutation representation of $G/\operatorname{core}(H)$ (cf. Theorem 1.18) of degree p. According to the Theorem of Burnside (1.35) exists a normal subgroup N of G with

$$\operatorname{core}(H) \overset{p}{\trianglelefteq} N \trianglelefteq G$$

hence $(N : \operatorname{core}(H)) = p$ (cf. Figure 1.5). Suppose T_1 is a transversal of $\operatorname{core}(H)\backslash N$ and use Corollary 1.14 to get

$$(\lambda_H \uparrow_{T_1} G) \downarrow N = (\lambda_H \downarrow \operatorname{core}(H)) \uparrow_{T_1} N$$

and $(\lambda_H \uparrow_T G) = (\lambda_H \uparrow_{T_1} G)^M$ with a monomial matrix M. Let $\lambda_{\operatorname{core}(H)} = \lambda_H \downarrow \operatorname{core}(H)$. Using the Theorem of Clifford 1.71 (on page 60) exactly one of the following two cases applies:

1. $\lambda_{\operatorname{core}(H)}$ has an extension λ_N to N. In this case also λ_H has an extension λ_G to G (Write $g \in G$ as $g = ht,\ h \in H,\ t \in T_1$ and define $\lambda_G(ht) = \lambda_H(h)\lambda_N(t)$). By Theorem 1.15 we get

$$(\lambda_H \uparrow_T G)^{D_1} = \lambda_G \cdot (1_H \uparrow_T G), \text{ with } D_1 = \mathrm{diag}(\lambda_G(t) \mid t \in T).$$

2. $\lambda_{\operatorname{core}(H)} \uparrow_{T_1} N$ is irreducible. Then also $\lambda_H \uparrow_{T_1} G$ as an extension is irreducible and hence μ. ∎

M-Groups There is a class of groups with the property that every representation is equivalent to a monomial one. In this paragraph we will briefly put together two most important results delimiting this class of groups. We follow Huppert, [36], pp. 578.

1.37 Definition *A group G is called M-group if every representation of G is equivalent to a monomial representation.*

Obviously, a group G is an M-group if and only if every *irreducible* representation is equivalent to an M-group.

The class of M-groups is delimited by the following two theorems.

1.38 Theorem *i) Every super-solvable group is an M-group.*

ii) Every nilpotent group is an M-group.

iii) Every solvable group having only abelian Sylow groups is an M-group.

1.39 Theorem *Every M-group is solvable.*

For a solvable group of order n to contain a non-abelian Sylow group it is necessary that n contains one prime factor with exponent 3, to avoid that the group is nilpotent n must contain at least two different prime factors. The smallest number satisfying both conditions is obviously 24. We will give an example of a group of order 24 which is not an M-group.

1.40 Example We consider the group $\mathsf{SL}(2,3)$ of (2×2)-matrices over the field $\mathbb{F}_3$. If Q_8 denotes the quaternion group then

$$\mathsf{SL}(2,3) \cong \mathsf{Z}_3 \ltimes \mathsf{Q}_8 \cong \langle r, s, t \mid r^3 = s^4 = t^4 = 1,\ s^t = s^{-1},\ t^r = st,\ s^r = t^{-1} \rangle.$$

The representation ρ given by

$$\rho:\ r \mapsto \frac{1}{2} \cdot \begin{bmatrix} -1+\mathrm{i} & -1-\mathrm{i} \\ 1-\mathrm{i} & -1-\mathrm{i} \end{bmatrix},\ t \mapsto \begin{bmatrix} 0 & \mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix},\ s \mapsto \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

is faithful, irreducible and not equivalent to a monomial representation. ∎

1.4 The Extension Formula of Minkwitz

The Extension Theorem of Minkwitz (cf. [47]) allows to constructively extend an irreducible representation of a subgroup to the entire group if this is possible. Necessary therefore is merely the character of the extension.

1.41 Theorem (Minkwitz' Extension Formula) *Let $H \leq G$ be a subgroup and ρ an irreducible representation of H of degree $\deg(\rho) = n$ with character χ. Assume there is a character $\overline{\chi}$ of G extending χ to G, i.e. $\overline{\chi} \downarrow H = \chi$. Then*

$$\overline{\rho}: \; g \mapsto \frac{n}{|H|} \cdot \sum_{h \in H} \overline{\chi}(gh^{-1})\rho(h)$$

defines a representation $\overline{\rho}$ extending ρ to G and has the character $\chi_{\overline{\rho}} = \overline{\chi}$.

The shortest proof is due to Michael Clausen and can be found in [15].

1.5 Intertwining Space

The term intertwining space has already been introduced in the Introduction and shall be defined now.

1.42 Definition *Assume ϕ, ψ are representations of the group G over the field $\mathbb{K}$ with degrees $\deg(\phi) = n,\ \deg(\psi) = m$ respectively. Then we call the vector space*

$$\mathsf{Int}(\phi, \psi) = \{A \in \mathbb{K}^{n \times m} \mid \forall g \in G: \;\; \phi(g) \cdot A = A \cdot \psi(g)\}$$

the "intertwining space" of ϕ and ψ. Further we will denote by

$$\langle \phi, \psi \rangle = \dim(\mathsf{Int}(\phi, \psi))$$

the dimension of the intertwining space or "intertwining number" of ϕ and ψ.

For exploring the mathematical structure of representations only the intertwining number is needed. It is invariant under conjugation of the arguments and hence depends only on the equivalence class of the representations. We will show below that the intertwining number is nothing but the scalar product of the corresponding characters like the notion already suggests.

However, as before the constructivity aimed for in this chapter requires more. We not only want to know how the intertwining*number* behaves under conjugation, direct sum, induction etc., we also would like to know how the intertwining*space* transforms. Thus, as in Section 1.2, we will refine known theorems.

This section is divided into two parts: in the first we collect known theorems dealing with intertwining numbers, in the second we will refine them to concrete statements about intertwining spaces.

Intertwining Number The notion $\langle \phi, \psi \rangle$ as scalar product follows Clausen/Baum, [16] and is justified through the following theorem.

1.43 Theorem *Let ϕ, ψ be representation of the group G. Then the following holds:*

i) If ϕ and ψ are irreducible then

$$\langle \phi, \psi \rangle = \begin{cases} 1, & \phi \cong \psi \\ 0, & \phi \not\cong \psi \end{cases} .$$

ii) $\langle \phi, \psi \rangle = \langle \psi, \phi \rangle$.

iii) $\langle \phi_1 \oplus \phi_2, \psi \rangle = \langle \phi_1, \psi \rangle + \langle \phi_2, \psi \rangle$.

iv) $\langle \phi, \phi \rangle = 1 \Leftrightarrow \phi$ *is irreducible.*

v) $\langle \phi, \psi \rangle = \langle \chi_\phi, \chi_\psi \rangle$.

For the proof we refer to [21], p. 320/321.

The last theorem shows that the dimension of the intertwining space is nothing but the well-known scalar product of the corresponding characters

$$\langle \chi_\phi, \chi_\psi \rangle = \frac{1}{|G|} \sum_{g \in G} \chi_\phi(g) \chi_\psi(g^{-1})$$

which provides a way to compute the intertwining number. Note that the intertwining number is invariant under conjugation of the arguments.

Two important theorems allow the computation with the intertwining number of inductions.

1.44 Theorem (Frobenius reciprocity) *Let $H \leq G$ be a subgroup, ϕ a representation of H and ψ a representation of G. Then*

$$\langle \phi \uparrow G, \psi \rangle = \langle \phi, \psi \downarrow H \rangle.$$

The proof can be found, e.g. in [25], pp. 48.

1.45 Theorem (Intertwining Number Theorem) *Assume $H_1, H_2 \leq G$ are two subgroups with representations ϕ_1, ϕ_2. For $x, y \in G$ the intertwining number*

$$\langle \phi_1^x \downarrow (H_1^x \cap H_2^y), \phi_2^y \downarrow (H_1^x \cap H_2^y) \rangle$$

depends only on the double coset $H_1 \cdot xy^{-1} \cdot H_2$. If S denotes a system of representatives of the double cosets $H_1 \backslash G / H_2$ then

$$\langle \phi_1 \uparrow G, \phi_2 \uparrow G \rangle = \sum_{s \in S} \langle \phi_1^s \downarrow (H_1^s \cap H_2), \phi_2 \downarrow (H_1^s \cap H_2) \rangle.$$

For the proof see [21], p. 327.

Using the Intertwining Number Theorem we obtain some simple formulas for permutation representations.

1.46 Corollary Let π_1, π_2 be two transitive permutation representations of the group G with $H_i = \mathrm{stab}_{\pi_i}(1),\ i = 1, 2$. Then

$$\langle \pi_1, \pi_2 \rangle = |H_1 \backslash G / H_2|.$$

Proof Assume S is a system of representatives of the double cosets $H_1 \backslash G / H_2$. Using Corollary 1.25 we have $\pi_i = 1_{H_i} \uparrow G,\ i = 1, 2$ and hence

$$\begin{aligned} \langle \pi_1, \pi_2 \rangle &= \langle 1_{H_1} \uparrow G, 1_{H_2} \uparrow G \rangle \\ &= \sum_{s \in S} \langle 1_{H_1 \cap H_2^s}, 1_{H_1 \cap H_2^s} \rangle \\ &= |H_1 \backslash G / H_2| \end{aligned}$$

by Theorem 1.45 as desired. ∎

1.47 Corollary Let π be a transitive permutation representation of a group G.Then the following holds:

i) $\langle \pi, 1_G \rangle = 1$.

ii) π is irreducible $\Leftrightarrow \deg(\pi) = 1$.

iii) π is two-fold transitive $\Leftrightarrow$ $\langle \pi, \pi \rangle = 2$, i.e. in this case $\pi \cong 1_G \oplus \rho$ with an irreducible representation ρ.

Proof Let $\pi = 1_H \uparrow G$. Because of $|H \backslash G / G| = 1$ and Corollary 1.46 follows i), ii) follows from i). If π is two-fold transitive then $|H \backslash G / H| = 2$ yielding iii) with Corollary 1.46. ∎

Intertwining Space We begin with some easy statements which can be viewed as a refinement of Theorem 1.43.

1.48 Theorem *Let $\phi, \phi_1, \phi_2 \ldots, \psi, \psi_1, \psi_2 \ldots$ be representations over $\mathbb{K}$ of the group G with degrees $\deg(\phi) = n, \deg(\psi) = m$. Then the following holds:*

i) For $A \in \mathsf{GL}_n(\mathbb{K}), B \in GL_m(\mathbb{K})$:

$$\mathsf{Int}(\phi^A, \psi^B) = A^{-1} \cdot \mathsf{Int}(\phi, \psi) \cdot B.$$

ii)

$$\mathsf{Int}(\phi_1 \oplus \phi_2, \psi_1 \oplus \psi_2) = \left\{ \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \mid A_{i,j} \in \mathsf{Int}(\phi_i, \psi_j) \right\}.$$

iii) (Schur's Lemma) If ϕ, ψ are irreducible of degree n, then

$$\mathsf{Int}(\phi, \psi) = \begin{cases} \mathbb{K} \cdot A, \text{ for an } A \in \mathsf{GL}_n(\mathbb{K}), & \phi \cong \psi \\ \mathbf{0}_n, & \phi \not\cong \psi \end{cases}.$$

iv) Assume $\phi = (\mathbf{1}_{n_1} \otimes \phi_1) \oplus \ldots \oplus (\mathbf{1}_{n_k} \otimes \phi_k)$ and $\psi = (\mathbf{1}_{m_1} \otimes \phi_1) \oplus \ldots \oplus (\mathbf{1}_{m_k} \otimes \phi_k)$ are two entirely decomposed representations with irreducible, pairwise different ϕ_i, then

$$\mathsf{Int}(\phi, \psi) = (\mathbb{K}^{n_1 \times m_1} \otimes \mathbf{1}_{\deg(\phi_1)}) \oplus \ldots \oplus (\mathbb{K}^{n_k \times m_k} \otimes \mathbf{1}_{\deg(\phi_k)}).$$

Hence every matrix $\in \mathsf{Int}(\phi, \psi)$ is block permuted according to the homogeneous components of ϕ and ψ.

Proof i) and ii) is straightforward. For iii) let ϕ, ψ be irreducible of degree n. We use Theorem 1.43 i). If $\phi \not\cong \psi$ then $\langle \phi, \psi \rangle = 0$, and hence $\mathsf{Int}(\phi, \psi) = \{\mathbf{0}_n\}$. If $\phi \cong \psi$ then exists an invertible matrix A satisfying $\phi^A = \psi$ which generates the intertwining space because of $\langle \phi, \psi \rangle = 1$. iv) follows from ii) and iii). ■

1.49 Example We consider the group $\mathsf{Z}_4 = \langle x \mid x^4 = 1 \rangle$ with representations of degree 1: $\lambda_k : \ x \mapsto \omega_4^k, \ k = 0 \ldots 3$, and the regular representation $\phi : \ x \mapsto [(1, 2, 3, 4), 4]$. Then

$$\mathsf{Int}(\lambda_k, \lambda_\ell) = \langle \delta_{k,\ell} \rangle, \ \mathsf{Int}(\lambda_1, \phi) = \langle [1, -\omega_4, -1, \omega_4] \rangle$$

and

$$\mathsf{Int}(\phi, \phi) = \langle \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix}, \begin{bmatrix} 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \\ 0&1&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&0&1 \\ 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \end{bmatrix} \rangle$$

is the vector space of circulant matrices. If we set $\rho = \lambda_0 \oplus \lambda_1 \oplus \lambda_2 \oplus \lambda_3$ then

$$\mathsf{Int}(\phi, \rho) = \langle \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \omega_4 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & -\omega_4 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_4 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & \omega_4 \end{bmatrix} \rangle$$

is the space also containing DFT_4 as the sum of the base elements. ∎

For our purpose the intertwining spaces of monomial representations are of particular interest. By Theorem 1.48 i) and ii) this is equivalent to the question of intertwining spaces of inductions of one-dimensional representations. We will provide the answer by proving constructive variants of the Frobenius reciprocity and of the Intertwining Number Theorem.

1.50 Theorem (Frobenius reciprocity) *Let $H \leq G$ be a subgroup of index n with transversal $T = (t_1, \ldots, t_n)$, let ϕ be a representation of G and ψ be a representation of H. Assume $t_i^{-1} = h_i t_{i'}$, $i = 1 \ldots n$. Then*

$$\Phi : \begin{cases} \mathsf{Int}(\phi \downarrow H, \psi) & \rightarrow & \mathsf{Int}(\phi, \psi \uparrow_T G) \\ A & \mapsto & [A_1, \ldots, A_n] \end{cases}, \text{ with } A_{i'} = \phi(t_i) \cdot A \cdot \psi(h_i)$$

is an isomorphism of vector spaces.

Proof We start with a preliminary remark. Suppose ϕ_1, ϕ_2 are representations (over $\mathbb{K}$) of degrees n_1, n_2 of the same group G. Then G operates on the vector space $\mathbb{K}^{n_1 \times n_2}$ via

$$g \bullet M = \phi_1(g) \cdot M \cdot \phi_2(g^{-1}), \; g \in G, \; M \in \mathbb{K}^{n_1 \times n_2}$$

from the left. Hence a matrix M is in the intertwining space $\mathsf{Int}(\phi_1, \phi_2)$ if and only if M is invariant under G via $\bullet$.

Now we are ready to prove Theorem 1.50. Let $d = \deg(\phi), e = \deg(\psi)$ and assume t_j is the (unique) transversal element $\in H$. Corresponding to a matrix $A \in \mathsf{Int}(\phi \downarrow H, \psi) \leq \mathbb{K}^{d \times e}$ we define the matrix $\overline{A} \in \mathbb{K}^{d \times en}$ by

$$\overline{A} = [\mathbf{0}_{d \times e}, \ldots, \mathbf{0}_{d \times e}, \underset{j}{A \cdot \psi(t_j^{-1})}, \mathbf{0}_{d \times e}, \ldots, \mathbf{0}_{d \times e}].$$

We show first that $\overline{A} \in \mathsf{Int}(\phi \downarrow H, (\psi \uparrow_T G) \downarrow H)$. Let $h \in H$. To compute $\overline{A} \cdot (\psi \uparrow_T G)(h)$ it is sufficient to know the jth block row of $(\psi \uparrow_T G)(h)$. This block row contains in the jth block column the matrix $\psi(t_j h t_j^{-1})$ and the all-zero matrix else. We calculate $A \cdot \psi(t_j^{-1}) \cdot \psi(t_j h t_j^{-1}) = \phi(h) \cdot A \cdot \psi(t_j^{-1})$ as desired. In other words $\overline{A}$ is invariant under H via $\bullet$. In order to construct from $\overline{A}$ an invariant under G, i.e. an element $B \in \mathsf{Int}(\phi, \psi \uparrow_T G)$, we use the Reynolds operator (cf. e.g.[59]), i.e. we build the sum of the images of $\overline{A}$ under the coset representatives:

$$B = \sum_{i=1}^{n} t_i \bullet \overline{A} = \sum_{i=1}^{n} \phi(t_i) \cdot \overline{A} \cdot (\psi \uparrow_T G)(t_i^{-1}).$$

To compute the product $\overline{A} \cdot (\psi \uparrow_T G)(t_i^{-1})$ again it is sufficient to know the jth block row of $(\psi \uparrow_T G)(t_i^{-1})$. We use $t_i^{-1} = h_i t_{i'}$ and get

$$t_j t_i^{-1} t_\ell^{-1} \in H \Leftrightarrow t_j h_i t_{i'} t_\ell^{-1} \in H \Leftrightarrow \ell = i'$$

and $\psi(t_j t_i^{-1} t_{i'}^{-1}) = \psi(t_j h_i)$. Further

$$\overline{A} \cdot (\psi \uparrow_T G)(t_i^{-1}) = [\mathbf{0}_{d\times e}, \ldots, \mathbf{0}_{d\times e}, \underset{i'}{A \cdot \psi(h_i)}, \mathbf{0}_{d\times e}, \ldots, \mathbf{0}_{d\times e}]$$

which implies that $B = [A_1, \ldots, A_n]$ where $A_{i'} = \phi(t_i) \cdot A \cdot \psi(h_i)$. The mapping Φ obtained by this construction is an isomorphism of vector spaces as can confirmed easily. ■

The following theorem is a constructive refinement of the Intertwining Number Theorem, 1.45.

1.51 Theorem *Let $H, K \le G$ be subgroups of index n, m resp. with transversals $T = (t_1, \ldots, t_n)$ and $S = (s_1, \ldots, s_m)$ and representations ϕ resp. ψ with degrees $d = \deg(\phi)$ and $e = \deg(\psi)$. Then the following holds:*

i) $H \cdot t_i s_j^{-1} \cdot K = H \cdot t_k s_\ell^{-1} \cdot K \Leftrightarrow \exists g \in G:\ H t_i g = H t_k$ *and* $K s_j g = K s_\ell$.

ii) The mapping

$$\Phi_{k,\ell} : \begin{cases} \mathsf{Int}(\phi^{t_k} \downarrow (H^{t_k} \cap K^{s_\ell}), \psi^{s_\ell} \downarrow (H^{t_k} \cap K^{s_\ell})) & \to \quad \mathsf{Int}(\phi \uparrow_T G, \psi \uparrow_S G) \\ A & \mapsto \quad [A_{i,j}]_{1\le i\le n,\ 1\le j\le m} \end{cases}$$

where

$$A_{i,j} = \begin{cases} \phi(t_i g t_k^{-1}) \cdot A \cdot \psi(s_\ell g^{-1} s_j^{-1}), & H \cdot t_i s_j^{-1} \cdot K = H \cdot t_k s_\ell^{-1} \cdot K, \\ & g \text{ as defined in i)} \\ \mathbf{0}_{d\times e}, & \text{else} \end{cases}$$

defines an injective homomorphism of vector spaces.

iii) *the image of $\Phi_{k,\ell}$ only depends on the double coset $D = H \cdot t_k s_\ell^{-1} \cdot K$ and will be denoted by V_D. It is*

$$\mathsf{Int}(\phi \uparrow_T G, \psi \uparrow_S G) = \bigoplus_{D \in H\backslash G/K} V_D \quad \textit{(direct sum of vector spaces).}$$

Proof First we show i):

$$\begin{aligned} H \cdot t_i s_j^{-1} \cdot K = H \cdot t_k s_\ell^{-1} \cdot K \quad &\Leftrightarrow \quad H \cap t_i s_j^{-1} \cdot K \cdot (t_k s_\ell^{-1})^{-1} \neq \emptyset \\ &\Leftrightarrow \quad t_i^{-1} H t_k \cap s_j^{-1} K s_\ell \neq \emptyset \\ &\Leftrightarrow \quad \exists g \in G: \ t_i g t_k^{-1} \in H \text{ and } s_j g s_\ell^{-1} \in K \\ &\Leftrightarrow \quad \exists g \in G: \ H t_i g = H t_k \text{ and } K s_j g = K s_\ell. \end{aligned}$$

To prove ii) we define corresponding to $A \in \mathsf{Int}(\phi^{t_k} \downarrow (H^{t_k} \cap K^{s_\ell}), \psi^{s_\ell} \downarrow (H^{t_k} \cap K^{s_\ell}))$ the matrix $B = [B_{i,j}]$ by $B_{k,\ell} = A,\ B_{i,j} = \mathbf{0}_{d\times e}$ else. Then $(\phi \uparrow_T G)(h) \cdot B = B \cdot (\psi \uparrow_S G)(h)$ for all $h \in (H^{t_k} \cap K^{s_\ell})$. In other words, B is invariant under $(H^{t_k} \cap K^{s_\ell})$ via the operation

$$g \bullet M = (\phi \uparrow_T G)(g) \cdot M \cdot (\psi \uparrow_S G)(g^{-1}), \ g \in G, \ M \in \mathbb{K}^{dn\times em}.$$

Similar to the proof of Theorem 1.50 we obtain an invariant under G by building the sum over a system of representatives of the cosets $(H^{t_k} \cap K^{s_\ell})\backslash G$:

$$C = \sum_{r \in R} r \bullet B.$$

The cosets correspond bijectively to the pairs (i, j) with the property $H \cdot t_i s_j^{-1} \cdot K = H \cdot t_k s_\ell^{-1} \cdot K$ since G operates on the pairs (Ht_i, Ks_j) of cosets via multiplication from the right: $(Ht_i, Ks_j) \cdot g = (Ht_i g, Ks_j g)$. The fix group of (Ht_i, Ks_j) is given by $H^{t_i} \cap K^{s_j}$ and by i) two pairs of cosets are contained in the same orbit if and only if the quotients of the representatives lie in the same double coset. Thus the matrix C contains a block $\neq \mathbf{0}_{d\times e}$ at position (i, j) if and only if $H \cdot t_i s_j^{-1} \cdot K = H \cdot t_k s_\ell^{-1} \cdot K$ (and $A \neq \mathbf{0}_{d\times e}$). Assume this condition is satisfied for a pair (i, j) and choose $r \in G$ such that $Ht_i r = Ht_k$ and $Ks_j r = Ks_\ell$. To compute the block $A_{i,j}$ of C at position (i, j) it is sufficient to consider the kth block row of $(\phi \uparrow_T G)(r)$ and the ℓth block column of $(\psi \uparrow_S G)(r^{-1})$. Since $t_i r t_k^{-1} \in H$ and $s_\ell r^{-1} s_j^{-1} \in K$ the matrix $A_{i,j}$ has the desired form.

Actually we have $A_{i,j} \in \mathsf{Int}(\phi^{t_i} \downarrow (H^{t_i} \cap K^{s_j}), \psi^{s_j} \downarrow (H^{t_i} \cap K^{s_j}))$ because: if C is an arbitrary invariant under G via $\bullet$, then, in particular, for $h \in (H^{t_i} \cap K^{s_j})$ the condition $\phi(t_i h t_i^{-1}) \cdot A_{i,j} \cdot \psi(s_j h^{-1} s_j^{-1})$ is satisfied by the block $A_{i,j}$. Assume that C is an invariant under G via $\bullet$ and let the block at a position (k,ℓ) be given. Then all blocks at positions (i,j) satisfying $H\cdot t_i s_j^{-1}\cdot K = H\cdot t_k s_\ell^{-1}\cdot K$ are uniquely determined. This implies that the image of the mapping $\Phi_{k,\ell}$ only depends on the double coset $H\cdot t_k s_\ell^{-1}\cdot K$. It can easily be seen that $\Phi_{k,\ell}$ defines an injective homomorphism of vector spaces. Obviously the sum of the images

$$\bigoplus_{D\in H\backslash G/K} V_D \leq \mathsf{Int}(\phi\uparrow_T G, \psi\uparrow_S G)$$

is direct, because: assume $\Phi_{k,\ell}$, $\Phi_{k',\ell'}$ are two homomorphisms corresponding two different double cosets with image matrices A_1, A_2 resp., then there is no position where A_1 *and* A_2 contain an entry $\neq 0$. Comparison of the dimensions (Intertwining Number Theorem, 1.45) establishes equality. ∎

This very technical theorem allows to deduce some interesting conclusions for monomial representations.

Assume $\mu_1 = \lambda_H\uparrow_T G$ and $\mu_2 = \lambda_K\uparrow_S G$ are two transitive monomial representations with $\deg(\lambda_H) = \deg(\lambda_K) = 1$. Let $T = (t_1,\ldots,t_n)$ and $S = (s_1,\ldots,s_m)$. Using Theorem 1.51 we can compute a base of $\mathsf{Int}(\mu_1,\mu_2)$. For this purpose we choose a system of representatives of the double cosets $H\backslash G/K$ from the quotients $t_i s_j^{-1}$, $i = 1\ldots n$, $j = 1\ldots m$. Such a system already can be chosen among the quotients $t_1 s_j^{-1}$, $j = 1\ldots m$ (use Theorem 1.51, i), with $g = t_1 t_i^{-1}$) for example $t_1 s_{j_1}^{-1},\ldots,t_1 s_{j_d}^{-1}$, $d = |H\backslash G/K|$. For every quotient $t_1 s_{j_r}^{-1}$ there are exactly two possibilities:

1. $\langle \lambda_H^{t_1}\downarrow (H^{t_1}\cap K^{s_{j_r}}), \lambda_K^{s_{j_r}}\downarrow (H^{t_1}\cap K^{s_{j_r}})\rangle = 0$. In this case the double coset $H\cdot t_1 s_{j_r}^{-1}\cdot K$ contributes nothing to the intertwining space $\mathsf{Int}(\mu_1,\mu_2)$.
2. $\langle \lambda_H^{t_1}\downarrow (H^{t_1}\cap K^{s_{j_r}}), \lambda_K^{s_{j_r}}\downarrow (H^{t_1}\cap K^{s_{j_r}})\rangle = 1$, i.e. the corresponding intertwining space is generated by the (1×1)-Matrix $[1]$. The corresponding base vector $\Phi_{1,j_r}(1)$ of $\mathsf{Int}(\mu_1,\mu_2)$ hence is given by

$$\begin{aligned}\Phi_{1,j_r}(1) &= [a_{k,\ell}\mid k\in\{1,\ldots,n\},\ \ell\in\{1,\ldots,m\}],\ \text{where}\\ a_{k,\ell} &= \begin{cases}\lambda_H(t_k g t_1^{-1})\cdot\lambda_K(s_{j_r} g^{-1} s_\ell^{-1}), & Ht_i g = Ht_1 \text{ and}\\ & Ks_\ell g = Ks_{j_r}\\ 0, & \text{else}\end{cases}.\end{aligned}$$

In this way we obtain a base $B = \{B_1,\ldots,B_e\}$ of $\mathsf{Int}(\mu_1,\mu_2)$ of length $e\leq d$. Suppose $\mu_1' = \mu^{D_1}$ and $\mu_2' = \mu^{D_2}$ with diagonal matrices D_1, D_2 (which is the general case of transitive monomial matrices) then by Theorem 1.48, i), $B = \{D_1^{-1}B_1D_2,\ldots,D_1^{-1}B_eD_2\}$ is a base of $\mathsf{Int}(\mu_1',\mu_2')$. B has the following properties.

1.52 Lemma Let μ_1, μ_2 be two transitive monomial representations of the group G with decompositions $\mu_1 = (\lambda_H \uparrow_T G)^{D_1}$ and $\mu_2 = (\lambda_K \uparrow_S G)^{D_2}$. Let further $T = (t_1, \ldots, t_n)$ and $S = (s_1, \ldots, s_m)$. D_1, D_2 are diagonal matrices. The base $B = \{B_1, \ldots, B_e\}$ of $\mathsf{Int}(\mu_1, \mu_2)$ as constructed above has the following properties:

i) For $i \neq j$ there is no position at which B_i *and* B_j contain an entry $\neq 0$.

ii) The base B is by property i) uniquely determined up to scalar multiplication of its elements.

iii) If the matrix B_i contains an entry $\neq 0$ at position (k, ℓ) then it contains exactly $(H^{t_k} : (H^{t_k} \cap K^{s_\ell}))$ entries $\neq 0$ in every row and $(K^{s_\ell} : (H^{t_k} \cap K^{s_\ell}))$ entries $\neq 0$ in every column.

Proof Property i) follows from the construction. Any base can be expressed by linear combinations from B which implies ii). To prove iii) we calculate:

$$\begin{aligned} & B_i \text{ has entries} \neq 0 \text{ at positions } (k,\ell),\ (k,\ell') \\ \Leftrightarrow\quad & H \cdot t_k s_\ell^{-1} \cdot K = H \cdot t_k s_{\ell'}^{-1} \cdot K \\ \Leftrightarrow\quad & H \cdot t_{k'} \underbrace{t_{k'}^{-1} t_k s_\ell^{-1} K}_{= s_{\ell_1} K} = H \cdot t_{k'} \underbrace{t_{k'}^{-1} t_k s_{\ell'}^{-1} K}_{= s_{\ell'_1} K} \\ \Leftrightarrow\quad & B_i \text{ has entries} \neq 0 \text{ at positions } (k',\ell_1), (k',\ell'_1). \end{aligned}$$

Multiplication from left with $t_{k'}^{-1} t_k$ permutes the cosets G/K hence B_i contains in every row the same number of entries $\neq 0$. Analogously we deduce that B_i contains in every column the same number of entries $\neq 0$. Assume B_i has an entry $\neq 0$ at position (k, ℓ) then B_i contains $(G : (H^{t_k} \cap K^{s_\ell}))$ entries $\neq 0$ altogether (follows from the proof of Theorem 1.45). Since B_i is a matrix with $n = (G : H) = (G : H^{t_k})$ rows and $m = (G : K) = (G : K^{s_\ell})$ columns we obtain the formulas in iii). ∎

Assume $\mu_1 = 1_H \uparrow G$ and $\mu_2 = 1_K \uparrow G$ are two transitive permutation representation. Using Lemma 1.52 we can reveal a connection to design theory. In this case the base B is of length $e = d = |H \backslash G / K|$ and the entries $\neq 0$ in the base vectors all are $= 1$ which determines B uniquely. The sum of the base vectors is the all-one matrix. Lemma 1.52, iii) says that every base vector describes a design. For the theory of designs refer to the standard books of Beth/Jungnickel/Lenz (1985), [9], Buekenhout (1995), [11] and Colbourn (1996), [17].

Understanding the special structure of the base B is crucial for the proof of the following theorem.

1.53 Theorem *Let $\mu_1 = (\lambda_H \uparrow_T G)^{D_1} \cong \mu_2 = (\lambda_K \uparrow_S G)^{D_2}$ be two equivalent monomial representations. A monomial matrix M with $\mu_1^M = \mu_2$ exists if and only if H and K are conjugated in G and λ_H and λ_K are inner conjugates in G.*

Proof Assume H and K are conjugated in G and $\lambda_K = \lambda_H^g$. Then

$$\mu_2 = (\lambda_K \uparrow_S G)^{D_2} = (\lambda_H^g \uparrow_S G)^{D_2} = (\lambda_H \uparrow_{gS} G)^{D_2}$$

by Theorem 1.10. The transition from $\mu_2^{D_2^{-1}}$ to $\mu_1^{D_1^{-1}}$ hence can be achieved by a change of transversal which implies the result by Theorem 1.1.

For the reverse direction we assume that the intertwining space $\mathsf{Int}(\mu_1, \mu_2)$ contains a monomial matrix. By Lemma 1.52 already the base B (as constructed above) contains a monomial matrix. Counting the entries $\neq 0$ by Lemma 1.52, iii), yield $(H^{t_k} : (H^{t_k} \cap K^{s_\ell})) = (K^{s_\ell} : (H^{t_k} \cap K^{s_\ell})) = 1$ hence $H^{t_k} = K^{s_\ell}$ and H and K are conjugated in G. Furthermore $\lambda^{t_k} \downarrow (H^{t_k} \cap K^{s_\ell}) = \lambda^{s_\ell} \downarrow (H^{t_k} \cap K^{s_\ell})$ which implies that λ_H and λ_K are inner conjugate as required. ∎

For permutation representation the following corollary is immediate.

1.54 Corollary Let $\pi_1 = (1_H \uparrow_T G) \cong \pi_2 = (1_K \uparrow_S G)$ be two equivalent permutation representations. Then there is a permutation matrix P with $\pi_1^P = \pi_2$ if and only if H and K are conjugated in G.

The following is an example for two equivalent permutation representations which cannot be conjugated onto eachother using a permutation matrix.

1.55 Example we consider the simple group $\mathsf{PSL}(3,2) = \langle x, y \mid x^2 = y^3 = (xy)^7 = 1\rangle$ of size 168 with the two following two-fold transitive permutation representations.

$$\pi_1 : \ x \mapsto [(4,5)(6,7), 7], \ y \mapsto [(1,2,4)(3,6,5), 7],$$

and

$$\pi_2 : \ x \mapsto [(1,2)(3,5), 7], \ y \mapsto [(1,4,7)(2,3,6), 7].$$

We have $\pi_1 \cong \pi_2$, $\langle \pi_1, \pi_2 \rangle = 2$ and $\mathsf{Int}(\pi_1, \pi_2) = \langle M_1, M_2 \rangle$ with

$$M_1 = \begin{bmatrix} 1&1&1&0&1&0&0 \\ 0&0&1&1&1&1&0 \\ 1&1&0&1&0&1&0 \\ 0&1&0&0&1&1&1 \\ 1&0&1&0&0&1&1 \\ 0&1&1&1&0&0&1 \\ 1&0&0&1&1&0&1 \end{bmatrix}, \ M_2 = \begin{bmatrix} 0&0&0&1&0&1&1 \\ 1&1&0&0&0&0&1 \\ 0&0&1&0&1&0&1 \\ 1&0&1&1&0&0&0 \\ 0&1&0&1&1&0&0 \\ 1&0&0&0&1&1&0 \\ 0&1&1&0&0&1&0 \end{bmatrix}.$$

M_1 and M_2 describe two complementary 2-designs corresponding two the smallest projective geometry, the so-called *Fano plane*. Obviously there is no permutation matrix in $\mathsf{Int}(\pi_1, \pi_2)$. ∎

Computing the intertwining space is expensive in the general case, however, plays an important role in constructive representation theory. It allows, e.g. to determine a conjugating matrix for two arbitrary, equivalent representations. The computation requires the solution of a system of linear equations.

Assume ϕ, ψ are representations of the same group $G = \langle g_1, \dots, g_n \rangle$. Obviously, a matrix $A = [a_{i,j}]$ lies in $\mathsf{Int}(\phi, \psi) \subset \mathbb{K}^{\deg(\phi) \times \deg(\psi)}$ if and only if the equations

$$\phi(g_i) \cdot A = A \cdot \psi(g_i) \Leftrightarrow \phi(g_i) \cdot A - A \cdot \psi(g_i) = \mathbf{0}_{\deg(\phi), \deg(\psi)}, \; i = 1 \dots n,$$

are satisfied. Thus we obtain for every generator g the following $\deg(\phi) \cdot \deg(\psi)$ equations in the same number of unknowns:

$$\sum_{i=1}^{\deg(\phi)} \phi(g)_{k,i} \cdot a_{i,\ell} - \sum_{j=1}^{\deg(\psi)} a_{k,j} \cdot \psi(g)_{j,\ell}, \; k = 1 \dots \deg(\phi), \; \ell = 1 \dots \deg(\psi).$$

We put this down in the following theorem.

1.56 Theorem *The intertwining space of two representations ϕ, ψ of the group G generated by $\{g_1, \dots, g_n\}$ can be computed by solving a system of $n \cdot \deg(\phi) \cdot \deg(\psi)$ linear equations in $\deg(\phi) \cdot \deg(\psi)$ unknowns.*

If the representations ϕ, ψ are monomial then the system of equations is sparse.

1.6 Decomposition Matrices

Let ϕ be a representation of the group G. We will refer to a decomposition matrix of ϕ as any matrix A decomposing ϕ into a direct sum of irreducible representations, i.e.

$$\phi^A = (x \mapsto A^{-1} \cdot \phi(x) \cdot A) = \bigoplus_{i=1}^{n} \rho_i, \; \rho_i \text{ irreducible for } i = 1 \dots n.$$

Decomposition matrices play virtually no role in books on representation theory, their mere existence is sufficient for the purposes there. In this dissertation, however, they are of central importance. On the one side they are necessary in order to decompose constructively a given representation: a function decomposing a monomial representation ϕ shall return not only the ρ_i but a representation which is equal to ϕ only having a different structure, namely $(\rho_1 \oplus \ldots \oplus \rho_n)^{A^{-1}}$. On the other side, decomposition matrices are the most important building block for the decomposition of matrices with symmetry (cf. Introduction).

In this section we will develop, up to a certain extent, a theory of decomposition matrices for monomial representations. Since monomial representations essentially are direct sums of inductions (cf. Theorem 1.22 and 1.24) the induction again will play the central role. Among other things a theorem will be presented allowing to construct from a decomposition matrix of a representation ϕ of a normal subgroup of prime index a decomposition matrix for the induction $\phi \uparrow G$. Using this it will be possible, e.g. to decompose regular representations of solvable groups in the sense above. The algorithm in Chapter 2 essentially uses the results of this section.

Again we want to emphasize that we exclusively consider representations satisfying the Maschke condition. The term decomposition matrix as defined in the beginning of this section can be formulated in the notion of intertwining spaces.

1.57 Definition *Let ϕ be a representation of a group G and ρ an arbitrary decomposition of ϕ into irreducibles ρ_i, i.e. $\phi \cong \rho = \bigoplus_{i=1}^{n} \rho_i$. We will call every invertible matrix in $A \in \mathsf{Int}(\phi, \rho)$ a "decomposition matrix" for ϕ. We write*

$$\phi \xrightarrow[\text{dec}]{A} \rho.$$

If ϕ is even a regular representation of G then we will call $A = \mathrm{DFT}_G$ a "discrete Fourier transform" for G.

The definition of the Fourier transform is a bit sloppy since the regular representation of a group is only is determined up to conjugation with a permutation (the transversal consists of elements of G in any order). Hence it would be better to call it DFT_ϕ. We will assume that always a concrete regular representation is given. The connection with the well-known DFT_n defined by

$$\mathrm{DFT}_n = [\omega_n^{ij} \mid i,j \in \{0,\ldots,n-1\}],\ \omega_n \text{ primitive } n\text{th root of unity},$$

is given by the fact that the latter is a decomposition matrix for a particular regular representation ϕ of $\mathsf{Z}_n = \langle x \mid x^n = 1\rangle$, namely

$$\phi = 1_{\mathsf{E}} \uparrow_T \mathsf{Z}_n,\ T = (x^0, x^1, \ldots, x^{n-1}),\ \mathsf{E} = \langle 1 \rangle$$

which is the same as

$$\phi:\ x \mapsto [(1,\ldots,n), n]$$

and we have the well-known equation

$$\phi^{\mathrm{DFT}_n} = \bigoplus_{i=0}^{n-1} (x \mapsto \omega_n^i).$$

An arbitrary regular representation given by a conjugate of ϕ is not decomposed by DFT_n.

The DFT is the most important building block for the decomposition of monomial representations of solvable groups. It turns out that the DFT_p, p prime, plays a similar role in the decomposition of a regular representation of G as the group Z_p plays in building up G.

Reduction to Permutation Representations One could get the idea that the problem of decomposing monomial representation can be reduced to the problem of decomposing permutation representations. This is not the case which is already indicated by the fact that monomial representations of arbitrary degree may be irreducible. Theorem 1.15 shows a particular case where it is possible.

1.58 Theorem *Let μ be a transitive monomial representation of G with decomposition $\mu^D = \lambda \uparrow_T G$, λ is a representation of H of degree 1. Assume λ has an extension $\overline{\lambda}$ to G. Then*

$$\mu^{D \cdot D_1} = \overline{\lambda} \cdot (1_H \uparrow_T G), \;\; D_1 = \mathrm{diag}(\overline{\lambda}(t) \mid t \in T).$$

If A is a decomposition matrix for the permutation representation $1_H \uparrow_T G$ then $D \cdot D_1 \cdot A$ is a decomposition matrix for μ.

Proof Follows from Theorem 1.15. ∎

Theorems on the decomposition of permutation representation which will be presented in the following always can be (slightly) extended by Theorem 1.58.

Constructions and Decomposition Matrices It would be desirable to derive a number of theorems which provide for every construction of representation (induction, extension, tensor product, etc.) a corresponding formula for the decomposition matrices. However, this is only possible in some cases which will be investigated in the following.

1.59 Theorem *Let ϕ_1, ϕ_2 be representations of G with decomposition matrices A_1, A_2. Then $A_1 \oplus A_2$ is a decomposition matrix for $\phi_1 \oplus \phi_2$.*

Proof trivial. ∎

1.60 Theorem *Let ϕ_1, ϕ_2 be representations of N_1 resp. N_2 with decomposition matrices A_1 and A_2 respectively. Then $A_1 \otimes A_2$ is a decomposition matrix for $\phi_1 \# \phi_2$.*

Proof Follows from the distributivity of "#" and the fact that the outer tensor product of two irreducible representations again is irreducible. ∎

Note that the corresponding statement does not hold for the inner tensor product. If ϕ, ψ are irreducible representations then $\phi \otimes \psi$ is in general not irreducible. The decomposition of $\phi \otimes \psi$ is also known as the *Clebsch-Gordan-Problem.* See, e.g. the book of Fulton/Harris (1991), [30].

The relation between the decomposition matrix of a representation and the decomposition matrix of a restriction is revealed in the following theorem from Minkwitz, [45].

1.61 Theorem *Let $H \leq G$ be a subgroup, ϕ_G a representation of G and $\phi_H = \phi_G \downarrow H$. Then it is possible to derive a decomposition matrix A_G for ϕ_G from a decomposition matrix A_H for ϕ_H by*

$$A_G = A_H \cdot C,$$

where C denotes a block permuted matrix. The block sizes correspond to the sizes of the homogeneous components of ϕ_H. The blocks corresponding to irreducible homogeneous components equal the identity.

$$\begin{tikzcd}
G \arrow[d, dash] & \phi_G \arrow[rrr, "A'_G", "\mathrm{dec}"'] \arrow[d, "\mathrm{res}"'] & & & \rho_G \arrow[d, "\mathrm{res}"'] \\
H & \phi_H \arrow[r, "A_H", "\mathrm{dec}"'] & \rho_H \arrow[r, "C"] & \sigma_H & (\rho_G \downarrow H) \arrow[l, "B"', "\mathrm{dec}"]
\end{tikzcd}$$

Figure 1.6: Situation in Theorem 1.61

Proof Assume A'_G is a decomposition matrix for ϕ_G, i.e. $\phi_G^{A'_G} = \rho_G$. Then $\rho_G \downarrow H = \phi_H^{A'_G}$ is decomposed by a matrix B, i.e. $\phi_H^{A'_G \cdot B} = \sigma_H$ where B only decomposes the irreducible components of ρ_G which turned reducible by restriction (cf. Figure 1.6). In particular $A_G = A'_G \cdot B$ is a decomposition matrix for ϕ_G too. If now A_H is a decomposition matrix for ϕ_H then $C = A_H^{-1} \cdot A_G \in \mathsf{Int}(\rho_H, \sigma_H)$ is block permuted according to the homogeneous components of ρ_H. ∎

A particular nice case occurs if the decomposition matrix of the restriction ϕ_H already decomposes ϕ_G.

1.62 Corollary Assume the same situation as in Theorem 1.61. If the irreducible components of ϕ_H are pairwise inequivalent then for every decomposition matrix A_H of ϕ_H exists a permutation matrix P such that $A_H \cdot P$ is a decomposition matrix for ϕ_G.

Proof In this case all homogeneous components of ϕ_H contain exactly one irreducible and using Theorem 1.61 we can choose $P = C$ as a permutation matrix. ∎

The situation in Corollary 1.62 in particular occurs if ϕ is a monomial representation of G and G contains an abelian normal subgroup which is transitively represented by ϕ (cf. 1.34), i.e. $\phi \downarrow H$ is transitive. In this case also ϕ consists of pairwise inequivalent irreducible components. The permutation matrix P in Corollary 1.62 conjugates those irreducible components in a row which are contained in the same irreducible component of ϕ.

We will use this observation to decompose representations with regular abelian normal subgroups. It turns out that in particular primitive permutation representations of solvable groups fall in this category.

Regular Abelian Normal Subgroup If N is a normal subgroup of G then G operates on the irreducible representations of N (up to equivalence) via inner conjugation. The stabilizer of an irreducible representation under this operation is a group between N and G and plays an important role in the relation between the irreducible representations of N and those of G. The Theorem of Clifford in its most general form (cf. e.g. [22], pp. 259) investigates this relationship. We will need only some aspects therefrom and will state this theorem only for the case that the index $(G : N)$ is prime. Recall that the kernel of a permutation representation $\pi = 1_H \uparrow_T G$ is given by $\mathrm{core}(H) = \bigcap_{x \in G} H^x$ (cf. Theorem 1.18).

1.63 Definition *Let $N \trianglelefteq G$ be a normal subgroup and ϕ a representation of N. Then we will call*

$$G_\phi = \{x \in G \mid \phi^x \cong \phi\}$$

the "inertia group" of ϕ. It is

$$N \leq G_\phi \leq G.$$

1.64 Lemma Let G be a permutation group with regular normal subgroup N and let H be the stabilizer of 1 under G. Then

$$G = H \ltimes N.$$

Proof Let $x \in G$ and $1^x = i$. Since N is transitive it exists $y \in N$ with $1^y = i$ and hence $xy^{-1} = h \in H$ which implies $x \in HN$. Since N is regular we get $H \cap N = \langle 1 \rangle$. ∎

1.65 Lemma Let $G = H \ltimes N$, where N is abelian with irreducible representation λ. Then the following holds:

i) $G_\lambda = (G_\lambda \cap H) \ltimes N$.

ii) λ has an extension $\overline{\lambda}$ to G_λ.

Proof i) is evident. To prove ii) define $\overline{\lambda}(hn) = \lambda(n)$ for $h \in G_\lambda \cap H,\ n \in N$. We show that this defines a homomorphism.

$$\begin{aligned}
\overline{\lambda}(hn)\overline{\lambda}(h_1 n_1) &= \lambda(n)\lambda(n_1) \\
\overline{\lambda}(hnh_1 n_1) &= \overline{\lambda}(hh_1 n^{h_1} n_1) \\
&= \lambda(n^{h_1} n_1) \\
&= \lambda(n^{h_1})\lambda(n_1) \\
&= \lambda(n)\lambda(n_1).
\end{aligned}$$

The last equality holds because λ is of degree 1. ∎

Now we are ready to prove the theorem allowing to decompose permutation representations with regular abelian normal subgroup. In particular all primitive permutation representations of solvable groups fall in this category.

1.66 Theorem *Let $\pi = 1_H \uparrow_T G$ be a permutation representation of degree n of G with kernel* core(H). *Assume $N \trianglelefteq G$ is a normal subgroup such that $N/$*core(H) *is abelian and regularly represented by π (cf. Figure 1.7). Let A be a decomposition matrix of $\pi \downarrow N$, i.e.*

$$(\pi \downarrow N)^A = \bigoplus_{i=1}^{n} \lambda_i,\ \deg(\lambda_i) = 1,\ i = 1 \ldots n,\ \lambda_i \textit{ pairwise different.}$$

G operates on the set $\{\lambda_1, \ldots, \lambda_n\}$ via inner conjugation. Assume $O_1, \ldots, O_k$ are the orbits under this operation, $O_i = \{\lambda_{i,1}, \ldots, \lambda_{i,r_i}\}$ with corresponding conjugations $T_i = \{t_{i,1}, \ldots, t_{i,r_i}\}$, i.e. $\lambda_{i,1}^{t_{i,j}} = \lambda_{i,j},\ j = 1 \ldots r_i$. Further denote by $\overline{\lambda}_{i,1}$ an extension of $\lambda_{i,1}$ to $G_{\lambda_{i,1}},\ i = 1 \ldots k$. Then the following holds:

i) T_i is a transversal of $G_{\lambda_{i,1}} \backslash G$ and $\overline{\lambda}_{i,1} \uparrow_{T_i} G$ is irreducible.

ii) If σ is the permutation mapping $\lambda_1, \ldots, \lambda_n$ onto the concatenation $O_1 \cup \cdots \cup O_k$ then $A \cdot [\sigma, n]$ is a decomposition matrix for π and

$$\pi^{A\cdot[\sigma,n]} = \bigoplus_{i=1}^{k} \overline{\lambda}_{i,1} \uparrow_{T_i} G.$$

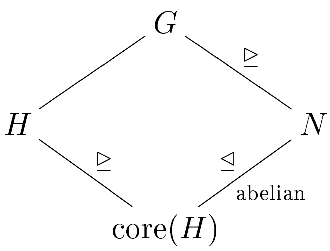

Figure 1.7: Situation in Theorem 1.66

Proof To i): Obviously, T_i is a transversal of $G_{\lambda_{i,1}} \backslash G$. For the proof of $\overline{\lambda}_{i,1} \uparrow_{T_i} G$ being irreducible we refer to [22], p. 265. To ii): We have $G_{\lambda_{i,1}} \backslash G/N = G_{\lambda_{i,1}} N \backslash G = G_{\lambda_{i,1}} \backslash G$ and $G^t_{\lambda_{i,1}} \cap N = N$ for all $t \in T_i$. Using Mackey's subgroup theorem (1.12) we get

$$\begin{aligned}
(\overline{\lambda}_{i,1} \uparrow_{T_i} G) \downarrow N &= \bigoplus_{t \in T_i} (\overline{\lambda}^t_{i,1} \downarrow \underbrace{G^t_{\lambda_{i,1}} \cap N}_{=N}) \uparrow_{(1)} N \\
&= \bigoplus_{t \in T_i} \overline{\lambda}^t_{i,1} \downarrow N \\
&= \bigoplus_{t \in T_i} \lambda^t_{i,1} \\
&= \bigoplus_{j=1}^{r_i} \lambda_{i,j}
\end{aligned}$$

and by Corollary 1.62 we get result. ∎

It is easy to show that $A \cdot P$ is a decomposition matrix for π. The main statement in Theorem 1.66 is the explicit description of the corresponding irreducibles. For algorithmic purposes this of great importance since it allows to forego the (expensive) conjugation $\pi^{A \cdot P}$. Vividly spoken the theorem explains how the one-dimensional representation of N must be collected in order to extend them to G (cf. Figure 1.8) and constructs this extension. In other words: the one-dimensional representations of N "fuse" by extension to G via the operation (inner conjugation) of G.

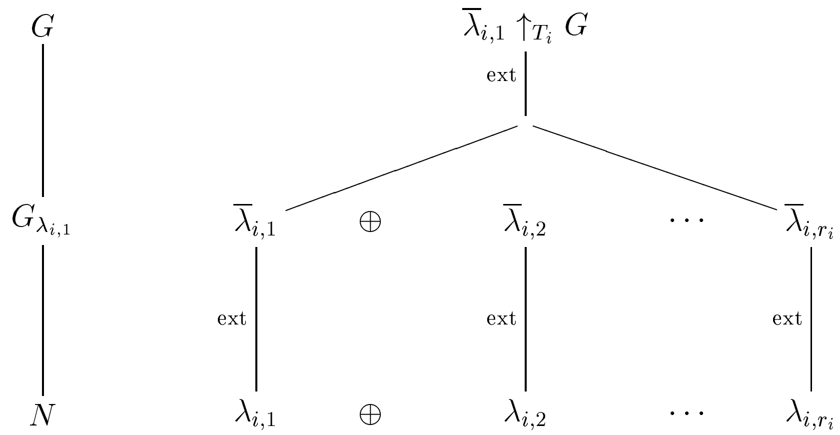

Figure 1.8: Extension of an orbit of one-dimensional representations of N

An algorithmic problem is the computation of the regularly represented normal subgroup N if it exists. In the worst case all normal subgroups of $G/\ker(\pi)$ need to be tested. For a particular class of primitive permutation representations, however, such a normal subgroup is uniquely determined by its size. The special case of a monomial representation of prime degree now can almost be solved completely.

Prime Degree The characterization of the monomial representations μ of prime degree $\deg(\mu) = p$ in Theorem 1.36 provides a class of representations for which Theorem 1.66 can be applied.

1.67 Theorem *Let μ be a transitive monomial representation of G of degree $\deg(\mu) = p$ prime. Assume μ is not two-fold transitive and has the decomposition $\mu^D = \lambda_H \uparrow_T G$ where λ is a representation of $H \le G$ of degree 1. Then exactly one of the two following cases applies:*

i) μ is irreducible.

ii) λ_H has an extension λ_G to G, i.e.

$$\mu^{D \cdot D_1} = \lambda_G \cdot (1_H \uparrow_T G), \;\; D_1 = \mathrm{diag}(\lambda_G(t) \mid t \in T).$$

The p-Sylow group of $G/\operatorname{core}(H)$ is $\cong \mathsf{Z}_p$ and is regularly represented by $1_H \uparrow_T G$. In particular exists a monomial matrix M and a permutation matrix P such that $M \cdot \mathrm{DFT}_p \cdot P$ is a decomposition matrix for μ.

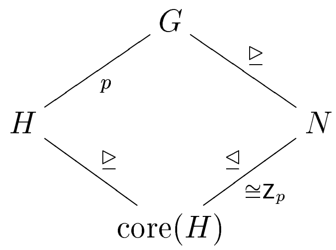

Figure 1.9: Situation in Theorem 1.67

Proof By Theorem 1.35 of Burnside $G/\operatorname{core}(H)$ has a normal p-Sylow subgroup $\cong \mathsf{Z}_p$ which is regularly represented by $1_H \uparrow_T G$. The rest follows from Theorem 1.36 an Theorem 1.66. ∎

The case where μ is two-fold transitive is subject of the next paragraph.

Two-fold Transitive Representations If $\mu \cong \lambda_H \uparrow G$ is a two-fold transitive monomial representation then there are only two double cosets $H \backslash G / H$ with representatives $1, s$. Using the Intertwining Number Theorem (1.45) it follows that

$$\begin{aligned} \langle \lambda_H \uparrow G, \lambda_H \uparrow G \rangle &= \langle \lambda_H, \lambda_H \rangle + \langle \lambda_H^s \downarrow (H^s \cap H), \lambda_H \downarrow (H^s \cap H) \rangle \\ &= 1 + \langle \lambda_H^s \downarrow (H^s \cap H), \lambda_H \downarrow (H^s \cap H) \rangle \end{aligned}$$

and hence μ is either irreducible or contains exactly 2 irreducible components. If λ_H has an extension to G then it is, according to Theorem 1.58 essentially a permutation representation and we are able to decompose.

1.68 Theorem *Let μ be a two-fold transitive monomial representation of degree n of G with decomposition $\mu^D = \lambda_H \uparrow_T G$. Assume λ_G is an extension of λ_H to G. Further we* *define the* $(n \times n)$*-Matrix* SOR_n *(* SOR = *Split OneRep) by*

$$\mathrm{SOR}_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & -1 & 0 & \cdots & 0 \\ 1 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & 0 & 0 & & -1 \end{bmatrix}.$$

Then exists a diagonal matrix A *such that* $A \cdot \mathrm{SOR}_n$ *decomposes* μ.

Proof By Theorem 1.15 we get

$$\mu^{D \cdot D_1} = \lambda_G \cdot (1_H \uparrow_T G), \text{ with } D_1 = \mathrm{diag}(\lambda_G(t) \mid t \in T).$$

$1_H \uparrow_T G$ is double transitive, too, and decomposes according to Theorem 1.47 into $1_G \oplus \rho$ with an irreducible representation ρ. The representation space of the one-representation is nothing but the Eigenspace of the Eigenvalue 1, generated by $v = (1, \ldots, 1)$. Since permutation matrices are orthogonal we get that the representation space of ρ is the orthogonal complement W of $\langle v \rangle$. Obviously the rows $2, \ldots, \deg(\mu)$ of SOR_n generate W which implies that SOR_n is a decomposition matrix for $1_H \uparrow_T G$ and $D \cdot D_1 \cdot \mathrm{SOR}_n$ one for μ. ∎

The proof shows that a two-fold transitive permutation representation $1_H \uparrow_t G$ of degree n is decomposed by any $(n \times n)$-matrix M with the following two properties:

1. The first row of M only contains ones.
2. The rows $2, \ldots, n$ span the orthogonal complement of the first row.

These properties are also, for instance, satisfied by DFT_n. The reason for taking SOR_n is that this matrix contains the least number of entries $\neq 0$ among all satisfying these properties.

Primitive Representations The decomposition of a large class of primitive permutation representations is possible by using the following theorem which already has been found by Galois (cf. [36], p. 159).

1.69 Theorem (Galois) *Let* G *be a primitive permutation group of degree* n*, let* G_1 *be the stabilizer of* 1 *and* N *a minimal normal subgroup of* G*. If* N *is solvable then:*

i) N *is regular and elementary abelian, in particular* n *is a prime power.*

ii) $G = G_1 \ltimes N$.

iii) N *is the only minimal normal subgroup of* G.

Transferred to permutation representations this reads as:

1.70 Theorem *Let* $\pi = 1_H \uparrow_T G$ *be a primitive permutation representation of* G. *Assume* $G/\operatorname{core}(H)$ *has a minimal solvable normal subgroup* $N/\operatorname{core}(H)$. *Then* $\deg(\pi) = (N : \operatorname{core}(H)) = p^n$, *p prime and* $N/\operatorname{core}(H) \cong \mathsf{Z}_p^n$ *is regularly represented by* π *(cf. Figure 1.10).*

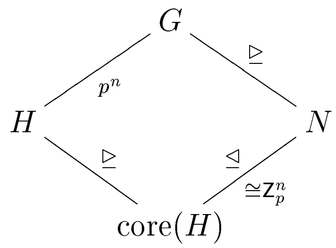

Figure 1.10: Situation in Theorem 1.70

Proof Follows from Theorem 1.69. ∎

Using Theorem 1.66 this means that we are in particular able to decompose primitive permutation representations of solvable groups. Necessarily, the degree is a prime power in this case. The normal subgroup $N \cong \mathsf{Z}_p^n$ can be identified by its size since it is the *unique* minimal normal subgroup according to Theorem 1.69.

A far more powerful tool for the decomposition of a representation is provided by the Theorem of Clifford which we are now going to prove constructively for an important particular case.

A Recursion Formula Transitivity of induction (cf. Theorem 1.4)

$$\phi \uparrow_{TS} G = (\phi_H \uparrow_T K) \uparrow_S G,$$

allows to perform it stepwise, e.g. along a chain of subgroups $G = K_1 \geq K_2 \geq \cdots \geq K_n = H$, where K_i is a maximal subgroup in K_{i-1} for $i = 2 \ldots n$. An obvious idea is to perform the decomposition of a monomial representation likewise along this chain. This requires to answer the following two questions:

1. How do the irreducible components of $\phi_H \uparrow K_i$ arise from those of $\phi_H \uparrow K_{i-1}$?

2. How to compute a decomposition matrix of $\phi_H \uparrow K_i$ from one of $\phi_H \uparrow K_{i-1}$?

Unfortunately the answers to this questions does not exist in general. In the case that the chain above is part of a composition series, however, the Theorem of Clifford provides an exact answer to question 1. An answer to question 2. will be presented in Theorem 1.75. The Theorem of Clifford in its most general form deals (cf. e.g. [21], p. 345) with the restriction of an irreducible representation of a group G to a normal subgroup $N \trianglelefteq G$. With regard to a composition series we will consider the particular case $(G : N) = p$ prime. In the spirit of this chapter the theorem will be presented in a constructively refined form.

1.71 Theorem (Theorem of Clifford) *Let $N \overset{p}{\trianglelefteq} G$ be a normal subgroup of prime index p, $T = (t^0, t^1, \ldots, t^{p-1})$ a transversal of G/N and ρ an irreducible representation of N. Then exactly one of the two following cases applies:*

1. *(cf. Figure 1.11) $\rho \cong \rho^{t^i}$ for $i = 0 \ldots p-1$. Then ρ has exactly p pairwise inequivalent extensions to G. Assume $\overline{\rho}$ is one of these and $\lambda : \; t \mapsto \omega_p$ (ω_p primitive pth root of unity) a representation of G/N, then all extensions are given by $\lambda^i \cdot \overline{\rho}$, $i = 0 \ldots p-1$. The induction decomposes into irreducibles according to*

$$(\rho \uparrow_T G)^A = \bigoplus_{i=0}^{p-1} \lambda^i \cdot \overline{\rho},$$

where

$$A = \mathrm{diag}(\overline{\rho}(t)^i \mid i \in \{0, \ldots, p-1\}) \cdot (\mathrm{DFT}_p \otimes \mathbf{1}_{n/p}).$$

2. *(cf. Figure 1.12) $\rho \not\cong \rho^{t^i}$ for $i = 0 \ldots p-1$. Then the induction $\rho \uparrow G$ is irreducible,*

$$(\rho \uparrow_T G) \downarrow N = \bigoplus_{i=0}^{p-1} \rho^{t^i}$$

and

$$\rho^{t^i} \uparrow_T G = (\rho \uparrow_T G)^B,$$

where

$$B = \left([(1, \ldots, p)^{-i}, p] \otimes \mathbf{1}_{\deg(\rho)}\right) \cdot \left(\mathbf{1}_{(p-i) \cdot \deg(\rho)} \oplus (\mathbf{1}_i \otimes \rho(t^p))\right).$$

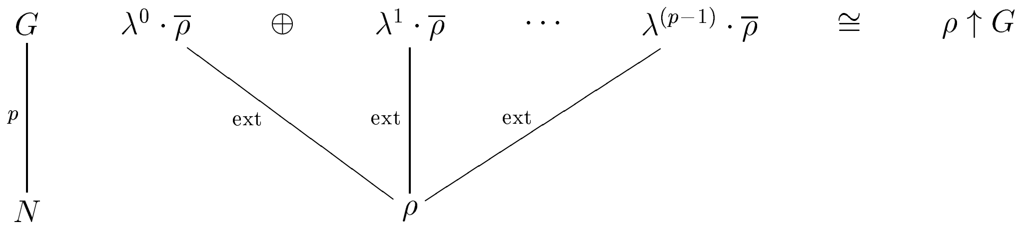

Figure 1.11: Theorem of Clifford, case: $\rho \cong \rho^{t^i}$

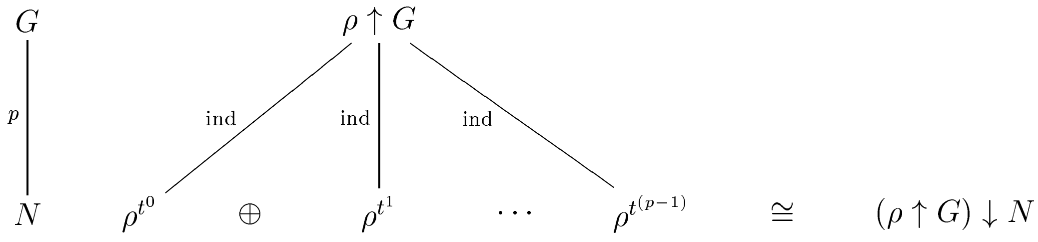

Figure 1.12: Theorem of Clifford, case: $\rho \ncong \rho^{t^i}$

Proof The proof can be found, e.g. in [16], pp. 88. We will prove only the three equations. Equation in 1.: Using Theorem 1.15 leads to

$$(\rho \uparrow_T G)^D = (1_N \uparrow_T G) \otimes \overline{\rho}, \;\; D = \mathrm{diag}(\overline{\rho}(t)^i \mid i \in \{0, \ldots, p-1\}).$$

The representation $1_N \uparrow_T G$ is a regular representation $t \mapsto [(1, \ldots, p), p]$ of $G/N \cong \mathsf{Z}_p$ and is decomposed by DFT_p into p representations $t \mapsto \omega_p^i$, $i = 0 \ldots p-1$ of degree 1 which gives first equation.

First equation in 2.: follows from Corollary 1.13. Second equation in 2.: By Theorem 1.10 we have $\rho^{t^i} \uparrow_T G = \rho \uparrow_{t^i T} G$. Multiplication of T with t^i permutes the cosets as $\sigma = (1, \ldots, p)^{-i}$. The transition from T^σ to $t^i T$,

$$T^\sigma = (t^i, \ldots, t^{(p-1)}, t^0, \ldots, t^{(i-1)}) \to t^i T = (t^i, \ldots, t^{(p-1)}, t^p, \ldots, t^{(p-i+1)}),$$

means multiplication of the last i transversal elements by t^p. Using Theorem 1.1 gives the result. ∎

1.72 Example We consider $G = \mathsf{S}_3 = \langle x, y \mid x^3 = y^2 = 1,\ x^y = x^{-1} \rangle$ with normal subgroup $N = \mathsf{Z}_3 = \langle x \rangle \trianglelefteq G$ of index 2. N has exactly 3 different irreducible representations of degree 1:

$$\lambda_1 = 1_N, \quad \lambda_2 : \; x \mapsto \omega_3, \quad \lambda_3 : \; x \mapsto \omega_3^{-1}.$$

Obviously, λ_1 is equal to its inner conjugates and hence has 2 extensions

$$\mu_1 = 1_G, \quad \mu_2 : \ x \mapsto 1, \ y \mapsto -1.$$

In Example 1.9 we have seen that $\lambda_2^y = \lambda_3 \not\cong \lambda_2$, hence the induction $\mu_3 = \lambda_2 \uparrow G \cong \lambda_3 \uparrow G$ is irreducible:

$$\mu_3 = \lambda_2 \uparrow_{(1,y)} G : \ x \mapsto \begin{bmatrix} \omega_3 & 0 \\ 0 & \omega_3^{-1} \end{bmatrix}, \ y \mapsto \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

It follows that $\mu_1, \ \mu_2, \ \mu_3$ are (up to equivalence) all irreducible representations of G.■

In the case $\rho \cong \rho^t$, ρ has an extension to G. This can for instance be calculated using the Extension Formula of Minkwitz (Theorem 1.41). The formula requires the determination of an extending character and the evaluation of ρ for all $h \in H$. In the special situation in the Theorem of Clifford, however, there is also another method which is used in the proof of this theorem in [16] and based on the following lemma.

1.73 Lemma Assume the situation of Theorem 1.71. In the case $\rho \cong \rho^t$ putting $\rho(t) = A$ defines an extension of ρ to G if and only if:

i) $A \in \mathsf{Int}(\rho^t, \rho)$ and

ii) $A^p = \rho(t^p)$.

If the degree of ρ is small and $|N|$ large then this lemma provides a better possibility to compute the extension as Minkwitz' formula. Another advantage is the fact that no extending character has to be computed. But of course the Minkwitz extension is far more general.

In the situation of Lemma 1.73 A is uniquely determined up to a scalar factor since $\mathsf{Int}(\rho^t, \rho)$ is of dimension 1. Assume arbitrary $A \in \mathsf{Int}(\rho^t, \rho)$, then A^p and $\rho(t^p)$ only differ by a scalar factor α:

$$\alpha \cdot A^p = \rho(t^p).$$

If β is a pth root of α then $t \mapsto \beta \cdot A$ is an extension of ρ.

1.74 Algorithm (Extension without character) Given is an irreducible representation of degree n of $N \overset{p}{\trianglelefteq} G$. Let t generate the factor group G/N and $\rho \cong \rho^t$. An extension $\overline{\rho}$ of ρ to G shall be computed.

1. Compute an arbitrary matrix $\mathbf{0}_n \neq A \in \mathsf{Int}(\rho^t, \rho)$.

2. Compute a pth root β of the number at position $(1,1)$ of the matrix $\rho(t^p) \cdot A^{-p}$.

An extension is given by $t \mapsto \beta \cdot A$. All possible extension are given corresponding to the p different roots of α. ∎

If in particular $\deg(\rho) = 1$, then an extension is defined by putting $\rho(t) = \sqrt[p]{\rho(t^p)}$.

The Theorem of Clifford shows how the irreducible representations of G arise from those of N. Accordingly the following theorem shows how a decomposition matrix of a representation ϕ of N gives rise to a decomposition matrix of $\phi \uparrow G$.

1.75 Theorem *Let $N \overset{p}{\trianglelefteq} G$ be a normal subgroup of prime index p with transversal $T = (t^0, t^1, \ldots, t^{(p-1)})$. Assume ϕ is a representation of N of degree n with decomposition matrix A such that $\phi^A = \bigoplus_{i=1}^k \rho_i$ where $\rho_1, \ldots, \rho_j$ are exactly those among the ρ_i having an extension $\overline{\rho}_i$ to G (Theorem 1.71, case 1). Denote by $d = \deg(\rho_1) + \ldots + \deg(\rho_j)$ the entire degree of the extensible ρ_i and set $\overline{\rho} = \overline{\rho}_1 \oplus \ldots \oplus \overline{\rho}_j$. Then exists a permutation matrix P such that*

$$M = (\mathbf{1}_p \otimes A) \cdot P \cdot \left(\bigoplus_{t \in T} \overline{\rho}(t) \oplus \mathbf{1}_{p(n-d)} \right) \cdot \left((\mathrm{DFT}_p \otimes \mathbf{1}_d) \oplus \mathbf{1}_{p(n-d)} \right)$$

is a decomposition matrix of $\phi \uparrow_T G$. If we denote by $\lambda_i : \ t \mapsto \omega_p^i, \ i = 0 \ldots p-1$, the p one-dimensional representations of G/N then

$$(\phi \uparrow_T G)^M = \bigoplus_{i=0}^{p-1} \bigoplus_{\ell=1}^{j} \lambda_i \cdot \overline{\rho}_\ell \oplus \bigoplus_{i=j+1}^{k} \rho \uparrow_T G$$

is the corresponding decomposition into irreducibles.

Proof By Theorem 1.6 we get

$$(\phi \uparrow_T G)^{(\mathbf{1}_p \otimes A)} = \phi^A \uparrow_T G = \left(\bigoplus_{i=1}^{k} \rho_i \right) \uparrow_T G.$$

We use Theorem 1.5 with the block decomposition $\rho = \rho_1 \oplus \ldots \oplus \rho_j, \rho_{j+1}, \ldots, \rho_k$ to compute a permutation matrix P with

$$(\phi \uparrow_T G)^{(\mathbf{1}_p \otimes A) \cdot P} = \rho \uparrow_T G \oplus \rho_{j+1} \uparrow_T G \oplus \ldots \oplus \rho_k \uparrow_T G.$$

Since ρ has an extension $\overline{\rho}$ to G we get by Theorem 1.15

$$(\rho \uparrow_T G)^{\bigoplus_{t \in T} \overline{\rho}(t)} = (1_N \uparrow_T G) \otimes \overline{\rho}.$$

The representation $(1_N \uparrow_T G)$ is decomposed by DFT_p into $\bigoplus_{i=0}^{p-1} \lambda_i$ where $\lambda_i : \ t \mapsto \omega_p^i$. Thus $\bigoplus_{t\in T} \overline{\rho}(t) \cdot (\mathrm{DFT}_p \otimes \mathbf{1}_d)$ is a decomposition matrix for $\rho \uparrow_T G$ with corresponding decomposition $\bigoplus_{i=0}^{p-1} \bigoplus_{\ell=1}^{j} \lambda_i \cdot \overline{\rho}_\ell$. The inductions of the $\rho_i,\ i = j+1 \ldots k$ are already irreducible which completes the proof. ∎

The condition, that ϕ is decomposed by A such that the extensible irreducibles come first, imposes no restriction since this always can be achieved by a permutation. Further we want to mention that the matrix above decomposes $\phi \uparrow_T G$ but that equivalent irreducibles in this decomposition not necessarily are equal. For algorithmic purposes, however, this is very important. In Chapter 2 we will go further into detail and explain thoroughly how the factors of the decomposition matrix are efficiently computed. Note that using Theorem 1.75 the decomposition matrix as well as the irreducibles can be constructed recursively hence an explicit conjugation is never performed. This is a basic requirement in order to be able to deal with representations of higher degree (> 100) at all.

Using only this theorem it is possible to decompose any monomial representation of a solvable group which arise as an induction of a subnormal subgroup. A subgroup $H \leq G$ of a solvable group G is called subnormal if a composition series exists which contains H.

The decomposition matrix M reminds very much of the well-known Cooley-Tukey decomposition of a DFT_{2^n} (cf. Cooley/Tukey (1965), [19]) and indeed this theorem is a generalization of this decomposition. Namely if G is an abelian group then all irreducible components of N are extensible (since $\phi \uparrow G$ has only irreducible components of degree 1) and we have $d = n$. The permutation matrix degenerates since the extensible irreducibles are treated as one block. We want to put this down in th following corollary.

1.76 Corollary If in the situation of Theorem 1.75 G is abelian then $d = n$ and

$$M = (\mathbf{1}_p \otimes A) \cdot \left(\bigoplus_{t\in T} \overline{\rho}(t) \right) \cdot (\mathrm{DFT}_p \otimes \mathbf{1}_n).$$

The diagonal elements of the matrix $\bigoplus_{t\in T} \overline{\rho}(t)$ are known as *Twiddle factors*.

Under a certain assumption the decomposition matrix of an induction can be determined in a slightly different way. This variant can be found in Minkwitz, [46], [48], but generally provides suboptimal decompositions. Nevertheless we want to state it here because the formula is more pleasant.

1.77 Theorem *Let $H \leq G$ be a subgroup of index k with representation ϕ of degree n having an extension $\overline{\phi}$ to G and let T a transversal of $H \backslash G$. Further assume A decomposes ϕ into irreducibles, i.e. $\phi^A = \rho$ and let $\overline{\rho}$ be an extension of ρ to G. Then*

$$(\phi \uparrow_T G)^{(\mathbf{1}_k \otimes A) \cdot D} = (1_H \uparrow_T G) \otimes \overline{\rho}, \text{ with } D = \bigoplus_{t \in T} \overline{\rho}(t).$$

If in particular $H \trianglelefteq G$, G/H abelian and $\overline{\rho}$ is in decomposed into irreducibles too, then

$$(\mathbf{1}_k \otimes A) \cdot D \cdot (\mathrm{DFT}_{G/H} \otimes \mathbf{1}_n)$$

is a decomposition matrix for $\phi \uparrow_T G$.

$$\begin{tikzcd} G & \overline{\phi} & \overline{\rho} \\ H \arrow[u, dash] & \phi \arrow[u, "\mathrm{ext}"] \arrow[r, "A", "\mathrm{dec}"'] & \rho \arrow[u, "\mathrm{ext}"] \end{tikzcd}$$

Figure 1.13: Situation in Theorem 1.75

Proof By Theorem 1.6 we get

$$(\phi \uparrow_T G)^{\mathbf{1}_k \otimes A} = \phi^A \uparrow_T G = \rho \uparrow_T G$$

and from Theorem 1.15 follows $(\rho \uparrow_T G)^D = (1_H \uparrow_T G) \otimes \overline{\rho},\ D = \bigoplus_{t \in T} \overline{\rho}(t)$.

If now $H \trianglelefteq G$ and G/H is abelian, then $1_H \uparrow_T G$ is a regular representation of G/H which is decomposed by $\mathrm{DFT}_{G/H}$ into $(G : H) = k$ irreducibles $\lambda_1, \ldots, \lambda_k$ of degree 1. Set $M = (\mathbf{1}_n \otimes A) \cdot D \cdot (\mathrm{DFT}_{G/H} \otimes \mathbf{1}_{\deg(\phi)})$, then

$$(\phi \uparrow_T G)^M = \left(\bigoplus_{i=1}^{k} \lambda_i \right) \otimes \overline{\rho} = \bigoplus_{i=1}^{k} (\lambda_i \cdot \overline{\rho})$$

which is decomposed into irreducibles. ∎

The additional condition, that ρ has an extension $\overline{\rho}$ to G, is certainly satisfied in the abelian case. If in addition $H \trianglelefteq G$ is a normal subgroup of prime index we get the same result as in Corollary 1.76. In all other cases, however, the decomposition from

Theorem 1.77 is worse than the one proposed by Theorem 1.75 since in the former occur n DFT_p's and in the latter only $d \leq n$ (number of extensible irreducibles).

Nevertheless we want to consider briefly the question when a representation of a normal subgroup of prime index has an extension.

1.78 Theorem *Let $N \overset{p}{\trianglelefteq} G$ be a normal subgroup of prime index p with representation ϕ and let $T = (t^0, t^1, \ldots, t^{(p-1)})$ be a transversal of G/N. Then ϕ has an extension $\overline{\phi}$ to G if and only if $\phi^t \cong \phi$.*

Proof If $\overline{\phi}$ is an extension of ϕ then

$$\phi^t(x) = \phi(txt^{-1}) = \phi(x)^{\overline{\phi}(t^{-1})},$$

and hence $\phi^t \cong \phi$. For the reverse direction we first remark that a representation has an extension if and only if this holds for any conjugate. Now assume $\phi^t \cong \phi$ and let $\rho = \bigoplus_{i=1}^n \rho_i \cong \phi$ be a decomposition of ϕ into irreducibles. Then G/N operates on the irreducibles via inner conjugation (ρ_i irreducible $\Leftrightarrow$ ρ_i^t irreducible) up to equivalence. According to the Theorem of Clifford there are two possibilities for the orbits: either an orbit is of length 1 ($\rho_i \cong \rho_i^t$) or of length p ($\rho_i^{t^0}, \rho_i^{t^1}, \ldots, \rho_i^{t^{(p-1)}}$). In the first case ρ_i has an extension, in the second the orbit can be extended simultaneously (after appropriate conjugation) by $\rho_i \uparrow_T G$ using Corollary 1.13. This implies that ρ has an extension and hence also ϕ. ∎

1.79 Corollary If in the situation of Theorem 1.78 $\phi = 1_{\mathsf{E}} \uparrow_S N$ is even a regular representation of N then ϕ has an extension to G.

Proof We have

$$\phi^t = (1_{\mathsf{E}} \uparrow_S N)^t = (1_{\mathsf{E}} \uparrow_{S^t} N) \cong \phi$$

by Theorem 1.11 and by Theorem 1.78 follows the result. ∎

Regular representations of solvable groups thus can be decomposed using Theorem 1.77. Now it is time for a little example.

1.80 Example Let $\mathsf{Z}_4 = \langle x \mid x^4 = 1 \rangle$ with trivial subgroup E. We want to decompose the regular representation

$$\phi = 1_{\mathsf{E}} \uparrow_T \mathsf{Z}_4, \; T = (1, x, x^2, x^3),$$

over $\mathbb{C}$. Of course this can be done using DFT_4 but we want to decompose along the composition series

$$\mathsf{E} \trianglelefteq \mathsf{Z}_2 \trianglelefteq \mathsf{Z}_4, \; \mathsf{Z}_2 = \langle x^2 \rangle$$

to also obtain a decomposition of DFT_4. Fist we decompose the induction. $T_1 = (1, x^2)$ is a transversal of $\mathsf{E}\backslash\mathsf{Z}_2$ and $T_2 = (1, x)$ one of $\mathsf{Z}_2\backslash\mathsf{Z}_4$ and $T_2T_1 = (1, x^2, x, x^3)$. Change of transversal by Theorem 1.1 leads to

$$(1_\mathsf{E} \uparrow_{T_1} \mathsf{Z}_2) \uparrow_{T_2} \mathsf{Z}_4 = 1 \uparrow_{T_2T_1} \mathsf{Z}_4 = (1 \uparrow_T \mathsf{Z}_4)^{[(2,3),4]}.$$

The representation $1_\mathsf{E} \uparrow_{T_1} \mathsf{Z}_2$ is decomposed by DFT_2 and we get

$$\rho = (1_\mathsf{E} \uparrow_{T_1} \mathsf{Z}_2)^{\mathrm{DFT}_2} = 1_{\mathsf{Z}_2} \oplus (x^2 \mapsto -1).$$

Both irreducible summand are extensible: 1_{Z_2} is obviously extended by 1_{Z_4} and $(x^2 \mapsto -1)$ by $(x \mapsto i)$ hence

$$\rho = 1_{\mathsf{Z}_2} \oplus (x^2 \mapsto -1) \xrightarrow{\text{ext}} \overline{\rho} = 1_{\mathsf{Z}_4} \oplus (x \mapsto i).$$

Evaluation of $\overline{\rho}$ at the transversal T_2 yields the Twiddle factors

$$\overline{\rho}(1) = \mathbf{1}_2,\ \overline{\rho}(x) = \mathrm{diag}(1, i)$$

and the decomposition matrix M is given by

$$M = [(2,3),4] \cdot (\mathbf{1}_2 \otimes \mathrm{DFT}_2) \cdot \mathrm{diag}(1,1,1,i) \cdot (\mathrm{DFT}_2 \otimes \mathbf{1}_2).$$

Since ϕ is decomposed by M exactly as by DFT_4 we get $\mathrm{DFT}_4 = M$. ∎

Abelian Representations The well-known decompositions of DFT_n can be deduced from the theorems of the preceding sections. The so-called Good-Thomas decomposition (cf. Agarwal (1977), [1]) corresponds to the decomposition of a regular representation of a Z_n into an outer tensor product. The Cooley-Tukey decomposition arises from an iterate application of Corollary 1.76. Since in this cases explicit formulas are available it makes sense to integrate them into a decomposition program to speed up computation.

Every abelian group decomposes into a direct product of cyclic groups of prime power order. By Theorem 1.34 monomial representations of abelian groups essentially are regular representations. By Corollary 1.31 regular representations decompose into an outer tensor product like the represented group into a direct product. This shows that for the purpose of decomposing a monomial representation of an abelian group the central building blocks are matrices DFT_{p^e}, p prime. A formula for the decomposition of the latter can, e.g. be found in the dissertation of Egner (1997), [26]. From there we cite the following lemma.

1.81 Lemma Let $n = p^e$, p prime and $e > 1$. Then

$$\begin{aligned}
\mathrm{DFT}_n \quad = \quad & (\mathbf{1}_1 \otimes \mathrm{DFT}_p \otimes \mathbf{1}_{p^{e-1}}) \cdot \mathrm{diag}\left(\omega_n^{t(e-1,k)} \mid k \in \{0,\ldots,n-1\}\right) \cdot \\
& (\mathbf{1}_p \otimes \mathrm{DFT}_p \otimes \mathbf{1}_{p^{e-2}}) \cdot \mathrm{diag}\left(\omega_n^{t(e-2,k)} \mid k \in \{0,\ldots,n-1\}\right) \cdot \\
& \qquad\qquad\qquad\qquad \vdots \\
& (\mathbf{1}_{p^{e-2}} \otimes \mathrm{DFT}_p \otimes \mathbf{1}_p) \cdot \mathrm{diag}\left(\omega_n^{t(1,k)} \mid k \in \{0,\ldots,n-1\}\right) \cdot \\
& (\mathbf{1}_{p^{e-1}} \otimes \mathrm{DFT}_p \otimes \mathbf{1}_1) \cdot [\sigma, n],
\end{aligned}$$

where the exponents $t(\alpha, k)$ of the Twiddle factors and the permutation σ of the set $\{0, \ldots, n-1\}$ are given by

$$\begin{aligned}
t(\alpha, k) \quad &= \quad \left((k \bmod p^{\alpha+1}) \operatorname{div} p^\alpha\right) \cdot (k \bmod p^\alpha) \cdot p^{(e-1)-\alpha} \quad \text{and} \\
k^\sigma \quad &= \quad \sum_{\ell=0}^{e-1} \left((k \operatorname{div} p^\ell) \bmod p\right) p^{(e-1)-\ell}, \\
& \qquad k \in \{0, \ldots, n-1\},\ \alpha \in \{1, \ldots, e-1\}.
\end{aligned}$$

(The permutation σ reverses the digits of k represented as a number with base p.)

2

Decomposing Monomial Representations

The main result of this dissertation is an algorithm for the decomposition of a large class of monomial representations and will be presented in this chapter. The results of Chapter 1 allow the formulation of such an algorithm in terms of macro operations. Furthermore, we will explain meticulously the actual realization of the algorithm at its critical points. Through various improvements the current implementation is about 100 times faster than its first version which is mainly due to a better understanding of the underlying mathematics.

First we explain what is meant by the decomposition of a representation with regard to the algorithm. If μ is a representation of degree n of a group G then we want to determine an $(n \times n)$-matrix A such that

$$\mu^A = A^{-1} \cdot \mu \cdot A = \bigoplus_{i=1}^{k} \rho_i, \text{ where } \rho_i \text{ is irreducible for } i = 1 \dots k.$$

Furthermore, equivalent irreducibles shall be equal, i.e.

$$\rho_i \cong \rho_j \Rightarrow \rho_i = \rho_j.$$

In addition equal irreducibles shall be adjacent and all irreducibles shall be ordered with respect to their degrees. Introduction of a total ordering on the irreducibles, e.g. via the character, has been turned out to be not useful since in most cases the computation of the character is not necessary for computing the decomposition. Making equal and collecting together of the irreducibles, however, leads to a considerable increase of performance: all computations (e.g. induction of an irreducible) have to be performed only once for a representation among a group of equals.

Essential for the generation of algorithms (cf. Chapter 3) is the fact that the decomposition matrix A is returned in factorized form, namely A is a product of sparse matrices. This product represents a fast algorithm for the multiplication with A.

Another important point is the fact that not only the decomposition matrix A is computed, but also the decomposition $\bigoplus_{i=1}^{k} \rho_i$, since the one cannot be easily computed from the other. If only A is given then the corresponding decomposition in general must be computed by explicit conjugation μ^A which is expensive for large ($n > 100$) degrees. If only the decomposition $\bigoplus_{i=1}^{k} \rho_i$ is given then a decomposition matrix A can be computed by Theorem 1.56 (Computation of the intertwining space, p. 49) solving a linear system of equations in n^2 unknowns. The result, however, is not factorized.

The main instrument for the decomposition of μ is provided by Theorem 1.75 on p. 63. If $\mu^D = \lambda_H \uparrow_T G$ is decomposed into an induction then this theorem allows to decompose recursively along a composition series from H to G. The construction along the composition series determines the factorization of the decomposition matrix A. The fact that such a composition series in general does not exist poses the only restriction to this method.

Another method, described in Theorem 1.66 on p. 54, solves the decomposition of a permutation representation π with regularly represented abelian normal subgroup N. In this case, the decomposition of the restriction $\pi \downarrow N$ essentially yields a decomposition of π. This class of representations contains all primitive representations of solvable groups.

A particular property of the algorithm is the fact that it is virtually never necessary to compute the character table of the underlying group.

2.1 The Algorithm

Again we want to recall that we exclusively consider representations satisfying the Maschke condition. First we will give the algorithm which allows to decompose a transitive permutation representation π of G in the case that G has a regularly represented abelian normal subgroup. The algorithm follows the proof of Theorem 1.66 on p. 54.

2.1 Algorithm (Decomposition by Regular Abelian Normal Subgroup)
Given is a transitive permutation representation π of G with kernel $\ker(\pi)$ and a normal subgroup $N \trianglelefteq G$ such that $N/\ker(\pi)$ is abelian and regularly represented. π shall be decomposed.

1. Decompose $\pi = 1_H \uparrow_T G$ by Algorithm 1.26 (Decomposition into an induction, p. 31).

2. Decompose $\pi \downarrow N$ by a matrix B, i.e.

$$(\pi \downarrow N)^B = \bigoplus_{i=1}^{n} \lambda_i, \ \deg(\lambda_i) = 1.$$

3. Compute the orbits $O_1, \ldots, O_k$ of $L = \{\lambda_1, \ldots, \lambda_n\}$ under the operation of G via inner conjugation with corresponding conjugations $T_1, \ldots, T_k$, i.e. $O_i = \{\lambda_{i,1}, \ldots, \lambda_{i,r_i}\}$, $T_i = \{t_{i,1}, \ldots, t_{i,r_i}\}$ and $\lambda_{i,1}^{t_{i,j}} = \lambda_{i,j}, \ j = 1 \ldots r_i$. Furthermore, determine the stabilizers $G_{\lambda_{i,1}}, \ i = 1 \ldots k$.

   Note that $G = H \ltimes N$ if H is the stabilizer of a point under π. The operation of G on the irreducibles hence corresponds to the operation of H. In order to compute the orbits, the character values of the λ_i are precomputed. H has a permutation representation on the conjugacy classes (of size 1) of N. Irreducibles λ_i, λ_j are equivalent if and only if the corresponding lists of character values can be mapped onto each other by a permutation.

4. Calculate a permutation matrix P_1 which maps L onto the concatenation $O_1 \cup \cdots \cup O_k$.

5. Extend $\lambda_{i,1}$ to $G_{\lambda_{i,1}}$ by $\overline{\lambda}_{i,1}(hn) = \lambda_{i,1}(n), \ h \in H, \ n \in N$.

6. Calculate the inductions

$$\rho_i = \overline{\lambda}_{i,1} \uparrow_{T_i} G, \ i = 1 \ldots k.$$

7. Sort the ρ_i by degree with a permutation matrix P_2. Note that the ρ_i are pairwise inequivalent.

$A = B \cdot P_1 \cdot P_2$ is a decomposition matrix for π and $\pi^A = \bigoplus_{i=1}^{k} \rho_i$. ■

The decomposition of $\pi \downarrow N$ is performed by the following algorithm as a special case. Note that in Algorithm 2.1 the normal subgroup N is passed as an argument, i.e. it is not calculated within this algorithm. Now we turn to the general case.

2.2 Algorithm (Decomposition of Monomial Representations)
Given is a monomial representation μ of degree n of a group G. For μ a (factorized) matrix A and irreducible representations $\rho_i, \ i = 1 \ldots k$ shall be computed such that the following conditions are satisfied:

1. $\rho_i \cong \rho_j \Rightarrow \rho_i = \rho_j$.
2. $i < j \Rightarrow \deg(\rho_i) \leq \deg(\rho_j)$.
3. Equal irreducibles are adjacent.

Case 1: μ is irreducible

The irreducibility is tested using the character of mu. Note that a permutation representation of degree > 1 is never irreducible.
$A = \mathbf{1}_n$ is a decomposition matrix with decomposition $\rho = \mu$.

Case 2: μ is not transitive

1. Decompose μ by Algorithm 1.23 (Orbit decomposition, p. 29) according to the orbits with a permutation matrix P_1:

$$\mu^{P_1} = \bigoplus_{i=1}^{k} \mu_i, \ \ \mu_i \text{ transitive.}$$

2. Decompose recursively μ_i for $i = 1 \ldots k$, i.e.

$$\mu_i^{A_i} = \bigoplus_{j=1}^{r_i} \rho_{i,j}.$$

3. Compute a block diagonal matrix D which conjugates equivalent irreducibles of different μ_i to be equal. This is done by solving a system of linear equations according to Theorem 1.56 (Computation of the intertwining space, p. 49). Note that equivalent irreducibles of the same μ_i are already equal. The blocks in D correspond (in the coarsest case) to the degrees of $\rho_{i,j}$.
4. Determine a permutation matrix P_2 which sorts the $\rho_{i,j}$ according to their degrees such that equals are adjacent.

$A = P_1 \cdot \bigoplus_{i=1}^{k} A_i \cdot D \cdot P_2$ decomposes μ.

Case 3: μ is a two-fold transitive permutation representation

The matrix

$$A = \mathrm{SOR}_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & -1 & 0 & \cdots & 0 \\ 1 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & 0 & 0 & & -1 \end{bmatrix}$$

decomposes μ into $1_G \oplus \rho$. In this case the two irreducibles are computed by conjugating μ with the sparse ($3n-2$ entries) matrix A. Note that then the irreducibles are already sorted by degree.

Case 4: $\deg(\mu) = p$ is prime and μ is not two-fold transitive

Case 4a: μ is no permutation representation

1. Decompose $\mu^{D_1} = \lambda_H \uparrow_T G$ by Algorithm 1.26 (Decomposition into an induction, p. 31). D_1 is a diagonal matrix.
2. Extend λ_H by λ_G to G using Algorithm 1.74 (Extension without character, p. 62).
3. Compute $D_2 = \operatorname{diag}(\lambda_G(t) \mid t \in T)$.
4. Decompose recursively $1_H \uparrow_T G$ with B, i.e.

$$(1_H \uparrow_T G)^B = \bigoplus_{i=1}^{k} \rho_i.$$

$A = D_1 \cdot D_2 \cdot B$ is a decomposition matrix for μ and $\mu^A = \bigoplus_{i=1}^{k} \lambda_G \cdot \rho_i$. Note that with the ρ_i also the $\lambda_G \cdot \rho_i$ are sorted. λ_H has an extension by Theorem 1.36 on p. 36, which can be calculated by Algorithm 1.74 (Extension without character, p. 62), since according to Theorem 1.35 of Burnside on p. 36 G and hence $G/N \cong H$ is solvable.

Case 4b: μ is a permutation representation

1. Decompose $\mu = 1_H \uparrow_T G$ by Algorithm 1.26 (Decomposition into an induction, p. 31) and determine $\ker(\mu) = \operatorname{core}(H)$.
2. Compute the p-Sylow subgroup N of $G/\operatorname{core}(H)$. This is an abelian, (by μ) regularly represented normal subgroup.
3. Decompose μ by Algorithm 2.1 with respect to N.

Case 5: μ is a primitive permutation representation with solvable minimal normal subgroup

According to Theorem 1.70 on p. 59 then necessarily $\deg(\mu)$ is a prime power.

1. Decompose $\mu = 1_H \uparrow_T G$ by Algorithm 1.26 (Decomposition into an induction, p. 31) and compute $\ker(\mu) = \operatorname{core}(H)$.

2. Compute a minimal normal subgroup N of $G/\operatorname{core}(H)$.
3. If N is not solvable, then μ cannot be decomposed. Otherwise $N \cong \mathsf{Z}_p^k$ is abelian and is regularly represented by μ. Decompose μ by Algorithm 2.1 with respect to N.

Case 6: $G/\ker(\mu)$ is abelian

Fall 6a: μ is no permutation representation

1. Decompose $\mu^{D_1} = \lambda_H \uparrow_T G$ by Algorithm 1.26 (Decomposition into an induction, p. 31). D_1 is a diagonal matrix.
2. Extend λ_H by λ_G to G using Algorithm 1.74 (Extension without character, p. 62).
3. Compute $D_2 = \operatorname{diag}(\lambda_G(t) \mid t \in T)$.
4. Decompose recursively $1_H \uparrow_T G$ by B, i.e.

$$(1_H \uparrow_T G)^B = \bigoplus_{i=1}^{k} \rho_i,\ \deg(\rho_i) = 1,\ i = 1 \ldots k.$$

$A = D_1 \cdot D_2 \cdot B$ is a decomposition matrix for μ and $\mu^A = \bigoplus_{i=1}^{k} \lambda_G \cdot \rho_i$. Note that with the ρ_i also the $\lambda_G \cdot \rho_i$ are sorted.

Fall 6b: $G_1 = G/\ker(\mu)$ is a cyclic group of prime power order p^k

1. Determine $g \in G$ generating G_1. Note that such a g must be contained in the generating set of G. μ is a regular representation of G_1. Compute $\mu(g)$.
2. Determine a permutation matrix P conjugating $\mu(g)$ onto $[(1, \ldots, p^k), p^k]$.
3. Compute a decomposition matrix B of μ^P, $g \mapsto [(1, \ldots, p^k), p^k]$ using Lemma 1.81 on p. 68.

$A = P \cdot B$ is a decomposition matrix for μ with decomposition

$$\mu^A = \bigoplus_{i=0}^{p^k-1} \rho_i,\ \rho_i:\ g \mapsto \omega_{p^k}^i.$$

Fall 6c: μ is a decomposition matrix

1. Decompose $\mu = 1_H \uparrow_T G$ by Algorithm 1.26 (Decomposition into an induction, p. 31). According to Theorem 1.34 on p. 36 μ is a regular representation of the factor group $G_1 = G/H$.

2. Decompose G_1 into a direct product $G_1 = N_1 \times \cdots \times N_k$ of cyclic groups of prime power order, $N_i = \langle x_i \rangle$.
3. Compute the permutation matrix P corresponding to the change of transversal $T \to T_1 \cdot T_2 \cdots T_k$, where $T_i = (x_i^0, x_i^1, x_i^2, \ldots)$ is a list of the elements of N_i.
4. Decompose recursively $1_{\mathsf{E}} \uparrow_{T_i} N_i$ by A_i

$$(1_{\mathsf{E}} \uparrow_{T_i} N_i)^{A_i} = \bigoplus_{j=1}^{r_i} \rho_{i,j}.$$

$A = P \cdot \bigotimes_{i=1}^{k} A_i$ is a decomposition matrix of μ with decomposition

$$\mu^A = \bigoplus_{j_1=1}^{r_1} \cdots \bigoplus_{j_k=1}^{r_k} (\rho_{1,j} \# \ldots \# \rho_{k,j}).$$

Case 7: $\boldsymbol{\mu}$ is equivalent to an outer tensor product

1. Decompose μ into a conjugated outer tensor product using Algorithm 1.32 (Decomposition into an outer tensor product, p. 35)

$$\mu^M = \mu_1 \# \ldots \# \mu_k,$$

where M is a monomial matrix.

2. Decompose recursively the representations $\mu_i,\ i = 1 \ldots k$

$$\mu_i^{A_i} = \bigoplus_{j=1}^{r_i} \rho_{i,j}.$$

3. Determine a permutation matrix P, such that

$$(\bigoplus_{j=1}^{r_1} \rho_{1,j} \# \ldots \# \bigoplus_{j=1}^{r_k} \rho_{k,j})^P = \bigoplus_{j_1=1}^{r_1} \cdots \bigoplus_{j_k=1}^{r_k} (\rho_{1,j} \# \ldots \# \rho_{k,j}).$$

P only depends on the degrees of the $\rho_{i,j}$. The computation is a simple combinatorial problem.

$A = M \cdot \bigotimes_{i=1}^{k} A_i \cdot P$ is a decomposition matrix for μ and

$$\mu^A = \bigoplus_{j_1=1}^{r_1} \cdots \bigoplus_{j_k=1}^{r_k} (\rho_{1,j} \# \ldots \# \rho_{k,j}).$$

Case 8: If $\mu^D = \lambda_H \uparrow_T G$, then a normal subgroup exists with $H \leq N \overset{p}{\trianglelefteq} G$ and p prime

1. Decompose $\mu^{D_1} = \lambda_H \uparrow_T G$ by Algorithm 1.26 (Decomposition into an induction, p. 31). D is a diagonal matrix.
2. Determine N with $H \leq N \overset{p}{\trianglelefteq} G$. For this purpose build the normal closure $\overline{H}$ of H in G and compute a composition series of $G/\overline{H}$.
3. Decompose $\lambda_H \uparrow_T G$ with a monomial matrix M into a double induction
$$(\lambda_H \uparrow_T G)^M = (\lambda_H \uparrow_{T_1} N) \uparrow_{T_2} G,$$
such that $T_2 = (t^0, t^1 \ldots t^{(p-1)})$. According to Theorem 1.4 on p. 16, M is the matrix corresponding to the change of transversals $T \to T_1T_2$ and is computed by Algorithm 1.2 (Change of transversals, p. 15).
4. Decompose recursively $\lambda_H \uparrow_{T_1} N$ by B, i.e.
$$(\lambda_H \uparrow_{T_1} N)^B = \bigoplus_{i=1}^{k} \rho_i.$$
5. Collect the ρ_i with respect to equality. The following computations then only have to performed for one irreducible in each group of equals. Permutations to sort the ρ_i can be performed on these groups.
6. Determine those ρ_i having an extension $\overline{\rho}_i$ to G (cf. Theorem 1.71 of Clifford, p. 60). We will denote these by η_ℓ and the direct sum of them by η:
$$\eta = \bigoplus_{\ell=1}^{k} \eta_\ell.$$
Use the fact that ρ_i has an extension if and only if $\rho_i \cong \rho_i^t$. To decide this, the character values of the ρ_i are precomputed as well as the the permutation raised by the conjugation of t on the conjugacy classes. Then the condition $\rho_i \cong \rho_i^t$ holds if and only if the list of character values of ρ_i is invariant under this permutation.
7. Determine for the ρ_i having no extension, whether they are inner conjugates of one another with respect to a power t^j, $j = 1 \ldots p-1$.

   For this purpose, again use the lists of character values and the permutation induced by t^j on them. Note that two such inner conjugates have this property only up to equivalence.

8. Compute a permutation matrix P_1 sorting the ρ_i such that first come the extensible, and then the others, sorted into groups with respect to inner conjugation (up to equivalence) of the form

$$\rho, \ldots, \rho, \rho^t, \ldots, \rho^t, \ldots, \rho^{t^{(p-1)}}, \ldots, \rho^{t^{(p-1)}}.$$

9. Calculate a block diagonal matrix D_2 which conjugates the irreducibles in the groups above, such that equality holds (this means: a representation which is $\cong \rho^t$ is conjugated such that $= \rho^t$ holds). For this purpose, a linear equations are solved using Theorem 1.56 (Computation of the intertwining space, p. 49). On the extensible ρ_i, D_2 is the identity. The direct sum of a group of inner conjugates will be denoted by ν_ℓ, $\ell = 1 \ldots m$:

$$\nu_\ell = \rho \oplus \ldots \oplus \rho \oplus \rho^t \oplus \ldots \oplus \rho^t \oplus \ldots \oplus \rho^{t^{(p-1)}} \oplus \ldots \oplus \rho^{t^{(p-1)}}.$$

Now the matrix $B \cdot P_1 \cdot D_2$ is a decomposition matrix for $\lambda_H \uparrow_{T_1} N$, too, and

$$(\lambda_H \uparrow_{T_1} N)^{B \cdot P_1 \cdot D_2} = \eta \oplus \nu_1 \oplus \ldots \oplus \nu_m$$

and

$$((\lambda_H \uparrow_{T_1} N) \uparrow_{T_2} G)^{\mathbf{1}_p \otimes B \cdot P_1 \cdot D_2} = (\eta \oplus \nu_1 \oplus \ldots \oplus \nu_m) \uparrow_{T_2} G$$

by Theorem 1.6 on p. 18.

10. Compute a permutation matrix P_2 by Theorem 1.5 on p. 17, which decomposes the induction of the direct sum above into a direct sum of inductions. Here η is considered as one summand and the irreducible summands of the ν_ℓ are the further summands.

11. Extend η by $\overline{\eta}$ to G. For this purpose extend the summands using Algorithm 1.74 (Extension without character, p. 62) or using the extension formula 1.41 of Minkwitz on p. 39.

12. Evaluate $\overline{\eta}$ at t and compute the block diagonal matrix

$$D_3 = \bigoplus_{i=0}^{p-1} \overline{\eta}(t)^i.$$

Then $(\eta \uparrow_{T_2} G)^{D_3} = (1_N \uparrow_{T_2} G) \otimes \overline{\eta}$ by Theorem 1.15 on p. 23. The representation $1_N \uparrow_{T_2} G$ is a regular representation of $G/N \cong \mathsf{Z}_p$. D_3 has size $p \cdot \deg(\eta)$.

13. Compute the p one-dimensional representations $\lambda^0, \lambda^1, \ldots, \lambda^{p-1}$ of G/N. Then

$$((1_N \uparrow_{T_2} G) \otimes \overline{\eta})^{\mathrm{DFT}_p \otimes \mathbf{1}_{\deg(\eta)}} = \bigoplus_{i=0}^{p-1} \bigoplus_{\ell=1}^{k} \lambda^i \cdot \eta_\ell,$$

which is decompose into irreducibles.

14. Determine a block diagonal matrix D_4, which conjugates the inductions of the summands of ν_ℓ to be equal.

    The inductions of the irreducible summands of ν_ℓ are already irreducible. The summands of one ν_ℓ have equivalent inductions, i.e. $\rho \uparrow G \cong \rho^{t^i} \uparrow G$. These can be conjugated by a matrix D_4 to be equal (Theorem 1.71 of Clifford). D_4 is of size $n - p \cdot \deg(\eta)$.

15. Sort the irreducibles with a permutation matrix P_3. Note that equal irreducible already are adjacent.

$$A = D_1 \cdot M \cdot (\mathbf{1}_p \otimes B \cdot P_1 \cdot D_2) \cdot P_2 \cdot ((D_3 \cdot (\mathrm{DFT}_p \otimes \mathbf{1}_{\deg(\eta)})) \oplus D_4) \cdot P_3$$

is a decomposition matrix for μ with decomposition

$$\mu^A = \left(\bigoplus_{i=0}^{p-1} \bigoplus_{\ell=1}^{k} \lambda^i \cdot \eta_\ell \oplus \bigoplus_{\ell=1}^{m} \underbrace{(\nu_{\ell,1} \uparrow_{T_2} G \oplus \ldots \oplus \nu_{\ell,1} \uparrow_{T_2} G)}_{N_\ell} \right)^{P_3},$$

where N_ℓ is the number of summands of ν_ℓ and $\nu_{\ell,1}$ is the first summand of ν_ℓ.

Case 9: μ is a permutation representation and it exists a regularly represented abelian normal subgroup of $G/\ker(\mu)$

1. Compute a normal subgroup N, $\ker(\mu) \leq N \trianglelefteq G$, such that $N/\ker(\mu)$ is abelian and regularly represented by μ. Necessarily, $|N| = \deg(\mu)$.
2. Decompose μ by Algorithm 2.1 with respect to N.

Case 10: If $\mu^D = \lambda_H \uparrow_T G$, then λ_H has an extension to G

1. Decompose $\mu^{D_1} = \lambda_H \uparrow_T G$ by Algorithm 1.26 (Decomposition into an induction, p. 31). D_1 is a diagonal matrix.
2. Extend λ_H by λ_G to G. λ_G is a character of G of degree 1.

    Note that for this purpose it is sufficient to compute the character table of the abelian group G/G' (G' is the derived subgroup of G).
3. Compute $D_2 = \mathrm{diag}(\lambda_G(t) \mid t \in T)$.

4. Decompose recursively $1_H \uparrow_T G$ by B, i.e.

$$(1_H \uparrow_T G)^B = \bigoplus_{i=1}^{k} \rho_i,\ \deg(\rho_i) = 1,\ i = 1 \ldots k.$$

$A = D_1 \cdot D_2 \cdot B$ is a decomposition matrix for μ and $\mu^A = \bigoplus_{i=1}^{k} \lambda_G \cdot \rho_i$. Note that with the ρ_i also the $\lambda_G \cdot \rho_i$ are sorted. ∎

By far the most powerful part of Algorithm 2.2 is given by Case 8. It alone allows the decomposition of a monomial representation $\mu = \lambda_H \uparrow_T G$ of the solvable group G, if H is subnormal in G, i.e. if H is in any composition series of G.

The decomposition of a permutation representation with abelian, regularly represented normal subgroup is another method, handling among other things a large class of primitive permutation representations. In the following lemma we will show that these methods do not collide, i.e. if a permutation representation $\pi = 1_H \uparrow_T G$ has an abelian, regularly represented normal subgroup, and a normal subgroup $N \trianglelefteq G$ of prime index p, then also $1_H \uparrow N$ has an abelian, regularly represented normal subgroup.

2.3 Lemma Let $\pi = 1_H \uparrow G$ be a permutation representation of G with kernel $\ker(\pi) = \operatorname{core}(H)$. Assume N is a normal subgroup, such that $N/\operatorname{core}(H)$ is abelian and regularly represented by π. Assume further the existence of a normal subgroup K with $H \leq K \trianglelefteq G$. By Lemma 1.64 on p. 54 we have $H \cap N = \operatorname{core}(H)$ and $HN = G$.

Then the abelian group $(K \cap N)/\operatorname{core}(H)$ is regularly represented by $1_H \uparrow K$.

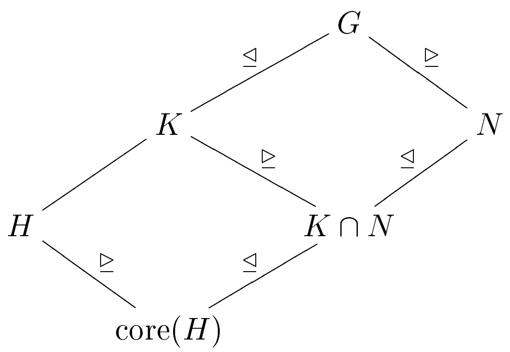

Figure 2.1: Situation in Lemma 2.3

Proof We have $(N : (K \cap N)) = (NK : K) = (G : K)$, $(H \cap (K \cap N)) = \mathrm{core}(H)$ and $H(K \cap N) = K$ (cf. Figure 2.1). Using Corollary 1.14 on p. 23 we get

$$(1_H \uparrow K) \downarrow (K \cap N) \cong 1_{\mathrm{core}(H)} \uparrow (K \cap N),$$

which is regular. ∎

If a monomial representation $\mu = \lambda_H \uparrow_T G$ is decomposed exclusively by using Case 8 in Algorithm 2.2, i.e. $\mu^A = \rho = \bigoplus_{i=1}^n \rho_i$, then a chain $\mathcal{T}$ of normal subgroups

$$H = N_{k+1} \trianglelefteq N_k \trianglelefteq \ldots \trianglelefteq G = N_1, \ (N_i : N_{i+1}) \text{ is prim}, \ i = 1 \ldots k,$$

has been chosen during decomposition. In this case the decomposition has two additional properties. First, it is by construction $\mathcal{T}$-adapted in the sense of Clausen/Baum, i.e. (cf. [16], p. 96):

1. The restrictions $\rho \downarrow N_i$, $i = 1 \ldots k+1$, are *equal* to a direct sum of irreducible representations.

2. Equivalent irreducible components in $\rho \downarrow N_i$ are equal.

The conception of $\mathcal{T}$-adapted decompositions is used in [16] to obtain results on the complexity of fast Fourier transforms.

Another property of the computed decomposition is the fact that every irreducible ρ_i, which is equivalent to a monomial representation, is even *equal* to a monomial representation. What is the reason? Assume, ϕ is a irreducible component of $\lambda_H \uparrow N_i$. According to Theorem 1.71 of Clifford on p. 60 there are two possibilities: either the induction $\phi \uparrow N_{i-1}$ is irreducible or ϕ has an extension $\overline{\phi}$ to N_{i-1}. The first case generates a monomial representation from a monomial one. In the second case, Algorithm 1.74 (Extension without character, p. 62) is used to compute a matrix M lying in the one-dimensional space $\mathsf{Int}(\phi, \phi^t)$, $t \in N_{i-1} \setminus N_i$ (difference of sets) and hence $\overline{\phi}$ is monomial if and only if M is monomial and hence every matrix $\neq \mathbf{0}$, $\in \mathsf{Int}(\phi, \phi^t)$.

2.2 An Example

We want to conclude this chapter with a detailed example for Algorithm 2.2.

2.4 Example For convenience we will define the following groups as permutation groups rather than by generators and relations. Let

$$G = \langle (1,2,3,4), (1,2) \rangle \cong \mathsf{S}_4$$

be the symmetric group on 4 points with representation

$$\begin{array}{lll}\mu: & (1,2,3,4) & \mapsto [\,(1,10,17,19)(2,9,18,20)(3,12,14,21)\\ & & \quad (4,11,13,22)(5,7,16,23)(6,8,15,24),\ 24\,],\\ & (1,2) & \mapsto [\,(1,7)(2,8)(3,9)(4,10)(5,11)(6,12)(13,15)\\ & & \quad (14,16)(17,18)(19,21)(20,22)(23,24),\ 24\,].\end{array}$$

μ is transitive of degree 24, hence a regular representation of G. We want to execute Algorithm 2.2 step by step. We easily see that Case 8 is applicable.

1. Using Algorithm 1.26 (Decomposition into an induction, p. 31) we decompose μ as
$$\mu^{D_1} = (1_H \uparrow_T G),$$
with $D_1 = \mathbf{1}_{24}$, $H = \mathsf{E}$ and
$$\begin{array}{rcl}T & = & (\,(1),(3,4),(2,3),(2,3,4),(2,4,3),(2,4),(1,2),(1,2)(3,4),\\ & & (1,2,3),(1,2,3,4),(1,2,4,3),(1,2,4),(1,3,2),(1,3,4,2),\\ & & (1,3),(1,3,4),(1,3)(2,4),(1,3,2,4),(1,4,3,2),(1,4,2),\\ & & (1,4,3),(1,4),(1,4,2,3),(1,4)(2,3)\,).\end{array}$$

2. It exists only one normal subgroup $N \trianglelefteq G$ of prime index, namely
$$N = \langle (1,3,2),(1,4)(2,3)\rangle \cong \mathsf{A}_4$$
and we have $(G:N) = p = 2$.

3. We decompose $1_{\mathsf{E}} \uparrow_T G$ by Algorithm 1.2 (Change of transversal, p. 15) into a double induction, i.e.
$$(1_{\mathsf{E}} \uparrow_T G)^M = (1_{\mathsf{E}} \uparrow_{T_1} N) \uparrow_{T_2} G$$
with
$$\begin{array}{rcl}M & = & [\,(2,22,19,21,11,14,18,17,9,5,3,23,20,10,13,7,15,16,8,4)\\ & & (6,24,12),\ 24\,],\\ T_1 & = & (\,(1),(2,3,4),(2,4,3),(1,2)(3,4),(1,2,3),(1,2,4),(1,3,2),\\ & & (1,3,4),(1,3)(2,4),(1,4,2),(1,4,3),(1,4)(2,3)\,),\\ T_2 & = & (\,(1),(1,2,3,4)\,).\end{array}$$
The transversal $T_2 = (t^0, t^1)$ is generated by $t = (1,2,3,4)$.

4. We decompose recursively the regular representation $1_{\mathsf{E}} \uparrow_{T_1} N$ of N. We will not execute the steps but only state the result. It is

$$(1_{\mathsf{E}} \uparrow_{T_1} N)^B = \rho_1 \oplus \rho_2 \oplus \rho_3 \oplus \rho_4 \oplus \rho_4 \oplus \rho_4,$$

where the ρ_i are given by

$$\begin{array}{lll} \rho_1 = 1_N & & \\ \rho_2 : & (1,3,2) \mapsto \omega_3^2, & (1,4)(2,3) \mapsto 1, \\ \rho_3 : & (1,3,2) \mapsto \omega_3, & (1,4)(2,3) \mapsto 1, \\ \rho_4 : & (1,3,2) \mapsto [\,(1,2,3),\ 3\,], & (1,4)(2,3) \mapsto \mathrm{diag}(1,-1,-1), \end{array}$$

and the decomposition matrix B by

$$\begin{aligned} B \;=\; & [\,(2,6,7,5,9,3,11,8,10,12,4),\ 12\,]\cdot \\ & (\mathbf{1}_3 \otimes [\,(2,3),\ 4\,] \cdot (\mathrm{DFT}_2 \otimes \mathrm{DFT}_2) \cdot [\,(2,4),\ 4\,]) \cdot \\ & [\,(2,4,8,9,3,12,7,10,6,5),\ 12\,]\cdot \\ & (\mathrm{DFT}_3 \oplus \mathbf{1}_9) \cdot [\,(2,3),\ 12\,]. \end{aligned}$$

5. We need to consider the 4 pairwise inequivalent irreducible representations $\rho_1, \ldots, \rho_4$.

6. $(\,(1),(2,3,4),(2,4,3),(1,2)(3,4)\,)$ is a system of representatives of the conjugacy classes of N. The characters of the ρ_i are given by the values on the conjugacy classes:

$$\chi_{\rho_1}:\ (1,1,1,1),\ \chi_{\rho_2}:\ (1,\omega_3^2,\omega_3,1),\ \chi_{\rho_3}:\ (1,\omega_3,\omega_3^2,1),\ \chi_{\rho_4}:\ (3,0,0,-1).$$

The transversal generator t permutes the conjugacy classes as $(2,3)$, hence ρ_i has an extension to G if and only if the list of character values is invariant under $(2,3)$. It follows that ρ_1 and ρ_4 has an extension and we get

$$\eta_1 = \rho_1,\ \eta_2 = \eta_3 = \eta_4 = \rho_4,\ \eta = \bigoplus_{i=1}^{4} \eta_i.$$

7. ρ_2 and ρ_3 have no extension to G. The character values show $\rho_2^t \cong \rho_3$ and since both are of degree 1, even $\rho_2^t = \rho_3$.

8. The permutation matrix $P_1 = [\,(2,11,9,7,5,3,12,10,8,6,4),\ 12\,]$ conjugates the ρ_i into the order
$$\rho_1 \oplus \rho_4 \oplus \rho_4 \oplus \rho_4 \oplus \rho_2 \oplus \rho_2^t.$$

9. Since already $\rho_2^t = \rho_3$, we can choose $D_2 = \mathbf{1}_{12}$. We get $\nu_1 = \rho_2 \oplus \rho_2^t$ and
$$(1_{\mathsf{E}} \uparrow_{T_1} N)^{B\cdot P_1 \cdot D_2} = \eta \oplus \nu_1,$$
resp.
$$((1_{\mathsf{E}} \uparrow_{T_1} N) \uparrow_{T_2} G)^{\mathbf{1}_2 \otimes B \cdot P_1 \cdot D_2} = (\eta \oplus \nu_1) \uparrow_{T_2} G.$$

10. Using Theorem 1.5 on p. 17 we get
$$P_2 = [\,(11,13,15,17,19,21)(12,14,16,18,20,22,23),\ 24\,]$$
and we have
$$((\eta \oplus \nu_1) \uparrow_{T_2} G)^{P_2} = (\eta \uparrow_{T_2} G) \oplus (\rho_2 \uparrow_{T_2} G) \oplus (\rho_2^t \uparrow_{T_2} G).$$

11. We extend η to G. The representation $\eta_1 = \rho_1 = 1_N$ obviously has the extension $\overline{\eta}_1 = 1_G$, $\eta_2 = \rho_4$ is extended using Minkwitz or Algorithm 1.74 (Extension without character, p. 62) by
$$\overline{\eta}_2 :\ (1,2,3,4) \mapsto [\,(1,2),\ (-1,1,1)\,],\ (1,2) \mapsto [\,(1,3),\ (-1,-1,-1)\,].$$
We have $\overline{\eta}_2 = \overline{\eta}_3 = \overline{\eta}_4 = \overline{\rho}_4$ and
$$\overline{\eta} = \bigoplus_{i=1}^{4} \eta_i.$$

12. $\overline{\eta}(t) = [\,(2,3)(5,6)(8,9),\ (1,-1,1,1,-1,1,1,-1,1,1)\,]$ and hence
$$D_3 = \mathbf{1}_{10} \oplus \overline{\eta}(t).$$
D_3 decomposes $\eta \uparrow_{T_2} G$ into a tensor product
$$(\eta \uparrow_{T_2} G)^{D_3} = (1_N \uparrow_{T_2} G) \otimes \overline{\eta}.$$

13. It is

$$\lambda: \ (1,2,3,4) \mapsto -1, \ (1,2) \mapsto -1,$$

the generating one-dimensional representation of G/N and

$$((1_N \uparrow_{T_2} G) \otimes \overline{\eta})^{\mathrm{DFT}_2 \otimes \mathbf{1}_{10}} = \bigoplus_{i=1}^{2} \bigoplus_{j=1}^{4} \lambda^i \cdot \overline{\eta}_j$$

is decomposed into irreducibles.

14. We have $(\rho_2^t \uparrow_{T_2} G)^{[(1,2),2]} = (\rho_2 \uparrow_{T_2} G)$, and hence $D_4 = [\,(3,4), \ 4\,]$.

15. The irreducibles are sorted with respect to the degree by

$$\begin{aligned} P_4 \ = \ & [\,(2,16,11)(3,17,12,7,21)(4,18,13,8,22) \\ & (5,19,14,9,23)(6,20,15,10,24),24), \ 24\,]. \end{aligned}$$

After simplifications we get

$$\mu^A = 1_G \oplus \lambda \oplus (\rho_2 \uparrow_{T_2} G) \oplus (\rho_2 \uparrow_{T_2} G) \oplus \lambda \cdot \overline{\rho}_4 \oplus \lambda \cdot \overline{\rho}_4 \oplus \lambda \cdot \overline{\rho}_4 \oplus \overline{\rho}_4 \oplus \overline{\rho}_4 \oplus \overline{\rho}_4,$$

with

$$\begin{aligned} A \ = \ & [\,(2,22,19,21,11,14,18,17,9,5,3,23,20,10,13,7,15,16,8,4) \\ & (6,24,12), \ 24\,]\cdot \\ & (\mathbf{1}_2 \otimes B)\cdot \\ & [\,(2,21,17,12,10,8,6,4)(3,24,20,15,23,18,14,22,19,16,13,11,9,7,5), \\ & (1,1,1,1,1,1,1,1,1,1,1,-1,1,1,-1,1,1,-1,1,1,1,1,1,1)\,]\cdot \\ & ((\mathrm{DFT}_2 \otimes \mathbf{1}_{10}) \oplus \mathbf{1}_4) \cdot \\ & [\,(2,16,11)(3,17,12,7,21)(4,18,13,8,22)(5,19,14,9,23) \\ & (6,20,15,10,24), \ 24\,]. \end{aligned}$$

The entire decomposition of μ using the implemented function `DecompositionMonRep` (cf. Appendix) takes 4.5 s CPU time on a SUN Ultra-Sparc 150 MHz. ∎

3

Symmetry and Decomposition of Matrices

In this chapter we will introduce the term *symmetry* for matrices and explain how a decomposition of a matrix can be derived from its symmetry. In general, a symmetry of an $(m \times n)$-matrix M is a pair (ϕ_1, ϕ_2) of representations of the same group G such that M lies in the intertwining space $\mathsf{Int}(\phi_1, \phi_2)$ of ϕ_1 and ϕ_2, i.e.

$$\phi_1(g) \cdot M = M \cdot \phi_2(g) \text{ for all } g \in G.$$

We call G a *symmetry group* of M. In this general definition every matrix has arbitrary many symmetries. If, e.g. M is an invertible $(n \times n)$-matrix and ϕ any representation of degree n of a group G, then we have $M \in \mathsf{Int}(\phi, \phi^M)$ which means that M has the symmetry (ϕ, ϕ^M).

The determination of a symmetry of a matrix is the dual operator to the computation of the intertwining space (cf. 1.5): the intertwining space of a pair (ϕ_1, ϕ_2) of representations is precisely the set of matrices having (ϕ_1, ϕ_2) as a symmetry.

The particular types of symmetry which will be considered in the three sections of this chapter, arise by restricting conditions to the representations ϕ_1 and ϕ_2. E.g. the *perm-perm-symmetry* (cf. Section 3.2) requires that ϕ_1 and ϕ_2 both are permutation representations. If, e.g.

$$M = \begin{bmatrix} a & b \\ b & a \end{bmatrix}, \ a, b \in \mathbb{K},$$

and

$$\mathsf{Z}_2 = \langle x \mid x^2 = 1 \rangle, \ \phi : \ x \mapsto [(1,2), 2],$$

then $M \in \mathsf{Int}(\phi, \phi)$, i.e. M has the symmetry (ϕ, ϕ). Obviously, perm-perm-symmetry means the existence of pairs (σ_1, σ_2) of permutations, such that permuting the rows of M by σ_1 yields the same result as permuting the columns of M by σ_2. Thus, if a matrix contains pairwise different entries then it has only the trivial perm-perm-symmetry.

The decomposition of a matrix using its symmetry is independent of the type of the symmetry and will be explained now. Let $M \in \mathbb{K}^{m\times n}$ be a matrix with symmetry (ϕ_1, ϕ_2), i.e. $M \in \mathsf{Int}(\phi_1, \phi_2)$. First we determine two decomposition matrices A_1, A_2 of ϕ_1 resp. ϕ_2. Then $\phi_1^{A_1} = \rho_1$ as well as $\phi_2^{A_2} = \rho_2$ are a direct sum of irreducibles. It follows that the matrix $D = A_1^{-1} \cdot M \cdot A_2 \in \mathsf{Int}(\rho_1, \rho_2)$ lies in the intertwining space of two entirely decomposed representations (cf. Figure 3.1).

$$\begin{tikzcd} \phi_1 \arrow[r, "M"] \arrow[d, "A_1"] & \phi_2 \arrow[d, "A_2"] \\ \rho_1 \arrow[r, "D"] & \rho_2 \end{tikzcd}$$

Figure 3.1: Decomposition of a matrix with symmetry (ϕ_1, ϕ_2)

By Theorem 1.48 D is a block permuted matrix. Solving for M yields the decomposition

$$M = A_1 \cdot D \cdot A_2^{-1}.$$

Decomposing different matrices with equal symmetry merely leads to different block permuted matrices D.

The matrices A_1, A_2^{-1} in this decomposition, however, are in general neither sparse nor structured in any respect. Furthermore, decomposition matrices for arbitrary representations are hard to compute. The motivation for considering symmetries where ϕ_1 and ϕ_2 are permutation resp. monomial representation lies in the fact that in these cases, A_1 and A_2 can be obtained as a product of sparse, even highly structured, matrices using Algorithm 2.2. More precise, these matrices are essentially composed from monomial matrices and DFT's of small sizes using the operators $\cdot$ (product), $\oplus$ and $\otimes$. In this case the decomposition $M = A_1 \cdot D \cdot A_2^{-1}$ hence can be used as a fast algorithm for the multiplication with M. The applications, however, are beyond that. Using known theorems on the behavior of functions like determinant, eigenvalues and inverse, a decomposed matrix (in our sense) can easily be handled in many respects.

Intuitively speaking, symmetry of one of the types considered in this chapter, captures part of the redundancy contained in the matrix. The decomposition of the symmetry allows to turn this redundancy into a decomposition of the matrix.

In order to automatically decompose a matrix M we hence have to perform the following two steps:

1. Determine the symmetry of an appropriate type of M.

2. Decompose M using this symmetry.

The first point was object of the Dissertation of Egner (1997), [26], in the framework of which also algorithms for finding certain types of symmetry had been implemented. Together with the implemented algorithms of this dissertation a program has been developed in the programming language GAP [57], which is able to generate automatically such matrix decompositions.

The three sections in this chapter correspond to the three considered types of symmetry: perm-irred-symmetry in Section 3.1, perm-perm-symmetry in Section 3.2, and finally mon-mon-symmetry in Section 3.3. In each case the type of symmetry is defined and the algorithm for decomposing a matrix by this symmetry is explained. The existing algorithms for finding perm-irred-symmetry and perm-perm-symmetry are presented. In Section 3.3 we will derive an algorithm allowing to compute a large class of mon-mon-symmetry.

For a more comprehensive description of the symmetries and algorithms we refer to the Dissertation of Egner (1997), [26]. The idea for the decomposition of a matrix with symmetry is due to Minkwitz (1993), [45], where already perm-irred-symmetry and perm-perm-symmetry are used to decompose matrices. Examples for decompositions can be found in Chapter 4. For the notation see Section 1.1. We assume the Maschke condition, as always in this work, to be satisfied.

3.1 Perm-Irred-Symmetry

3.1 Definition *Let $M \in \mathsf{GL}_n(\mathbb{K})$. A pair of representations (ϕ, ρ) of the same group G is called "Perm-Irred-Symmetry" of M, if $M \in \mathsf{Int}(\phi, \rho)$, ϕ is a permutation representation, and*

$$\rho = (\rho_1 \oplus \ldots \oplus \rho_k)^{[\sigma,n]} \text{ for a } \sigma \in \mathsf{S}_n,\ \rho_i \text{ irreducible },\ i = 1 \ldots n,$$

is a permuted direct sum of irreducible representations of G.

The condition imposed on ρ also can be formulated as: it exists a permutation $\sigma \in \mathsf{S}_n$ such that $M \cdot [\sigma^{-1}, n]$ is a decomposition matrix for ϕ. Note that M must be invertible in order to have a perm-irred-symmetry. First some examples.

3.2 Example Let $\mathsf{Z}_3 = \langle x \mid x^3 = 1\rangle$, $\phi :\ x \mapsto [(1,2,3),3]$ be a regular representation of Z_3, and $\rho :\ x \mapsto \mathrm{diag}(1, \omega_3, \omega_3^2)$ the corresponding decomposition. Then the matrix DFT_3 has the perm-irred-symmetry (ϕ, ρ).

Consider $\mathsf{S}_3 = \langle x, y \mid x^3 = y^2 = 1, x^y = x^{-1}\rangle$, with representation of degree 3

$$\psi :\ x \mapsto [(1,2,3),3],\ y \mapsto [(2,3),3],$$

and the following two representation which are decomposed into irreducibles

$$\rho_1 :\ x \mapsto \begin{bmatrix} 1 & 0 & 0 \\ 0 & \omega_3 & 0 \\ 0 & 0 & \omega_3^2 \end{bmatrix},\ y \mapsto [(2,3),3],$$

and

$$\rho_2 :\ x \mapsto \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & -1 \end{bmatrix},\ y \mapsto [(2,3),3].$$

Then the matrix DFT_3 also has the perm-irred-symmetry (ψ, ρ_1). The matrix

$$\mathrm{SOR}_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

has the perm-irred-symmetry (ψ, ρ_2). The restriction $\phi = \psi \downarrow \mathsf{Z}_3$, however, is not decomposed into irreducibles by SOR_3! More general, every matrix DFT_n has the perm-irred-symmetry (ϕ, ρ) of the group $\mathsf{Z}_n = \langle x \mid x^n = 1\rangle$ where

$$\phi :\ x \mapsto [(1,\ldots,n),n],\ \rho :\ x \mapsto \mathrm{diag}(\omega_n^0, \ldots, \omega^{(n-1)}).$$

We have $\phi^{\mathrm{DFT}_n} = \rho$. ■

Now we want to consider the decomposition of a matrix with perm-irred-symmetry. Let $M \in \mathsf{GL}_n(\mathbb{K})$ with perm-irred-symmetry (ϕ, ρ), where ϕ and ρ are representations of G and

$$\rho = (\rho_1 \oplus \ldots \oplus \rho_k)^{[\sigma,n]},\ \rho_i \text{ irreducible},\ i = 1\ldots k,\ \sigma \in \mathsf{S}_n.$$

Let A_ϕ be a decomposition matrix for ϕ in the sense of Chapter 2, i.e.

$$\phi^{A_\phi} = \psi = \bigoplus_{i=1}^{\ell} \underbrace{(\psi_i \oplus \ldots \oplus \psi_i)}_{r_i}, \ \psi_i \text{ irreducible}, \ \psi_i \not\cong \psi_j \text{ for } i \neq j.$$

Now we have two decomposition matrices for the same representation ϕ, namely $M \cdot [\sigma^{-1}, n]$ and A_ϕ. Let B be the block permuted matrix satisfying

$$(\rho_1 \oplus \ldots \oplus \rho_k)^B = \bigoplus_{i=1}^{\ell} \underbrace{(\psi_i \oplus \ldots \oplus \psi_i)}_{r_i},$$

i.e. B permutes the irreducibles in the first decomposition to be in the same order as the irreducibles in the second decomposition. Furthermore, equivalent irreducibles are conjugated to be equal. The blocks in B are maximal of size d_i^2, where $d_i = \deg(\rho_i)$. Altogether, B contains maximal $\sum_{i=1}^{k} r_i d_i^2$ entries $\neq 0$. Now the representation ϕ is decomposed identically by the matrices $M \cdot [\sigma^{-1}, n] \cdot B$ and A_ϕ (cf. Figure 3.2) which implies that

$$A_\phi^{-1} \cdot M \cdot [\sigma^{-1}, n] \cdot B = D \in \mathsf{Int}(\psi, \psi).$$

$$\begin{tikzcd}
& \psi \arrow[dd, "D"] \\
\phi \arrow[ur, "A_\phi"] \arrow[dr, "M\cdot[\sigma^{-1},n]\cdot B"'] & \\
& \psi
\end{tikzcd}$$

Figure 3.2: Decomposition via perm-irred-symmetry

By Theorem 1.48, iv) we get

$$\mathsf{Int}(\psi, \psi) = \bigoplus_{i=1}^{\ell} (\mathbb{K}^{r_i \times r_i} \otimes \mathbf{1}_{d_i}),$$

i.e. the matrix D contains maximal $\sum_{i=1}^{\ell} r_i^2 d_i$ entries $\neq 0$. Solving for M yields the decomposition of M:

$$M = A_\phi \cdot D \cdot B^{-1} \cdot [\sigma, n].$$

Altogether we obtain the following algorithm for the decomposition of a matrix using its perm-irred-symmetry:

3.3 Algorithm (Decomposition of a Matrix with Perm-Irred-Symmetry) Given is a matrix $M \in \mathsf{GL}_n(\mathbb{K})$ with perm-irred-symmetry (ϕ, ρ), $\phi^M = \rho^{[\sigma,n]} = (\bigoplus_{i=1}^k \rho_i)^{[\sigma,n]}$. M shall be decomposed into a product of sparse matrices.

1. Determine a decomposition matrix A_ϕ using Algorithm 2.2. We get

$$\phi^{A_\phi} = \psi = \bigoplus_{i=1}^{\ell} \underbrace{(\psi_i \oplus \ldots \oplus \psi_i)}_{r_i}, \ \psi_i \text{ irreducible}, \ \psi_i \not\cong \psi_j \text{ for } i \neq j.$$

2. Compute a block permuted matrix B with

$$(\rho_1 \oplus \ldots \oplus \rho_k)^B = \bigoplus_{i=1}^{\ell} \underbrace{(\psi_i \oplus \ldots \oplus \psi_i)}_{r_i},$$

   For this purpose, k matrices from the intertwining spaces $\mathsf{Int}(\rho_i, \psi_j)$ has to be computed. This is done according to Theorem 1.56 by solving systems of linear equations.

3. Determine a matrix D by matrix multiplication

$$D = A_\phi^{-1} \cdot M \cdot [\sigma^{-1}, n] \cdot B.$$

   Note that A_ϕ is, as a result of Algorithm 2.2, decomposed which makes inverting an easier task. The matrix multiplication above, however, is expensive.

We get

$$M = A_\phi \cdot D \cdot B^{-1} \cdot [\sigma, n].$$

∎

Note that the matrix A_ϕ above itself is decomposed.

Finding perm-irred-symmetry is one topic in Egner (1997), [26]. The algorithm, which is presented there, is exponential in n if all perm-irred-symmetries shall be found. Restricting the block sizes of a possible symmetry to k, however, makes the cost polynomial, more precisely given as $O(n^{2k+1})$ arithmetic operations in the base field $\mathbb{K}$. For $k = 3$ this makes the computation of the perm-irred-symmetry feasible up to a matrix size of about 20, for $k = 2$ up to a matrix size of about 50. Among the symmetry types presented in this chapter, the perm-irred-symmetry is by far the most expensive to compute.

Of course, it is possible to also define “mon-irred-symmetry” and decompose a matrix with this symmetry entirely analogous to the description above. We refrain from doing so since currently it is not possible to find symmetry of this type.

3.2 Perm-Perm-Symmetry

3.4 Definition *Let $M \in \mathbb{K}^{m \times n}$. A pair of permutation representations (ϕ, ψ) of the same group G is called "perm-perm-symmetry" of M, if $M \in \mathsf{Int}(\phi, \psi)$.*

In contrast to the perm-irred-symmetry, here M must be neither invertible nor quadratic. Furthermore, with (ϕ, ψ) also $(\phi \downarrow H, \psi \downarrow H)$ is a perm-perm-symmetry of M and obviously it exists a *maximal* perm-perm-symmetry containing every other.

A matrix M has a perm-perm-symmetry if and only if pairs (σ_1, σ_2) of permutations exist, such that permuting the rows of M with σ_1 yields the same result as permuting the columns with σ_2. Hence, for a perm-perm-symmetry it is important whether certain entries in the matrix are equal or not.

3.5 Example Let $\mathsf{Z}_3 = \langle x \mid x^3 = 1 \rangle$ with representation $\phi : \ x \mapsto [(1, 2, 3), 3]$. Then the matrix

$$M = \begin{bmatrix} a & b & c \\ c & a & b \\ b & c & a \end{bmatrix}, \ a, b, c \in \mathbb{K},$$

has the perm-perm-symmetry (ϕ, ϕ). More general, every circulant matrix

$$M_n = [x_{(j-i) \bmod n} \mid i, j \in \{0, \ldots, n-1\}], \ x_k \in \mathbb{K}, \ k = 1 \ldots n,$$

has the perm-perm-symmetry (ψ, ψ), where $\psi : \ x \mapsto [(1, \ldots, n), n]$ denotes the regular representation of $\mathsf{Z}_n = \langle x \mid x^n = 1 \rangle$.

For another example we consider the dihedral group with 8 elements, $\mathsf{D}_8 = \langle x, y \mid x^4 = y^2 = 1, \ x^y = x^{-1} \rangle$, with natural representation on 4 points

$$\phi : \ x \mapsto [(1, 2, 3, 4), 4], \ y \mapsto [(2, 4), 4],$$

and regular representation on 8 points

$$\psi : \ x \mapsto [(1, 4, 6, 7)(2, 3, 5, 8), 8], \ y \mapsto [(1, 2)(3, 7)(4, 8)(5, 6), 8].$$

The matrix

$$M = \begin{bmatrix} a & a & b & c & d & d & b & c \\ b & c & a & a & b & c & d & d \\ d & d & c & b & a & a & c & b \\ c & b & d & d & c & b & a & a \end{bmatrix}, \ a, b, c, d \in \mathbb{K},$$

has the perm-perm-symmetry (ϕ, ψ).

The DFT_n also has a perm-perm-symmetry. Let us consider the case $n = 5$. Let $\mathsf{Z}_4 = \langle x \mid x^4 = 1\rangle$ with representations

$$\phi : \; x \mapsto [(2,3,5,4),5], \;\; \psi : \; x \mapsto [(2,4,5,3),5].$$

Note that ϕ and ψ both are not transitive. DFT_5 has the (maximal) perm-perm-symmetry (ϕ, ψ). ■

Now we will explain how to decompose a matrix M with perm-perm-symmetry. Let A_ϕ and A_ψ be decomposition matrices of ϕ resp. ψ in the sense of Chapter 2, i.e.

$$\phi^{A_\phi} = \rho = \bigoplus_{i=1}^{k} \underbrace{(\rho_i \oplus \ldots \oplus \rho_i)}_{r_i}, \;\; \rho_i \text{ irreducible}, \;\; \rho_i \not\cong \rho_j \text{ for } i \neq j.$$

and

$$\psi^{A_\psi} = \theta = \bigoplus_{i=1}^{\ell} \underbrace{(\theta_i \oplus \ldots \oplus \theta_i)}_{s_i}, \;\; \theta_i \text{ irreducible}, \;\; \theta_i \not\cong \theta_j \text{ for } i \neq j.$$

Some irreducibles ρ_i resp. θ_j may occur (up to equivalence) in both decompositions, some don't. Permitting the value 0 for r_i resp. s_i, we can achieve that each ρ_i is equivalent to a certain θ_j and vice-versa. Thus the representations ρ and θ can be written as follows:

$$\rho = \bigoplus_{i=1}^{m} \underbrace{(\rho_i \oplus \ldots \oplus \rho_i)}_{r_i}, \;\; \theta = \bigoplus_{i=1}^{m} \underbrace{(\theta_i \oplus \ldots \oplus \theta_i)}_{s_i}.$$

In both cases the summation is up to a number $m \leq k + \ell$. Let B be the block permuted matrix satisfying

$$\theta^B = \bigoplus_{i=1}^{m} \underbrace{(\rho_i \oplus \ldots \oplus \rho_i)}_{s_i},$$

i.e. B permutes the irreducibles θ_i to be in the same order as the ρ_j and establishes equality. The blocks in B corresponds to the degrees d_i of the irreducibles ρ_i, i.e. B contains maximal $\sum_{i=1}^m s_i d_i^2$ entries $\neq 0$. It follows that (cf. Figure 3.3)

$$A_\phi^{-1} \cdot M \cdot A_\psi \cdot B = D \in \mathsf{Int}(\rho, \theta^B).$$

By Theorem 1.48, iv) we get

$$\mathsf{Int}(\rho, \theta^B) = \bigoplus_{i=1}^{m} (\mathbb{K}^{r_i \times s_i} \otimes \mathbf{1}_{d_i}),$$

$$\begin{tikzcd} \phi \arrow[r, "M"] \arrow[d, "A_\phi"] & \psi \arrow[d, "A_\psi \cdot B"] \\ \rho \arrow[r, "D"] & \theta^B \end{tikzcd}$$

Figure 3.3: Decomposition via perm-perm-symmetry

i.e. the matrix D contains maximal $\sum_{i=1}^{m} r_i s_i d_i$ entries $\neq 0$. Solving for M yields the decomposition

$$M = A_\phi \cdot D \cdot B^{-1} \cdot A_\psi^{-1}.$$

Altogether we obtain the following algorithm for the decomposition of a matrix using its perm-perm-symmetry:

3.6 Algorithm (Decomposition of a Matrix with Perm-Perm-Symmetry)
Given is a matrix $M \in \mathbb{K}^{m \times n}$ with perm-perm-symmetry (ϕ, ψ). M shall be decomposed into a product of sparse matrices.

1. Determine decomposition matrices A_ϕ, A_ψ of ϕ resp. ψ. We get

$$\phi^{A_\phi} = \rho = \bigoplus_{i=1}^{m} \underbrace{(\rho_i \oplus \ldots \oplus \rho_i)}_{r_i},\ \rho_i \text{ irreducible},\ \rho_i \ncong \rho_j,\ i \neq j,\ r_i \geq 0,$$

$$\psi^{A_\psi} = \theta = \bigoplus_{i=1}^{m} \underbrace{(\theta_i \oplus \ldots \oplus \theta_i)}_{s_i},\ \theta_i \text{ irreducible},\ \theta_i \ncong \theta_j,\ i \neq j,\ s_i \geq 0.$$

   Every ρ_i is equivalent to a θ_j and vice-versa.

2. Compute the block permuted matrix B with

$$\theta^B = \bigoplus_{i=1}^{m} \underbrace{(\rho_i \oplus \ldots \oplus \rho_i)}_{s_i}.$$

   For this purpose, for any θ_i which also appears (up to equivalence) as ρ_j, a matrix $\in \mathsf{Int}(\theta_i, \rho_j)$ has to be computed. This is done according to Theorem 1.56 by solving systems of linear equations. Maximal m such systems has to be solved.

3. Determine a matrix D through

$$D = A_\phi^{-1} \cdot M \cdot A_\psi \cdot B.$$

We get

$$M = A_\phi \cdot D \cdot B^{-1} \cdot A_\psi^{-1}.$$

∎

Note that the matrices A_ϕ and A_ψ^{-1} above itself are decomposed, too.

Finding perm-perm-symmetry is the topic of Leon (1991), [39]. The algorithm which is presented there is a special application of the partition based backtrack search developed by Leon. Also created by Leon, a very efficient C-implementation exists. The program is capable of finding the perm-perm-symmetry of (100×100)-matrices in a few seconds. The matrix, however, may contain only 256 different entries. Another implementation has been made by Egner in the language GAP without this restriction. In [26] it is shown that the problem of finding perm-perm-symmetry is not simpler than the problem of finding graph automorphisms.

3.3 Mon-Mon-Symmetry

3.7 Definition *Let $M \in \mathbb{K}^{m \times n}$. A pair of monomial representations (ϕ, ψ) of the same (not necessarily finite) group G is called "mon-mon-symmetry", if $M \in \mathsf{Int}(\phi, \psi)$.*

Hence the mon-mon-symmetry is a generalization of the perm-perm-symmetry. With (ϕ, ψ) obviously every pair $(\phi \downarrow H, \psi \downarrow H)$ is a mon-mon-symmetry of M, too, and it exists a *maximal* mon-mon-symmetry containing each other. If $M \in \mathbb{K}^{m \times n}$ and $\alpha \in \mathbb{K}$, then

$$\alpha \cdot \mathbf{1}_m \cdot M = M \cdot \alpha \cdot \mathbf{1}_n,$$

i.e. the maximal mon-mon-symmetry always contains $|\mathbb{K}|$ many scalar matrices from the center $Z(\mathsf{GL}_m(\mathbb{K}))$ resp. $Z(\mathsf{GL}_n(\mathbb{K}))$. These pose no restricting condition to M and hence are not of interest for purpose of decomposition. The mon-mon-symmetry consisting of the scalar matrices will be denoted in the following as trivial mon-mon-symmetry. In contrast to the perm-perm-symmetry, the mon-mon-symmetry of a matrix can well be irreducible (cf. Example 3.8). In this case, the symmetry has to be restricted to a subgroup in order to obtain a symmetry suitable for decomposition of the matrix. How to choose this subgroup is an open problem.

3.8 Example The mon-mon-symmetry of the DFT_n is obtained by collecting its perm-irred-symmetry and its perm-perm-symmetry. Since DFT_n is symmetric, every mon-mon-symmetry (ϕ, ψ) gives rise to another mon-mon-symmetry (ψ^T, ϕ^T) by transposition. We consider the case $n = 3$. The DFT(3) has the perm-perm-symmetry

$$[(2,3),3] \cdot \mathrm{DFT}_3 = \mathrm{DFT}_3 \cdot [(2,3),3]$$

and the perm-irred-symmetry

$$[(1,2,3),3] \cdot \mathrm{DFT}_3 = \mathrm{DFT}_3 \cdot \mathrm{diag}(1, \omega_3, \omega_3^2)$$

resp.

$$\mathrm{diag}(1, \omega_3, \omega_3^2) \cdot \mathrm{DFT}_3 = \mathrm{DFT}_3 \cdot [(1,3,2),3].$$

Altogether we get the symmetry group

$$\begin{aligned} G &= (\langle w \rangle \ltimes \langle x \rangle) \ltimes (\langle y \rangle \times \langle z \rangle) \\ &\cong (\mathsf{Z}_2 \ltimes \mathsf{Z}_3) \ltimes (\mathsf{Z}_3 \times \mathsf{Z}_3) \end{aligned}$$

of size 54 with representations

$$\phi: \; w \mapsto [(2,3),3], \; x \mapsto [(1,2,3),3], \; y \mapsto \mathrm{diag}(1, \omega_3, \omega_3^2), \; z \mapsto \mathrm{diag}(\omega_3, \omega_3, \omega_3)$$

and

$$\psi: \; w \mapsto [(2,3),3], \; x \mapsto \mathrm{diag}(1, \omega_3, \omega_3^2), \; y \mapsto [(1,3,2),3], \; z \mapsto \mathrm{diag}(\omega_3, \omega_3, \omega_3).$$

DFT_3 has the mon-mon-symmetry (ϕ, ψ). Both representations are irreducible.

For another example we consider $\mathsf{S}_4 = \langle x, y \mid x^4 = y^2 = (xy)^3 = 1 \rangle$ with the two faithful representations

$$\begin{aligned} \phi : x &\mapsto [(1,6,7,3)(2,5,8,4), \; (\omega_3, \omega_3^2, \omega_3, \omega_3^2, \omega_3, \omega_3^2, \omega_3^2, \omega_3)], \\ y &\mapsto [(1,6)(2,5)(3,4)(7,8), \; (\omega_3^2, \omega_3, \omega_3^2, \omega_3, \omega_3^2, \omega_3, \omega_3^2, \omega_3)], \\ \psi : x &\mapsto [(2,5,4,3), \; (\omega_4, \omega_4, -1, -\omega_4, -1, -\omega_4)], \\ y &\mapsto [(1,4)(2,6)(3,5), \; (\omega_4, \omega_4, -1, -\omega_4, -1, -\omega_4)]. \end{aligned}$$

The matrix

$$M = \begin{bmatrix} a & b & \omega_3^2 b & \omega_3^2 a & \omega_3 a & \omega_3 b \\ b & a & \omega_3 a & \omega_3 b & \omega_3^2 b & \omega_3^2 a \\ -\omega_{12}^7 a & -\omega_3^2 a & -\omega_{12}^7 b & -b & \omega_4 a & \omega_{12}^{11} b \\ -\omega_{12}^{11} b & -\omega_3 b & -\omega_{12}^{11} a & -a & \omega_4 b & \omega_{12}^7 a \\ \omega_{12}^{11} b & \omega_4 a & -b & -\omega_{12}^7 b & -\omega_3^2 a & -\omega_{12}^7 a \\ \omega_{12}^7 a & \omega_4 b & -a & -\omega_{12}^{11} a & -\omega_3 b & -\omega_{12}^{11} b \\ -\omega_3^2 a & -\omega_{12}^7 a & \omega_4 a & \omega_{12}^{11} b & -\omega_{12}^7 b & -b \\ -\omega_3 b & -\omega_{12}^{11} b & \omega_4 b & \omega_{12}^7 a & -\omega_{12}^{11} a & -a \end{bmatrix}, \quad a, b \in \mathbb{C},$$

has the mon-mon-symmetry (ϕ, ψ). ∎

The decomposition of a matrix with respect to its mon-mon-symmetry is entirely analogous to the decomposition with respect to the perm-perm-symmetry. The corresponding algorithm is obtained by simply replacing the word "perm-perm-symmetry" by "mon-mon-symmetry" in Algorithm 3.6.

Since the set of all monomial matrices of a given size is infinite (if the base field is), finding mon-mon-symmetry is more difficult than finding perm-perm-symmetry. The algorithm, which we are going to develop now, is able to find mon-mon-symmetry of a certain type, i.e. the set of all monomial matrices is restricted. The determination of the mon-mon-symmetry then is translated to finding the perm-perm-symmetry of a larger matrix.

3.9 Definition *Let $M \in \mathbb{K}^{m \times n}$ and $k \geq 1$. A pair (ϕ, ψ) of representations of the same group G is called "mon-mon-symmetry of order k", if (ϕ, ψ) is a mon-mon-symmetry of M and the entries $\neq 0$ in all matrices $\phi(g), \psi(g)$ are kth roots of unity.*

According to this definition, the perm-perm-symmetry is exactly the mon-mon-symmetry of order 1. This restriction reduces the monomial matrices to the finite set of those with kth roots of unity as entries $\neq 0$. If, e.g. $\mathbb{K} = \mathbb{F}_q$ is a finite field, then every mon-mon-symmetry is of order $q-1$. In the following we will develop an algorithm which finds the maximal mon-mon-symmetry of order k of a matrix.

The idea for this stems from Leon (1991), [39]. The matrix M is coded to a matrix $\mathcal{C}_k(M)$ such that every pair of monomial matrices of the mon-mon-symmetry of M corresponds to a pair of permutations of the perm-perm-symmetry of $\mathcal{C}_k(M)$. The parameter for the coding is k.

3.10 Definition *Let $a \in \mathbb{K}$ and ω_k a kth root of unity in $\mathbb{K}$. We call*

$$\begin{aligned}
\mathcal{C}_k(a) &= \begin{bmatrix} a \cdot \omega_k^0 & a \cdot \omega_k^1 & \cdots & a \cdot \omega_k^{(k-1)} \\ a \cdot \omega_k^1 & a \cdot \omega_k^2 & \cdots & a \cdot \omega_k^0 \\ \vdots & \vdots & \ddots & \vdots \\ a \cdot \omega_k^{(k-1)} & a \cdot \omega_k^0 & \cdots & a \cdot \omega_k^{(k-2)} \end{bmatrix} \\
&= [a \cdot \omega_k^{(i-1)+(j-1)} \mid i, j \in \{1, \ldots, k\}]
\end{aligned}$$

the coding of a with parameter k. If $A \in \mathbb{K}^{m\times n}$, then we analogously call

$$\begin{aligned}\mathcal{C}_k(A) &= \begin{bmatrix} \mathcal{C}_k(a_{1,1}) & \cdots & \mathcal{C}_k(a_{1,n}) \\ \vdots & \ddots & \vdots \\ \mathcal{C}_k(a_{m,1}) & \cdots & \mathcal{C}_k(a_{m,n}) \end{bmatrix} \\ &= [\mathcal{C}_k(a_{i,j}) \mid i \in \{1,\ldots,m\},\ j \in \{1,\ldots,n\}]\end{aligned}$$

the coding of A with parameter k.

Obviously, the coding described above is a bijective mapping. The parameter k corresponds, as we will see later, to the order of the monomial symmetry that can be found using this coding. If $a \in \mathbb{K}$, then the $(i+1)$th row of $\mathcal{C}_k(a)$ is obtained from the ith by multiplication with ω_k. An analogous observation holds for the columns and we get the following lemma:

3.11 Lemma It is

$$\mathcal{C}_n(a \cdot \omega_k^\ell) = \mathcal{C}_n(a) \cdot [(1,\ldots,k)^{-\ell}, k] = [(1,\ldots,k)^\ell, k] \cdot \mathcal{C}_n(a).$$

Proof Trivial. ∎

As suggested above, monomial matrices appearing in the mon-mon-symmetry of $M \in \mathbb{K}^{m\times n}$ shall correspond to block permutations in the perm-perm-symmetry of $\mathcal{C}_k(M) \in \mathbb{K}^{km\times kn}$. Such a block permutation $\sigma \in \mathsf{S}_{km}$ resp. $\in \mathsf{S}_{kn}$ permutes blocks of size k and the restriction of σ to each block has the form $(1,\ldots,k)^i$. Here an example for $k=2$, the block structure is enhanced:

$$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix} \longleftrightarrow \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right] = [\sigma, 6],\ \ \sigma = (1,4,6)(2,3,5).$$

The permutation matrix $[\sigma, 6]$ can be decomposed using its block structure:

$$\begin{aligned}[\sigma,6] &= ([(1,2),2] \oplus [(1),2] \oplus [(1,2),2]) \cdot ([(1,2,3),3] \otimes [(1),2]) \\ &= ([(1,2,3),3] \otimes [(1),2]) \cdot ([(1,2),2] \oplus [(1,2),2]) \oplus [(1),2]) .\end{aligned}$$

In general, the considered block permutations $\sigma \in \mathsf{S}_{kn}$ have the following structure:

$$[\sigma, kn] = ([\tau, n] \otimes \mathbf{1}_k) \cdot \left(\bigoplus_{i=1}^{n} [\sigma_i, k] \right),$$

where $\tau \in \mathsf{S}_n$ represents the macro permutation and $\sigma_i \in \mathsf{Z}_k = \langle (1, \ldots, k) \rangle$ is a k-cycle. Algebraically, the group of all these block permutations $\sigma \in \mathsf{S}_{kn}$ is the wreath product $\mathsf{Z}_k \wr \mathsf{S}_n$. For an explanation of wreath products we refer to the book of James/Kerber (1981), [37].

3.12 Lemma Let $\tau \in \mathsf{S}_n$ and $\sigma_i \in \mathsf{S}_k,\ i = 1 \ldots n$. Then

$$([\tau, n] \otimes \mathbf{1}_k) \cdot \left(\bigoplus_{i=1}^{n} [\sigma_i, k] \right) = \left(\bigoplus_{i=1}^{n} [\sigma_{i^\tau}, k] \right) \cdot ([\tau, n] \otimes \mathbf{1}_k).$$

Proof Straightforward computation. ∎

Now we will prove that coding of a matrix M allows to find the maximal mon-mon-symmetry of order k.

Before this we want to recall the used notation. A monomial $(m \times m)$-matrix is represented as $[\sigma, (s_1, \ldots, s_m)] = [\sigma, m] \cdot \operatorname{diag}(s_1, \ldots, s_m)$, $\sigma \in \mathsf{S}_m,\ s_i \in \mathbb{K},\ i = 1 \ldots n$. If $A = [a_{i,j}] \in \mathbb{K}^{m \times n}$, $\sigma \in \mathsf{S}_m,\ \tau \in \mathsf{S}_n$, then the following rules hold:

1. $[\sigma, m] \cdot [a_{i,j}] = [a_{i^\sigma, j}]$.
2. $[a_{i,j}] \cdot [\tau, n] = [a_{i, j^{\tau^{-1}}}]$.
3. $[a_{i,j}] \cdot \operatorname{diag}(s_1, \ldots, s_n) = [s_j \cdot a_{i,j}]$.
4. $\operatorname{diag}(s_1, \ldots, s_n) \cdot [a_{i,j}] = [s_i \cdot a_{i,j}]$.
5. $[\sigma, m] \cdot \operatorname{diag}(s_1, \ldots, s_m) = \operatorname{diag}(s_{1^\sigma}, \ldots, s_{m^\sigma}) \cdot [\sigma, m]$.

3.13 Theorem *Let $A \in \mathbb{K}^{m \times n}$. The mon-mon-symmetry of order k of A can be computed by computing the perm-perm-symmetry of the matrix $\mathcal{C}_k(A)$. More precisely: If $s_i = \omega_k^{u_i},\ i = 1 \ldots m,\ t_j = \omega_k^{v_j},\ j = 1 \ldots n,$ then*

$$\begin{aligned} & [\sigma, (s_1, \ldots, s_m)] \cdot A = A \cdot [\tau, (t_1, \ldots, t_n)] \\ \Leftrightarrow\ & ([\sigma, m] \otimes \mathbf{1}_k) \cdot \bigoplus_{i=1}^{m} [(1, \ldots, k)^{u_i}, k] \cdot \mathcal{C}_k(A) = \\ & \mathcal{C}_k(A) \cdot \bigoplus_{j=1}^{n} [(1, \ldots, k)^{-v_j}, k] \cdot ([\tau, n] \otimes \mathbf{1}_k)\,. \end{aligned}$$

Proof Assume the pair $[\sigma,(s_1,\ldots,s_m)],\ [\tau,(t_1,\ldots,t_n)]$ occurs in the monomial symmetry of $A=[a_{i,j}]$, where $s_i=\omega_k^{u_i},\ t_j=\omega_k^{v_j}$, as defined in the theorem. Then

$$
\begin{aligned}
& [\sigma,(s_1,\ldots,s_m)]\cdot A = A\cdot[\tau,(t_1,\ldots,t_n)]\\
\Leftrightarrow\ & [s_{i^\sigma}\cdot a_{i^\sigma,j}] = [t_{j^{\tau^{-1}}}\cdot a_{i,j^{\tau^{-1}}}]\\
\Leftrightarrow\ & [\omega_k^{u_{i^\sigma}}\cdot a_{i^\sigma,j}] = [\omega_k^{v_{j^{\tau^{-1}}}}\cdot a_{i,j^{\tau^{-1}}}]\\
\Leftrightarrow\ & \forall i,j:\ \mathcal{C}_k(\omega_k^{u_{i^\sigma}}\cdot a_{i^\sigma,j}) = \mathcal{C}_k(\omega_k^{v_{j^{\tau^{-1}}}}\cdot a_{i,j^{\tau^{-1}}})\\
& \text{(Lemma 3.11)}\\
\Leftrightarrow\ & \forall i,j:[(1,\ldots,k)^{u_{i^\sigma}},k]\cdot\mathcal{C}(a_{i^\sigma,j}) = \mathcal{C}_k(a_{i,j^{\tau^{-1}}})\cdot[(1,\ldots,k)^{-v_{j^{\tau^{-1}}}},k]\\
\Leftrightarrow\ & \bigoplus_{i=1}^{m}[(1,\ldots,k)^{u_{i^\sigma}},k]\cdot([\sigma,m]\otimes\mathbf{1}_k)\cdot\mathcal{C}_k(A) =\\
& \mathcal{C}_k(A)\cdot([\tau,n]\otimes\mathbf{1}_k)\cdot\bigoplus_{j=1}^{n}[(1,\ldots,k)^{-v_{j^{\tau^{-1}}}},k]\\
& \text{(Lemma 3.12)}\\
\Leftrightarrow\ & ([\sigma,m]\otimes\mathbf{1}_k)\cdot\bigoplus_{i=1}^{m}[(1,\ldots,k)^{u_i},k]\cdot\mathcal{C}_k(A) =\\
& \mathcal{C}_k(A)\cdot\bigoplus_{j=1}^{n}[(1,\ldots,k)^{-v_j},k]\cdot([\tau,n]\otimes\mathbf{1}_k)\,,
\end{aligned}
$$

as desired. ∎

The remaining problem is the appropriate choice of the parameter k. If $\mathbb{K}=\mathbb{F}_q$ is a finite field then $k=q-1$ can be chosen and even the maximal mon-mon-symmetry can be computed. In the case $\mathbb{K}\le\mathbb{C}$ a matrix definitely can have non-trivial mon-mon-symmetry of arbitrary order. If, e.g.

$$M=\begin{bmatrix} a & 0\\ 0 & b\end{bmatrix},\ a,b\in\mathbb{K},$$

then

$$\begin{bmatrix}\omega_n & 0\\ 0 & \omega_n^{-1}\end{bmatrix}\cdot M = M\cdot\begin{bmatrix}\omega_n & 0\\ 0 & \omega_n^{-1}\end{bmatrix}\quad\text{for all } n=1,2,\ldots\ .$$

If a matrix with complex entries has monomial symmetry of any order then the symmetry permutes entries with equal absolute values. Hence we consider quotients of entries of the same absolute value. This observation leads to the following procedure for the determination of the parameter k for the coding of the matrix. Let $A \in \mathbb{K}^{m\times n}$, $\mathbb{K} \leq \mathbb{C}$ and $M = \{a_1, \dots, a_r\}$ the set of all entries of A. We define the following equivalence relation on the entries of M:

$$a_i \sim a_j \Leftrightarrow |a_i| = |a_j|.$$

Let $P = \{P_1, \dots, P_\ell\}$ be the partition of M by "$\sim$" and $P_i = \{p_{i,1}, \dots, p_{i,r_i}\}$. For $i = 1 \dots \ell$ we define k_i as the lcm of the orders of the $r_i - 1$ roots of unity $p_{i,2}/p_{i,1}, \dots, p_{i,r_i}/p_{i,1}$, as long as $0 \notin P_i$ and the orders of these roots of unity are finite. Else we set $k_i = 1$. Then the parameter for the coding is $k = \mathrm{lcm}(k_i \mid i = 1 \dots \ell)$.

We obtain the following algorithm for finding the mon-mon-symmetry.

3.14 Algorithm (Computing Mon-Mon-Symmetry) Given is a matrix $M \in \mathbb{K}^{m\times n}$, $\mathbb{K} \leq \mathbb{C}$. A mon-mon-symmetry (ϕ, ψ) of M shall be computed where the entries $\neq$ in the images $\phi(g), \psi(g)$ all are roots of unity.

1. Determine the parameter k with the procedure described above.
2. Code M to $\mathcal{C}_k(M)$ with parameter k.
3. Compute the perm-perm-symmetry (ϕ_1, ψ_1) of $\mathcal{C}_k(M)$.
4. Restrict ϕ_1, ψ_1 (if necessary) to a subgroup H such that all images $\phi_1(h), \psi_1(h)$ are block permutations of the form described in Theorem 3.13.
5. Decode $\phi_1 \downarrow H, \psi_1 \downarrow H$ by Theorem 3.13 to ϕ, ψ.

M has the mon-mon-symmetry (ϕ, ψ). ∎

Concerning item 4. in Algorithm 3.14 we remark that in virtually all considered application we have $H = G$. However, a criterion to characterize the cases where this is not satisfied could not be derived.

4

Application To Signal Transforms

An application for the algorithms in Chapter 3 is the automatic generation of fast algorithms for discrete signal transforms. The decomposition of a matrix, as described there, can be used as a fast algorithm for matrix-vector resp. matrix-matrix multiplication.

In this chapter we will investigate some classical signal transforms for symmetry. It turns out that all considered transforms have a symmetry which gives rise to a substantial decomposition. Some examples of decompositions are given which are created by the implemented algorithms from Chapter 3. All decompositions has been computed in exactly the presented form. The computation time is in each case given as CPU-time on a SUN Ultra-Sparc 150 MHz.

Of course, the automatically generated decompositions does not provide optimal algorithms for the respective signal transform. The reason for this is that on the one side not every fast algorithm for a matrix-vector multiplication can be expressed as a product of sparse matrices, on the other side is fine-tuning beyond the capability of the presented methods. Furthermore is the efficiency of a certain fast algorithm very much dependent on the considered machine model. If we, however, count arithmetic operations in the base field, then the asymptotic complexity of the generated decompositions coincide with the asymptotic complexity of the known, fastest algorithms. This shows that representation theory not only provides, from a theoretical point of view, a satisfactory reason for the existence of a fast algorithm for a signal transform, it also provides a possibility to generate one!

Each of the following sections is considered with a certain series of signal transforms, which is investigated for symmetry of the considered types. In each case the size and the isomorphism type of the symmetry group is given as well as some properties of the corresponding representations. For the mon-mon-symmetry, in addition the order is supplied, i.e. the order of the roots of unity in the image matrices. For finding the perm-irred-symmetry and the perm-perm-symmetry the implemented algorithms of Egner has been used.

The results concerning the symmetry groups has been deduced by investigating the respective transform for small sizes and extrapolating the result. For the DFT, however, the symmetries are proven (cf. e.g. Egner, [26]).

For the notation we refer to Section 1.1. The definitions of the cosine transforms follow Mertins (1996), [44], the other transforms Elliott/Rao (1982), [28].

4.1 Discrete Fourier Transform

The discrete Fourier transform DFT_n of size n is defined as

$$\mathrm{DFT}_n = [\omega_n^{ij} \mid i, j \in \{0, \ldots, n-1\}].$$

The DFT_n is *the* transform in signal processing and is comprehensively examined in every standard book. Most of the numerous properties of this matrix are reflected in the symmetries given below.

The book of Beth (1984), [6] is considered with the algebraic interpretation of the Fourier transform and is the first essential work on the connection of signal transforms and representation theory of finite groups.

The perm-irred-symmetry and the perm-perm-symmetry are sufficiently known (cf. e.g. [26]). If $\mathsf{Z}_n = \langle x \mid x^n = 1 \rangle$ denotes the cyclic group of order n with regular representation

$$\phi: \ x \mapsto [(1, \ldots, n), n]$$

and corresponding decomposed representation

$$\rho: \ x \mapsto \mathrm{diag}(\omega_n^0, \omega_n^1, \ldots, \omega_n^{(n-1)}),$$

then the DFT_n has the perm-irred-symmetry (ϕ, ρ). If $\mathsf{Z}_n^\times = (\mathbb{Z}/n\mathbb{Z})^\times$, then

$$\phi_1: \ k \mapsto [i \mapsto ki \bmod n]$$

and

$$\phi_2 = \phi_1^{-1}: \ k \mapsto [i \mapsto k^{-1} i \bmod n]$$

defines two permutation representations of $\mathsf{Z}_n^\times$ on n points and the perm-perm-symmetry of DFT_n is given by (ϕ_1, ϕ_2).

symmetry type	group	size	properties
perm-irred (ϕ, ρ)	Z_n	n	ϕ is regular
perm-perm (ϕ_1, ϕ_2)	$\mathsf{Z}_n^\times$	$\varphi(n)$	ϕ_1, ϕ_2 not transitive
mon-mon (ϕ_1, ϕ_2) (order n)	$(\mathsf{Z}_n^\times \ltimes \mathsf{Z}_n) \ltimes (\mathsf{Z}_n \times \mathsf{Z}_n)$	$\varphi(n)n^3$	ϕ_1, ϕ_2 irreducible

Table 4.1: Symmetry of DFT_n

The mon-mon-symmetry of degree n arises by collecting both of the other symmetries and adding the trivial symmetry. We put the results together in Tabular 4.1. We denote by φ Euler's phi function ($\varphi(n)$ = number of primes residues modulo n).

As already mentioned, the Cooley-Tukey decomposition and the Good-Thomas decomposition of a DFT_n arise by using the perm-irred-symmetry. The algorithm of Rader (1968), [53] uses the perm-perm-symmetry in the case $n = p$, prime. The corresponding group in this case (cf. Tabular 4.1) is $\mathsf{Z}_p^\times \cong \mathsf{Z}_{p-1}$ isomorphic to a cyclic group of order $p-1$. Using this symmetry, the DFT_p essentially is decomposed into two $\mathrm{DFT}_{(p-1)}$'s.

It follows the automatic generated decomposition of DFT_5 using Algorithm 3.6, runtime: 0.2 s. Note that the asymptotic ($p \to \infty$) better complexity of this decomposition is not yet reflected for $p = 5$.

$$
\begin{aligned}
&\mathrm{DFT}_5 = \\
&\quad [(4,5),\ 5]\cdot \\
&\quad \Big(\mathbf{1}_1 \oplus (\mathrm{DFT}_2 \otimes \mathbf{1}_2) \cdot \mathrm{diag}(1,1,1,\omega_4) \cdot (\mathbf{1}_1 \otimes \mathrm{DFT}_2) \cdot [(2,3),\ 4]\Big)\cdot \\
&\quad \left(\begin{bmatrix} 1 & 4 \\ 1 & -1 \end{bmatrix} \oplus \mathrm{diag}(a,b,c)\right) \cdot \\
&\quad \Big(\mathbf{1}_1 \oplus \tfrac{1}{4} \cdot [(2,3),\ 4] \cdot (\mathbf{1}_2 \otimes \mathrm{DFT}_2) \cdot \mathrm{diag}(1,1,1,-\omega_4) \cdot (\mathrm{DFT}_2 \otimes \mathbf{1}_2)\Big) \cdot \\
&\quad [(3,4,5),\ 5],
\end{aligned}
$$

where

$$a = \omega_{20}^4 - \omega_{20}^{13} - \omega_{20}^{16} + \omega_{20}^{17},\ b = \omega_5 - \omega_5^2 - \omega_5^3 + \omega_5^4,\ c = \omega_{20}^4 + \omega_{20}^{13} - \omega_{20}^{16} - \omega_{20}^{17}.$$

The first and the last two rows in the decomposition correspond to the DFT_4, the matrix in the middle lies in the intertwining space of the decomposed representations.

4.2 Walsh-Hadamard Transform

The Walsh-Hadamard transform WHT_{2^k} is the k-fold tensor product of a DFT_2 with itself:

$$\mathrm{WHT}_{2^k} = \underbrace{\mathrm{DFT}_2 \otimes \ldots \otimes \mathrm{DFT}_2}_{k\text{-fold}}.$$

A comprehensive description of the WHT in signal processing provides the book of Beauchamp (1984), [4]. The significance of the WHT for Reed-Muller-Codes and other error-correcting codes is given in MacWilliams/Sloane (1992), [41]. For coding on quantum computers cf. Beth/Grassl (1996), [8].

A fast algorithm for WHT of course is implied in the definition above. If, however, the structure was not known, it could be found using the perm-irred-symmetry.

Analogous to the DFT, the mon-mon-symmetry of the WHT arises from gathering perm-irred-symmetry and perm-perm-symmetry.

symmetry type	group	size	properties
perm-irred (ϕ, ρ)	Z_2^k	2^k	ϕ is regular
perm-perm (ϕ_1, ϕ_2)	$\mathsf{GL}(k, 2)$	$(2^k - 2^{k-1}) \cdots (2^k - 1)$	ϕ_1, ϕ_2 not transitive
mon-mon (ϕ_1, ϕ_2)	G_k	$\lvert \mathsf{GL}(k, 2) \rvert \cdot 2^{2k+1}$	ϕ_1, ϕ_2 irreducible
(order 2)	$G_k \cong (\mathsf{GL}(k, 2) \ltimes \mathsf{Z}_2^k) \ltimes (\mathsf{Z}_2 \times \mathsf{Z}_2^k)$		

Table 4.2: Symmetry of WHT_{2^k}

4.3 Discrete Cosine Transform, Type I

The cosine transform DCT-I$_n$ is a $((n+1) \times (n+1))$-matrix defined by

$$\text{DCT-I}_n = \left[\sqrt{2/n} \cdot c_i c_j \cdot \cos\left(\frac{ij\pi}{n}\right) \mid i, j \in \{0, \ldots, n\} \right],$$

where

$$c_k = \begin{cases} 1/\sqrt{2} & \text{for } k = 0, n \\ 1 & \text{else} \end{cases}.$$

The DCT-I plays virtually no role in signal processing. For its symmetry cf. Tabular 4.3.

symmetry type	group	size	properties
perm-perm (ϕ_1, ϕ_2)	n even: $\mathsf{Z}_{2^{k-1}} \times \mathsf{Z}_m^\times$	$\varphi(n)$	ϕ_1, ϕ_2 intransitive
	$n = 2^k m, 2 \not| m$		
	n odd: G_n	$\varphi(n)/2$	
	with $(\mathsf{Z}_n^\times : G_n) = 2$		
mon-mon (ϕ_1, ϕ_2)	n even: $\mathsf{Z}_2^3 \times \mathsf{Z}_n^\times$	$8 \cdot \varphi(n)$	ϕ_1, ϕ_2 transitive
(order 2)	n odd: $\mathsf{D}_8 \times H_n$	$4 \cdot \varphi(n)$	$\Leftrightarrow n = 1, 3$
	with $(\mathsf{Z}_n^\times : H_n) = 2$		

Table 4.3: Symmetry of DCT-I_n

4.4 Discrete Cosine Transforms, Type II and III

The cosine transform DCT-III_n is defined by

$$\text{DCT-III}_n = \left[\sqrt{2/n} \cdot c_j \cdot \cos\left(\frac{(i+1/2)j\pi}{n}\right) \mid i, j \in \{0, \ldots, n-1\}\right],$$

where

$$c_j = \begin{cases} 1/\sqrt{2} & \text{for } j = 0 \\ 1 & \text{else} \end{cases}.$$

The DCT-II is the transpose of DCT-III. Hence the perm-perm-symmetry and the mon-mon-symmetry of both matrices correspond to each other. Only the DCT-III, however, has a perm-irred-symmetry, which has been discovered by Minkwitz in the framework of his dissertation [45].

The DCT-II is of great significance for image compression. It is used in the JPEG standard (cf. Pennebaker/Mitchell (1993), [51]). There it is used in the size 8×8, in the MPEG standard in the size 16×16. For the latter see the book of Rao/Hwang (1996), [54].

symmetry type	group	size	properties
perm-irred (ϕ, ρ)	D_{2n}	$2n$	ϕ is transitive
mon-mon (ϕ_1, ϕ_2)	$n = 2$: D_8	8	ϕ_1 is transitive
(order 2)	$n = 4$: $\mathsf{Z}_2 \ltimes (\mathsf{Z}_2 \times \mathsf{Z}_4)$	16	$\Leftrightarrow n \leq 2$,
	else: $\mathsf{Z}_2 \times \mathsf{Z}_{2^k} \times \mathsf{Z}_m^\times$,	$2^{(k+1)}\varphi(m)$	ϕ_2 is transitive
	with $n = 2^k m,\ 2 \not| m$		$\Leftrightarrow n = 2^k$

Table 4.4: Symmetry of DCT-III_n

Here a decomposition of DCT-III$_8$, generated by Algorithm 3.3 using the perm-irred-symmetry, runtime: 10.5s.

$$\begin{aligned}
\text{DCT-III}_8 = &\\
&[(1,2,6,8)(3,7,5,4),\ 8] \cdot (\mathbf{1}_2 \otimes (\mathbf{1}_2 \otimes \text{DFT}_2) \cdot [(2,3),\ 4] \cdot (\text{DFT}_2 \oplus \mathbf{1}_2)) \cdot \\
&[(2,7,6,8,5,4,3),\ 8] \cdot \left(\mathbf{1}_4 \oplus \tfrac{1}{\sqrt{2}} \cdot \text{DFT}_2 \oplus \mathbf{1}_2\right) \cdot [(5,6),\ 8] \cdot \\
&((\text{DFT}_2 \otimes \mathbf{1}_3) \oplus \mathbf{1}_2) \cdot [(2,7,3,8,4),\ 8] \cdot \\
&\tfrac{1}{4} \cdot \left(\text{diag}(\sqrt{2},\sqrt{2}) \oplus \begin{bmatrix} a & b \\ b & -a \end{bmatrix} \oplus \begin{bmatrix} c & -d \\ d & c \end{bmatrix} \oplus \begin{bmatrix} e & f \\ f & -e \end{bmatrix}\right),
\end{aligned}$$

where

$$\begin{aligned}
a &= \omega_{16} - \omega_{16}^7, \quad b = \omega_{16}^3 - \omega_{16}^5, \quad c = -\omega_{32} + \omega_{32}^{15}, \\
d &= \omega_{32}^7 - \omega_{32}^9, \quad e = -\omega_{32}^3 + \omega_{32}^{13}, \quad f = -\omega_{32}^5 + \omega_{32}^{11}.
\end{aligned}$$

The first three rows correspond to the decomposition matrix of the permutation representation of D_{16}, the last row is the block diagonal matrix from the intertwining space of the decomposed representations. A decomposition of DCT-II$_8$ can be obtained by transposing the decomposition above.

4.5 Discrete Cosine Transform, Type IV

The cosine transform DCT-IV$_n$ is defined by

$$\text{DCT-IV}_n = \left[\sqrt{2/n} \cdot \cos\left(\frac{(i+1/2)(j+1/2)\pi}{n}\right) \mid i,j \in \{0,\ldots,n-1\}\right].$$

The DCT-IV is of a certain importance as subroutine for fast algorithms for the DCT-II and for the calculation of so-called lapped orthogonal transforms (cf. Malvar (1992), [42]).

The DCT-IV merely has a mon-mon-symmetry which, however, leads to an efficient decomposition of the matrix.

symmetry type	group	size	properties
mon-mon (ϕ_1,ϕ_2) (order 2)	$\mathsf{Z}_{2^{(k+1)}} \times \mathsf{Z}_m^\times$ with $n = 2^k m,\ 2 \not| m$	$2^{(k+1)} m$	ϕ_1,ϕ_2 transitive $\Leftrightarrow n = 2^k$

Table 4.5: Symmetry ofDCT-IV$_n$

It follows a decomposition of the DCT-IV_8 generated using the mon-mon-symmetry, runtime: 1.4 s.

$$
\begin{aligned}
&\text{DCT-IV}_8 = \\
&\quad [(3,4,7,6,8,5),\ (\omega_4, \omega_{16}^5, \omega_8^3, -\omega_{16}^7, 1, -\omega_{16}, \omega_8, -\omega_{16}^3)] \cdot (\text{DFT}_2 \otimes \mathbf{1}_4) \cdot \\
&\quad \text{diag}(1,1,1,1,1,\omega_8,\omega_4,\omega_8^3) \cdot (\mathbf{1}_2 \otimes \text{DFT}_2 \otimes \mathbf{1}_2) \cdot \\
&\quad \text{diag}(1,1,1,\omega_4,1,1,1,\omega_4) \cdot (\mathbf{1}_4 \otimes \text{DFT}_2) \cdot \\
&\quad \text{diag}(-\omega_{64}, -\omega_{64}, \omega_{64}^9, -\omega_{64}^9, \omega_{64}^{23}, -\omega_{64}^{23}, \omega_{64}^{31}, \omega_{64}^{31}) \cdot \\
&\quad \left(\mathbf{1}_4 \otimes \tfrac{1}{2} \cdot \text{DFT}_2\right) \cdot \text{diag}(1,1,1,-\omega_4,1,1,1,-\omega_4) \cdot \\
&\quad \left(\mathbf{1}_2 \otimes \tfrac{1}{2} \cdot \text{DFT}_2 \otimes \mathbf{1}_2\right) \cdot \text{diag}(1,1,1,1,1,-\omega_8^3,-\omega_4,-\omega_8) \cdot \\
&\quad \left(\tfrac{1}{2} \cdot \text{DFT}_2 \otimes \mathbf{1}_4\right) \cdot [(2,6,3,4,7,5,8),\ (\omega_4, \omega_{16}^5, -\omega_{16}^7, \omega_8, \omega_8^3, -\omega_{16}^3, -\omega_{16}, 1)].
\end{aligned}
$$

The first and the last two rows correspond to the decomposition matrices of the monomial representations. The diagonal matrix in the middle is in the intertwining space of the decomposed representations.

4.6 Haar Transform

The Haar transform HT_{2^k} is defined recursively by:

$$
\text{HT}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \ \text{HT}_{2^{k+1}} = \begin{bmatrix} \text{HT}_{2^k} & \otimes [1 \quad 1] \\ 2^{k/2} \cdot \mathbf{1}_{2^k} & \otimes [1 \ -1] \end{bmatrix}, \ k > 1.
$$

The Haar transform can be considered as the first wavelet transform, cf. Haar (1910), [32]. Like most of the wavelet transforms it can be computed by its recursive definition in $O(2^k)$ additions and multiplications.

symmetry type	group	size	properties
mon-mon (ϕ_1, ϕ_2) (order 2)	?	2^{2^k+1}	ϕ_1 is transitive $\Leftrightarrow k = 1$ ϕ_2 is transitive

Table 4.6: Symmetry of HT_{2^k}

As shown in the tabular, the Haar transform has a symmetry group (with respect to the mon-mon-symmetry) of exponential size making it difficult to determine the isomorphism type. For $k = 1 \dots 4$ it turned out that the left part ϕ_1 of the symmetry is a permuted direct sum of irreducible representations, revealing that the transform even has a mon-irred-symmetry (more precise: irred-mon-symmetry).

A decomposition of the Haar transform is implied in its definition. E.g. for $k = 3$:

$$\mathrm{HT}_8 = \left(\mathrm{DFT}_2 \oplus \sqrt{2} \cdot \mathbf{1}_2 \oplus \mathbf{1}_4\right) \cdot \left(\begin{bmatrix} \mathbf{1}_2 \otimes [1 \quad 1] \\ \mathbf{1}_2 \otimes [1 \ -1] \end{bmatrix} \oplus 2 \cdot \mathbf{1}_4\right) \cdot \begin{bmatrix} \mathbf{1}_4 \otimes [1 \quad 1] \\ \mathbf{1}_4 \otimes [1 \ -1] \end{bmatrix}.$$

In this decomposition some tensoring with a (1×2)-matrix occurs, which cannot appear in an automatic generated decomposition (only square matrices appear, if a square matrix is decomposed). The third factor above, however, can be transformed into a tensor product $\mathbf{1}_4 \otimes \mathrm{DFT}_2$ by multiplication with permutation matrices. Transforming the decomposition above in this sense, yields essentially the decomposition generated by our algorithm as shown below, runtime: 20 s for a group size of 512.

$$\begin{aligned}
&\mathrm{HT}_8 = \\
&\quad [(1,8,6,4,2,7,5,3),\ 8] \cdot \\
&\quad \left(\mathrm{diag}\left(-\sqrt{2}, \sqrt{2}\right) \oplus \mathbf{1}_4 \oplus \mathrm{DFT}_2\right) \cdot \\
&\quad [(1,5,4,8,6,3,7,2),\ 8] \cdot \\
&\quad (\mathbf{1}_2 \otimes ([(1,2),\ 4] \cdot (\mathrm{DFT}_2 \oplus 2 \cdot \mathbf{1}_2) \cdot [(2,3),\ 4] \cdot (\mathbf{1}_2 \otimes \mathrm{DFT}_2))) \cdot \\
&\quad [(1,8,4,7)(3,6,2,5),\ (1,1,1,1,-1,-1,-1,-1)].
\end{aligned}$$

Appendix A

AREP – a Software Package for Constructive Representation Theory

In the framework of this dissertation, the software package AREP has been created for the calculation with matrix representations of finite groups. The used programming language is GAP, [57], Version 3.4.4. AREP will be available from the middle of 98 as a GAP share package.

The package is divided into three parts:

1. The category `AMat` (**A**bstract **Mat**rices) provides the data types and functions for the calculation with structured matrices.

2. Analogous provides `ARep` (**A**bstract **Rep**resentations) the data types and functions for the calculation with matrix representations of finite groups.

3. In the third part, higher functions for the calculation with representations are provided. These are based on the results from the Chapters 1 and 2.

It follows the correspondingly divided documentation.

```
The category AMat of structured matrices
========================================

The category AMat contains elements representing matrices
(rectangular matrices over GAP-fields) in a structured way.
In fact AMat is a term-algebra representing matrices. Basic
```

```
building blocks are literal matrices (Lists of Lists), monomial
matrices and permutation matrices (Permutations). Important
compositions are sums, products, direct sums (diag) and
tensor products.

We define the representation of AMat recursively in BNF as
the disjoint union of the following cases

<AMat> ::=
; atomic cases
    <perm>                                ; "perm" (invertible)
  | <mon>                                 ; "mon"  (invertible)
  | <mat>                                 ; "mat"

; composed cases
  | <scalar> * <AMat>                     ; "scalarMultiple"
  | <AMat> * .. * <AMat>                  ; "product"
  | <AMat> ^ <int>                        ; "power"
  | <AMat> ^ <AMat>                       ; "conjugate"
  | DirectSum(<AMat>, .., <AMat>)         ; "directSum"
  | TensorProduct(<AMat>, .., <AMat>)     ; "tensorProduct"
  | GaloisConjugate(<AMat>, <aut>)        ; "galoisConjugate".

An AMat is a GAP-Rec A with the following mandatory
fields common to all types of AMat

  isAMat       := true, identifies AMats
  operations   := AMatOps, GAP operations record for A
  type         : a string identifying the type of A
  dimensions   : size of the matrix represented
                 (= [rows, columns])
  char         : the characteristic of the field

The following fields are mandatory to special types

A.type = "perm"
  A.element      : defining permutation

A.type = "mon"
  A.element      : defining mon-object

A.type = "mat"
  A.element      : defining matrix
  A.isDFT        : optional field indicating that A is a DFT
```

```
  A.isSOR        : optional field indicating that A is a SOR

A.type = "scalarMultiple"
  A.element      : the AMat multiplied
  A.scalar       : the scalar

A.type = "product"
  A.factors      : List of AMats of suitable dimensions
                   and characteristic
A.type = "power"
  A.element      : the square AMat to be raised to A.exponent
  A.exponent     : the exponent (an integer)

A.type = "conjugate"
  A.element      : the square AMat to be conjugated
  A.conjugation  : the conjugating invertible AMat

A.type = "directSum"
  A.summands     : List of AMats of the same characteristic

A.type = "tensorProduct"
  A.factors      : List of AMats of the same characteristic

A.type = "galoisConjugate"
  A.element      : the AMat to be Galois conjugated
  A.galoisAut    : the Galois automorphism

Optional fields for all types of AMats

  A.name         : a string for the name
  A.isInvertible : flag to indicate, that A is invertible
  A.inverse      : an AMat representing A^-1
  A.determinant  : the determinant of A
  A.trace        : the trace of A
  A.rank         : the rank of A
  A.isPermMat    : a flag indicating that A is a
                   permutation matrix
  A.perm         : the perm represented by A
  A.isMonMat     : a flag indicating that A is a
                   monomial matrix
  A.mon          : the mon represented by A
  A.mat          : the mat represented by A
  A.isSimplified : a flag indicating, that A is simplified
                   by the function SimplifyAMat
```

```
Fundamental Constructors and Tests for AMats
--------------------------------------------

IsAMat( <obj> )
  tests whether <obj> is an AMat. The particular case of AMat
  is provided by A.type and the additional fields are available
  directly from the record.

IsInvertibleMat( <amat> )
  tests if <amat> is invertible and sets the field
  <amat>.isInvertible.

IsIdentityMat( <amat> )
  checks whether <amat> is a square IdentityMat.

IdentityPermAMat( <size> [, <char/field> ] )
IdentityMonAMat( <size> [, <char/field> ] )
IdentityMatAMat( <size> [, <char/field> ] )
IdentityMatAMat( <dimensions> [, <char/field> ] )
  construct an "perm"/"mon"/"mat"-amat representing
  the identity matrix of size <size>. In the case "mat"
  the matrix may be rectangle containing at the position
  (i, j) one, if i = j and zero else.

AllOneAMat( <size> [, <char/field> ] )
AllOneAMat( <dimensions> [, <char/field> ] )
  constructs the all-one matrix of type "mat" of given
  size and characteristic. The default characteristic is 0.

NullAMat( <size> [, <char/field> ] )
NullAMat( <dimensions> [, <char/field> ] )
  construct the all-zero matrix of type "mat" of given
  size and characteristic. The default characteristic is 0.

DiagonalAMat( <list> )
  construct the amat having <list> as diagonal. If all
  elements of <list> are <> 0, then the resulting amat
  is of type "mon", else of type "directSum".
  Note that the elements of list must lie in a common
  field.

DFTAMat( <size> [, <char/field>] )
  constructs a "mat"-amat representing a DFT of
```

size <size>. The default characteristic is 0.
If <char> is given, then a DFT is constructed
iff <size> and <char> are coprime, if <field> is
given, then a DFT is constructed iff <size>
divides Size(<field>) - 1.
The field <amat>.isDFT is set.

SORAMat(<size> [, <char/field>])
constructs a "mat"-amat representing an SOR of
size <size>. SOR(n) (Split One Rep) is defined
as the following (n x n) - matrix

```
             _               _
            |1  1  1  1  .  .    |
            |1 -1  0  0          |
            |1  0 -1  0          |
  SOR(n) =  |1  0  0 -1          |
            |.           .       |
            |.              .    |
            |                    |
```

The SOR(n) is the sparsest matrix that splits off
the onerep contained in a permrep. The number of
entries is 3n - 2.
The default characteristic is 0. The field
<amat>.isSOR is set.

AMatPerm(<perm>, <degree> [, <char/field>])
constructs a "perm"-amat from <perm> of given degree
<degree> and characteristic <char/field>.

AMatMon(<mon>)
constructs a "mon"-amat from <mon>.

AMatMat(<mat> [, <hint>])
constructs a "mat"-amat from <mat>. The <hint>
"invertible" may be given.

Structural Symbolic Constructors for AMats

ScalarMultipleAMat(<scalar>, <amat>)
forms the amat of type "scalarMultiple" representing
the scalar multiple of <scalar> and <amat>, which must

```
  have the same characteristic. In addition, the integers
  operate on the finite fields. Hence, -<amat> is possible.

<amat> * <amat>
<scalar> * <amat>
  forms the product resp. the scalarMultiple. In the first
  case the sizes must be compatible, in both cases the
  characteristic must be compatible. In addition, the integers
  operate on the finite fields. Hence, -<AMat> is admissible.
<amat> / <amat>
  forms the quotient of the amats. Characteristic and
  size have to be compatible, the second amat must be
  square and invertible.

PowerAMat( <amat>, <int> [, <hint> ] )
  forms the amat <amat> ^ <int> of type "power". If <int>
  is negative then <amat> is checked for invertibility if
  not the hint "invertible" is supplied.

ConjugateAMat( <amat1>, <amat2> [, <hint> ] )
  forms the conjugate <amat1> ^ <amat2> of type "conjugate" .
  The amat <amat2> is checked if it is invertible if not the
  hint "invertible" is supplied

<amat> ^ <int>
<amat> ^ <amat>
  forms the power of an amat resp. the conjugate of
  an amat by another amat. The amats have to be square.


DirectSumAMat( <amat1>, .., <amatN> ) ; N >= 1
DirectSumAMat( <list-of-amat> )
  forms the direct sum of the amats given. Note that the
  amats does not have to be square, but must be of common
  characteristic.

TensorProductAMat( <amat1>, .., <amatN> ) ; N >= 1
TensorProductAMat( <list-of-amat> )
  forms the tensor (or kronecker) product of the
  amats given. Note that the amats does not have to be
  square, but must be of common characteristic.

GaloisConjugateAMat( <amat>, <gal-aut/int> )
  forms the Galois-conjugate of A under the field automorphism
```

```
  aut. The automorphism may be specified by an integer k.
  In case of a finite field
    aut = FrobeniusAutomorphism(A.baseField)^k.
  In case of a number field (in GAP a subfield of a cyclotomic
  field) it is
    aut = NFAutomorphism(A.baseField, k) = x -> GaloisCyc(x, k).

Pretty Printing of AMats
------------------------

AMatOps.Print( <amat> [, <indent> , <indentStep> , <bp> ] )

Fundamental Operations with AMats
---------------------------------

InverseAMat( <amat> )
  returns an amat representing the inverse of
  <amat> if possible. The calculation uses the fact,
  that inversion is compatible with most of the
  structures represented by an mat.

DeterminantAMat( <amat> )
  calculates the determinant of <amat> if it is square
  and stores the result in <amat>.determinant

TraceAMat( <amat> )
  calculates the trace of <amat> if it is square
  and stores the result in <amat>.trace

RankAMat( <amat> )
  calculates the rank of <amat>. The result is stored
  in A.rank.

Flattening out AMats
--------------------

IsPermMat( <amat> )
IsMonMat( <amat> )
  test if <amat> can be flattened into a "perm" or "mon" amat.
  The result is memorized in <amat>.isPermMat/.isMonMat.
  A "mon" amat is always invertible!
  Note that the names of the operations are not IsPermAMat,
  IsMonAMat since <amat> can be of any type but represents
  a permutation/monomial matrix in a mathematical sense.
```

```
PermAMat( <amat> )
MonAMat( <amat> )
MatAMat( <amat> )
  converts <amat> into a Perm/Mon/Mat-object and memorizes
  the result in <amat>.perm/mon/mat. If <amat> cannot be
  converted then false is returned.

PermAMatAMat( <amat> )
MonAMatAMat( <amat> )
MatAMatAMat( <amat> )
  construct a "perm"/"mon"/"mat"-amat equal to <amat> if
  possible or returns false otherwise.

Simplifying AMats
-----------------

SimplifyAMat( <amat> )
  simplifies <amat>, e.g. by removing identity matrices
  in products.

Functions for AMats
-------------------

kbsAMat( <amat1>, .., <amatN> ) ; N >= 1
kbsAMat( <list-of-amat> )
  calculates the joined kbs of the amats given
  (cf. kbs() in permblk.g).

SubmatrixAMat( <amat>, <list-of-indices> )
  calculates the submatrix of amat obtained by
  extracting all components with indices in <inds>.
  A "mat"-amat is returned.

kbsDecompositionAMat( <amat> )
  decomposes <amat> into a conjugated (by a "perm"-amat)
  direct sum of "mat"-amats as far as possible.

LinearComplexityAMat( <amat> )
  computes an upper bound for the linear complexitiy
  according to the model of Clausen.
```

```
Abstract Representations of Finite Groups (ARep)
================================================

The class ARep provides a term algebra for matrix
representations of finite groups. This means there
are a number of symbolic constructions for AReps
like direct sum or Galois conjugation which return
expressions (terms) in the original representations.
  The strategy for simplification of the expressions
is to construct even trivial expressions and provide
functions to apply simplifications. Hence, a term
can be R^id unless you explicitly request simplification.

<ARep> ::=
; atomic cases
    <perm>                                  ; "perm"
  | <mon>                                   ; "mon"
  | <mat>                                   ; "mat"

; composed cases
  | <ARep> ^ <AMat>                         ; "conjugate"
  | DirectSum(<ARep>, .., <ARep>)           ; "directSum"
  | InnerTensorProduct(<ARep>, .., <ARep>)  ; "innerTensorProduct"
  | OuterTensorProduct(<ARep>, .., <ARep>)  ; "outerTensorProduct"
  | GaloisConjugate(<ARep>, <aut>)          ; "galoisConjugate".
  | Restriction(<ARep>, <subgrp>)           ; "restriction"
  | Induction(<ARep>, <supergrp>)           ; "induction"
  | Extension(<ARep>, <ext-char>)           ; "extension"

An ARep is a GAP-Rec R with the following
mandatory fields common to all types of ARep

  R.isARep       := true, identifies AReps
  R.operations   := ARepOps, GAP-operations record of R
  R.char         : characteristic of the base field
  R.degree       : degree of the representation
  R.source       : group being represented; the group must
                   contain a list R.source.theGenerators
                   of generators which are never changed;
                   representations are specified by giving
                   images for R.source.theGenerators
  R.type         : string identifying the type of ARep

The following fields are mandatory to special types
```

```
R.type = "perm"
  R.theImages    : list of Perms for the images of
                   R.source.theGenerators
  R.isTransitive : optional field to indicate, that
                   R is transitive
  R.transitivity : optional field containing the degree of
                   transitivity of R, at the moment
                   this is an integer
  R.induction    : optional field containing a conjugated
                   (by a "mon"-AMat) "induction"-ARep of a
                   one dimensional "mon"-ARep equal to R

R.type = "mon"
  R.theImages    : list of Mons (cf. mon.g)
  R.isTransitive : optional field to indicate, that
                   R is transitive
  R.transitivity : optional field containing the degree of
                   transitivity of R, at the moment
                   this is an integer
  R.induction    : optional field containing a conjugated
                   (by a "mon"-AMat) "induction"-ARep of a
                   one dimensional "mon"-ARep equal to R

R.type = "mat"
  R.theImages    : list of Mats

R.type = "conjugate"
  R.rep          : an ARep to be conjugated
  R.conjugation  : an AMat conjugating R.rep

R.type = "directSum"
  R.summands     : list of AReps of the same R.source, R.char
                  The sources of the areps is the same GAP-Rec.

R.type = "innerTensorProduct"
  R.factors      : list of AReps of the same R.source, R.char
                  The sources of the areps is the same GAP-Rec.

R.type = "outerTensorProduct"
  R.isOuter      : flag to indicate that R.source is the
                  outer direct product of the sources of
                  the factors; otherwise R.source is the
                  inner direct product of the sources of
```

```
                  the factors (which are normal subgroups)
  R.factors      : list of AReps of the same R.char; note that
                  the R.source is the direct product of all
                  Rk.source for Rk in R.factors
  R.projections  : list of group homomorphisms from R.source
                  into R.factors[i].source; the individual
                  entries of this list may be empty if
                  the projection has not been used before
  R.embeddings   : list of group homomorphisms from the
                  sources of the factors into R.source; the
                  entries may be empty if the embeddings
                  have not been used before

R.type = "galoisConjugate"
  R.rep          : an ARep to be conjugated
  R.galoisAut    : a Galois automorphism or an integer for
                  the Galois conjugation in CF(n) or GF(p^n)

R.type = "restriction"
  R.rep          : an ARep of a supergroup of R.source;
                  the group R.source and R.rep.source
                  have a common parent group

R.type = "induction"
  R.rep          : an ARep of a subgroup of R.source;
                  the group R.rep.source has the same
                  parent group as R.source
  R.transversal  : a transversal of Cosets(R.source, R.rep.source)

R.type = "extension"
  R.rep          : an irreducible ARep of a subgroup of
                  R.source with R.rep.character being bound;
                  the groups R.source and R.rep.source share
                  a common parent group
  R.character    : the irreducible character of the extended
                  representation of R.source such that the
                  restriction to R.rep.source is R.rep.character

Optional fields common to all types of ARep

  R.character     : the GAP-character belonging to R
  R.isIrreducible : set if R is known to be irreducible or not
  R.kernel        : the kernel of R if known
  R.hom           : the GAP-group homomorphism corresponding
```

```
                    to R constructed as
                      GroupHomomorphismByImages(
                        R.source,             Group(R.theImages),
                        R.source.theGenerators, R.theImages )
  R.name           : field to give a name to R
  R.isPermRep      : flag to indicate if R can be replaced by
                     a "perm"-ARep equal to R
  R.permARep       : the "perm"-ARep equal to R
  R.isMonRep       : flag to indicate if R can be replaced by
                     a "mon"-ARep equal to R
  R.monARep        : the "mon"-ARep equal to R
  R.matARep        : a "mat"-ARep equal to R

Fundamental Constructors for AReps
----------------------------------

IsARep( <obj> )
  tests of <obj> is an ARep.

TrivialPermARep( <grp>, [, <degree> [,<char/field> ] )
TrivialMonARep(  <grp>, [, <degree> [,<char/field> ] )
TrivialMatARep(  <grp>, [, <degree> [,<char/field> ] )
  the "perm"/"mon"/"mat"-ARep of mapping every group element
  onto a one of degree <degree>. The default for the degree is 1,
  the  default for the characteristic is 0.

RegularARep( <grp> [, <char/field> ] )
  returns an "induction"-arep of the onerep on
  the trivial subgroup of <grp>.

NaturalARep( <matgrp> )
NaturalARep( <mongrp> )
NaturalARep( <permgrp>, <degree> [, <char/field> ] )
  a group taken as a representation of itself.

ARepByImages( <grp>, <list-of-perm>, <degree>
              [, <char/field> ] [, <hint> ] )
ARepByImages( <grp>, <list-of-mon> [, <hint> ] )
ARepByImages( <grp>, <list-of-mat> [, <hint> ] )
  the representation defined by mapping the list of generators
  grp.theGenerators pointwise onto the elements of the list
  given as the second argument. The optional argument <hint>
  is a string which gives a hint to avoid the check if the
  list of images actually defines a group homomorphism. The
```

```
  possible hints are "hom" (image do define a homomorphism)
  and "faithful" (image define an injective homomorphism).

ARepByHom( <grphom-to-matgrp> )
ARepByHom( <grphom-to-mongrp> )
ARepByHom( <grphom-to-permgrp>, <degree> [, <char/field> ] )
  the representation defined by a group homomorphism.

ARepByCharacter( <1dim-character> )
  the monomial representation defined by a
  1-dimensional character.

Structural Symbolic Constructors for AReps
------------------------------------------

ConjugateARep( <arep>, <amat> [, <hint> ] )
<arep> ^ <amat> ; shorthand
  the representation <arep> conjugated with an invertible matrix
  represented by <amat>, an object of type AMat. Note that the
  <hint> can be the string "invertible" to avoid the check for
  <amat> to be invertible.

DirectSumARep( <arep1>, .., <arepN> ) ; N >= 1
DirectSumARep( <list-of-areps> )
  direct sum of AReps. Note that the areps have to represent a
  common group in a common characteristic.

InnerTensorProductARep( <arep1>, .., <arepN> ) ; N >= 1
InnerTensorProductARep( <list-of-areps> )
  inner tensor product of AReps. Note that the areps have to
  represent a common group in a common characteristic.

OuterTensorProductARep( [ <grp> ,] <arep1>, .., <arepN> ) ; N >= 1
OuterTensorProductARep( [ <grp> ,] <list-of-arep> )
  outer tensor product of AReps. Note that the areps have to
  have a common characteristic. If the first argument is the
  optional <grp>, then this group is the source of the result.
  Note that <grp> has to be the inner direct product of the
  sources of all factors and that this is not tested! If no
  <grp> is given, then the GAP-function DirectProduct() is
  used to construct the outer direct product of the sources
  of the factors.

GaloisConjugateARep( <arep>, <gal-aut/int> )
```

```
  the Galois conjugate of <arep> with the galois conjugation
  defined by <gal-aut/int>. This can be a field automorphism
  or an integer k, in which case x -> x^(FrobeniusAut^k) or
  x -> GaloisCyc(x, k) is meant.

RestrictionARep( <arep>, <subgrp> )
  the representation <arep> restricted to a subgroup.
  It is allowed that <subgrp> does not have the same
  parent group as <arep>.source.

InductionARep( <arep>, <supergroup> [, <transversal> ] )
  the induced representation <arep> on the <supergroup>.
  If no transversal is provided then the function will
  choose one as [t_1, .., t_r]. The convention for the
  induction is RG = g -> [ RHDot(t_i g t_j^-1) | i, j ],
  what implies, that [t_1, .., t_r] is a right transversal.
  It is allowed that <supergroup> does not have the same
  parent group as <arep>.source. The given <transversal>
  is not checked to be one.

ExtensionARep( <arep>, <extending-character> )
  the representation <arep> extended to a representation
  of a supergroup affording the extending character.
  Note that the extending character and the character of
  <arep> must both be irreducible. (The extension is
  evaluated using Minkwitz's extension formula.)
  It is allowed that <supergroup> does not have the same
  parent group as <arep>.source.


Pretty Printing of AReps
------------------------

GroupWithGenerators( <group> )
GroupWithGenerators( <list-of-grpelts> )
  construct a group with the field .theGenerators being
  set to a fixed non-empty generating list. If <group>
  is given then this group record is modified by adding
  the field .theGenerators if this does not exist already.
  If <list-of-grpelts> is given then this must be a non-
  empty list of group elements acting as the fixed generating
  list of the resulting group.

ARepOps.Print( <arep> [, <indent> ] )
```

```
  prints the <arep> beginning at the current cursor
  position which is assumed to be at the beginning
  of a line, indented at <indent> spaces.

Fundamental Operations with AReps
---------------------------------

ImageARep( <grpelt/list-of-grpelts>, <arep> )
<grpelt> ^ <arep> ; shorthand
  evaluates the ARep at the group element;
  the result is an AMat.

IsEquivalentARep( <arep1>, <arep2> )
  determines, whether <arep1> and <arep2> are equivalent
  representations of the same source and char.

CharacterARep( <arep> )
  the character of the representation. The result
  is stored in <arep>.character.

IsIrreducibleARep( <arep> )
  determines if <arep> is an irreducible representation.

KernelARep( <arep> )
  the kernel of the representation.

IsFaithfulARep( <arep> )
  determines if <arep> is an injective homomorphism.

Flattening Out AReps
--------------------

IsPermRep( <arep> )
IsMonRep(  <arep> )
  test if <arep> can be turned into a "perm" or "mon" ARep.
  The result is memorized in .isPermRep/.isMonRep. Note that
  the names of the operations are *not* 'IsPermARep' etc.
  since <arep> can be any type but it *represents* a
  permutation representation in the mathematical sense.

PermARepARep( <arep> )
MonARepARep(  <arep> )
MatARepARep(  <arep> )
  constructs a "perm"/"mon"/"mat"-ARep equal to the
```

```
given <arep> or returns false if this is not possible.
The result is memorized in .permARep/.monARep/.matARep.
```

Constructive Representation Theory

Based on the class ARep a number of functions is
provided to calculate with representations. The
most functions deal with permutation and monomial
representations and are decompositions in a sense.

Functions for Characters

```
IsRestrictedCharacter( <chi>, <chisub> )
  tests if <chisub> is a restriction of <chi>.

AllExtendingCharacters( <chi>, <supergrp> )
  calculates the list of all irreducible characters
  of <supergrp> whose restriction is the irreducible
  character <chi>.

OneExtendingCharacter( <character>, <supergroup> )
  calculates a character of <supergroup> extending
  <character> or returns false.
```

Intertwining Space of Representations

```
IntertwiningSpaceARep( <arep1>, <arep2> )
  computes a base of the intertwining space
    Int( <arep1>, <arep2> ) = { M | <arep1> M = M <arep2> }
  represented by a list of amats.
  The convention for the intertwining space follows
  Clausen, Baum. It is consistent with Minkwitz.
  Note that Int( <arep1>, <arep2> ) consists of
  deg(rep2) x deg(rep1)-matrices.

IntertwiningNumberARep( <arep1>, <arep2> )
  calculates the intertwining number of <arep1>, <arep2>
  which is the dimension of the intertwining space
  or the scalar product of the characters resp.
```

Functions for Permutation and Monomial Representations

```
UnderlyingPermARep( <arep> )
```

```
  constructs the underlying permrep of the monrep <arep>
  as a "perm"-ARep, which can be obtained by replacing
  all entries in the monomial matrices by 1 (in the
  suitable field).

IsTransitiveMonRep( <arep> )
  decides whether <arep> is a transitive monrep or not.
  The result is stored in the field <arep>.isTransitive.

IsPrimitiveMonRep( <arep> )
  decides whether <arep> is a primitive monrep or not.

TransitivityDegreeMonRep( <arep> )
  returns the transitivity degree of the monrep <arep>

OrbitDecompositionMonRep( <arep> )
  decomposes the monrep <arep> with respect to the
  orbits as
    <arep> = DirectSum(R1, .., Rn) ^ P
  where R1 .. Rn are "mon"-AReps and P denotes a
  "perm"-AMat.

TransitiveToInductionMonRep( <arep> [, <point> ] )
  decomposes a transitive monrep <arep> as
    <arep> =
      ConjugateARep(
        InductionARep(L, <arep>.source, T),
        D
      )
  where Stab is the stabilizer of <point>, L a onedimensional
  "mon"-arep of Stab and D a diagonal "mon"-AMat. The list
  T is a transversal of Stab in <arep>.source.
  The default for <point> is the largest point (<arep>.degree).
  If <arep> is a permrep then D is the IdentityMonAMat of
  suitable size and char. If <point> = <arep>.degree, then
  the result is stored in <arep>.induction.

DirectSumOfInducedMonRep( <arep> )
  decomposes a monrep <arep> with respect to the
  orbits into a conjugated (by a "mon"-AMat) direct
  sum of induced onedimensional "mon"-AReps.

General Functions
-----------------
```

```
kbsARep( <arep> )
  determines the conjugated block structure of <arep>
  (cf. kbs in permblk.g). Note that if <arep> is
  monomial, the kbs is exactly the list of orbits
  of <arep> on [1..R.degree].

Decomposing Representations
---------------------------

InsertedInductionARep( <"induction"-arep>, <group> )
  given an "induction"-ARep RUG = InductionARep(RU, G)
  and the <group> H such that U <= H <= G, this function
  decomposes RUG into
    RUG = InductionARep(InductionARep(RU, H), G) ^ M
  where M is an AMat with structure similar to the
  induced representation RUG.

ConjugationPermLists (
  [ <permgrp>,] <list-of-perm1>, <list-of-perm2>
)
  calculates a permutation p in <permgrp> which conjugates
  <list-of-perm1> elementwise onto <list-of-perm2>.
  The default for <permgrp> is the symmetric group on degree
  many points.

ConjugationTransitivePermReps( <arep1>, <arep2> )
  returns a "perm"-amat p on <arep1>.degree many points
  such that <arep1> ^ p = <arep2> and false if this is
  not possible. The areps must have common source and
  characteristic.

ConjugationTransitiveMonReps( <arep1>, <arep2> )
  returns a "mon"-amat m on <arep1>.degree many points
  such that <arep1> ^ m = <arep2> and false if this is
  not possible. The areps must have common source and
  characteristic.

ConjugationPermReps( <arep1>, <arep2> )
  returns for permutation areps <arep1>, <arep2> a
  "perm"-amat p on <arep1>.degree many points such
  that <arep1> ^ p = <arep2> and false if this is
  not possible. The areps must have common source
  and characteristic.
```

```
TransversalChangeInductionARep(
  <"induction"-arep>, <transversal> [, <hint> ]
)
  Given an "induction"-ARep R,
    R = InductionARep(L, G, T)
  and a <transversal> of L.source\G, R is decomposed as
    R = InductionARep(L, G, <transversal>) ^ M.
  If L is a monrep (e.g. L.degree = 1), then M is a
  "mon"-AMat, else an AMat with a structure similar to R.
  The <hint> "isTransversal" can be supplied to avoid
  testing it.

OuterTensorProductDecompositionMonRep( <arep> )
  decomposes the transitive monrep <arep> into a
  conjugated (by a "mon"-AMat) outer tensorproduct
  of "mon"-AReps as far as possible, namely
    <arep> =
      ConjugateARep(
        OuterTensorProductARep(
          <arep>.source,
          "mon"-ARep1, .., "mon"-ARepN
        ),
        M
      )
  with a monomial matrix M.

InnerConjugationARep( <arep>, <supergroup>, <element> )
  calculates <arep> ^ <element> as representation of
  <arep>.source ^ <element>, <element> must lie in <supergroup>.
  For a representation R of H <= G, and an
  element t in G the representation R ^ t is defined as
    (R ^ t)(x) = R(t x t^-1) for all x in H ^ t.
  The returned arep is of type "perm", "mon" or "mat",
  whatever poosible.
  If H is normal in G, then R ^ t is returned as rep of H,
  else  H ^ t is constructed as a group with
    (H ^ t).theGenerators = H.theGenerators ^ t.

RestrictionInductionARep( <"induction"-arep>, <subgroup> )
  Given an <"induction"-arep> R of G, induced from a
  onedimensional representation L of a subgroup H <= G
  and a <subgroup> K of G, the function calculates
  an arep equal to R decomposing (R restriction K)
```

```
  according to Mackey's subgroup theorem.
  If s_1, .., s_k represent the double cosets H\G/K,
  the the decomposition is given by
    (R restriction K) =
      ConjugateARep(
        DirectSumARep(
          InductionARep(R_i, K),
          i = 1..k
        ),
        M
      ),
   where M is a monomial matrix and R_i are onedimensional
   "mon"-areps obtained by restricting L^(s_i) to
   (H^(s_i) intersect K).

AllMaximalNormalSubgroupsBetween( <group>, <subgroup> )
  calculates all normal subgroups N of <group> with
    <subgroup> <= N < <group> and (<group> : N) = prime,
  if no such N exists, then false is returned.

OneMaximalNormalSubgroupBetween( <group>, <subgroup> )
  calculates one normal subgroup N of <group> with
    <subgroup> <= N < <group> and (<group> : N) = prime,
  if no such N exists, then false is returned.

RestrictionToSubmoduleARep( <arep>, <list> [, <hint> ] )
  calculates the restriction of <arep> to
  the submodule generated by the basevectors
  with the indices in <list>.
  The optional hint "hom" indicates, that the restriction
  yields a representation, i.e. that <list-of-posints>
  is contained in the kbs of <arep>.
  The restriction is of type "perm", "mon", "mat", if
  <arep> is a perm-, mon-, matrep.

kbsDecompositionARep( <arep> )
  decomposes <arep> into a conjugated direct sum
  according to the kbs (cf. permblk.g) as far as
  possible
    <arep> =
      ConjugateArep(
        DirectSumARep( <arep1>, .., <arepN> ),
        P
      )
```

```
  with a permutation matrix P. For monreps this
  function does exactly the same as the function
  OrbitDecompositionMonRep.
  The <arepi> are of type "perm", "mon", "mat", if
  <arep> is  a perm-, mon-, matrep resp.

ExtensionOnedimensionalAbelianRep( <arep>, <group> )
  calculates an extension of an <arep> of degree 1 of
  a subgroup H of <group>, if <group>/kernel(R) is abelian.
  The extension is a "mon"-arep and chosen to be over the
  smallest possible extension field. No character theory is used.

ARepWithCharacter( <chi> )
  calculates an arep with character <chi>.
  The group must be solvable.

DecompositionMonRep( <arep> [, <hint> ] )
  decomposes <arep> into irreducibles and determines
  a factorized decomposition matrix A. More precisely
  <arep> is decomposed as
    <arep> =
      ConjugateARep(
        DirectSumARep(R_i, i = 1..k),
        A ^ -1
      )
  where all R_i are irreducible. The R_i are ordered
  according to their character with the exception, that
  the trivial onerep is the smallest rep. Equivalent
  R_i are equal.
  If the <hint> "noOuter" is supplied, <arep> is not
  checked to be an outer tensor product.
  Note, that the decomposition matrix A is accessible by
  A = R.conjugation.element. A is simplified by the
  function SimplifyAMat.
  The structure of A represents a fast algorithm for
  multiplication with A.
```

Bibliography

[1] Agarwal, R. C., und Cooley, J. W. New algorithms for digital convolution. *IEEE Trans. Acoustics, Speech, and Signal Processing ASSP-25* (1977), 392–410.

[2] Apple, G., und Wintz, P. Calculation of Fouriertransforms on Finite Abelian Groups. *IEEE Trans. Inform. Theory IT-16* (1970), 233–236.

[3] Atkinson, M. *Computational Group Theory.* Academic Press, 1984.

[4] Beauchamp, K. *Applications of Walsh and related functions.* Academic Press, 1984.

[5] Besche, H. U., und Eick, B. The Groups of Order up to 1000, except 512 and 768. *http://www-gap.dcs.st-and.ac.uk/~gap/Info/datalib.html* (1996).

[6] Beth, T. *Methoden der Schnellen Fouriertransformation.* Teubner Verlag, 1984.

[7] Beth, T. On the computational complexity of the general discrete Fourier transform. *Theoretical Computer Science 51* (1987), 331–339.

[8] Beth, T., und Grassl, M. Improved decoding of quantum error correcting codes from classical codes. In *PhysComp96* (1996), T. Toffoli, M. Biafore, und J. Leao, Hrsg.

[9] Beth, T., Jungnickel, D., und Lenz, H. *Design Theory.* BI-Wiss.-Verl., 1985.

[10] Bluestein, L. I. A linear filtering approach to the computation of the discrete fourier transform. *IEEE Trans. AU-18 18* (1970), 451–455.

[11] Buekenhout, F. *Handbook of Incidence Geometry.* Elsevier, North Holland, 1995.

[12] Butler, G. *Fundamental Algorithms for Permutation Groups.* Lecture Notes in Computer Science, 559. Springer, 1991.

[13] Clausen, M. *Beiträge zum Entwurf schneller Spektraltransformationen (Habilitationsschrift).* Univ. Karlsruhe, 1988.

[14] Clausen, M. Fast generalized Fourier transforms. *Theoretical Computer Science 67* (1989), 55–63.

[15] Clausen, M. A Direct Proof of Minkwitz's Extension Theorem. *Applicable Algebra in Engineering, Communication and Computing 8* (1997), 305–306.

[16] Clausen, M., und Baum, U. *Fast Fourier Transforms.* BI-Wiss.-Verl., 1993.

[17] Colbourn, C., und Dinitz, J. *The CRC Handbook of Combinatorial Designs.* CRC Press, 1996.

[18] Conway, J. H., Curtis, R. T., Norton, S. P., und Wilson, R. A. *Atlas of Finite Groups.* Clarendon Press, Oxford, 1985.

[19] Cooley, J. W., und Tukey, J. W. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation 19* (1965), 297–301.

[20] Crouch, R. B. Monomial groups. *Trans. Am. Math. Soc. 80* (1955), 187–215.

[21] Curtis, W., und Reiner, I. *Representation Theory of Finite Groups.* Interscience, 1962.

[22] Curtis, W., und Reiner, I. *Methods of Representation Theory*, Vol. 1. Interscience, 1981.

[23] Diaconis, P., und Rockmore, D. Efficient computation of the Fourier transform on finite groups . *Amer. Math. Soc. 3(2)* (1990), 297–332.

[24] Dixon, J., und Mortimer, B. *Permutation Groups.* Springer, 1996.

[25] Dornhoff, L. *Group Representation Theory.* Pure and Applied Mathematics. Dekker New York, 1971.

[26] Egner, S. *Zur Algorithmischen Zerlegungstheorie Linearer Transformationen mit Symmetrie.* Diss., Univ. Karlsruhe, Informatik, 1997.

[27] Egner, S., Püschel, M., und Beth, T. Decomposing a Permutation into a Conjugated Tensor Product. *Proceedings of the ISSAC* (1997), 101–108.

[28] Elliott, D. F., und Rao, K. R. *Fast Transforms — Algorithms, Analyses, Applications.* Academic Press, 1982.

[29] Feit, W. *The Representation Theory of Finite Groups.* North-Holland, 1982.

[30] Fulton, W., und Harris, J. *Representation Theory.* Springer, 1991.

[31] Gauss, C. F. *Carl Friedrich Gauss Werke*, Vol. 3. Königliche Gesellschaft der Wissenschaften: Göttingen, 1866.

[32] Haar, A. Zur Theorie der orthogonalen Funktionen-Systeme. *Math. Ann. 69* (1910), 331–371.

[33] Hall jr., M. *The Theory of Groups.* Chelsea Publ., 1976.

[34] Heidemann, M., Johnson, D., und Burrus, C. Gauss and the History of the Fast Fourier Transform. *Archive for History of Exact Sciences 34* (1985), 265–277.

[35] Huang, T. How the Fast Fourier Transform Got its Name . *Computer 3* (1971), 15.

[36] Huppert. *Endliche Gruppen I.* Springer, 1967.

[37] James, G., und Kerber, A. *The representation theory of the symmetric group.* Addison-Wesley, 1981.

[38] James, G., und Liebeck, M. *Representations and Characters of Groups.* Cambridge Univ. Press, 1993.

[39] Leon, S. J. Permutation Group Algorithms Based on Partitions, I: Theory and Algorithms . *Journal of Symbolic Computation 12* (1991), 533–583.

[40] Leopoldt, H.-W. Darstellungstheorie endlicher Gruppen. Vorlesungsskript (1979) an der Universität Karlsruhe, Fakultät für Mathematik.

[41] MacWilliams, F., und Sloane, N. *The theory of error-correcting codes.* North-Holland Publ.Comp., 1992.

[42] Malvar, H. S. *Signal Processing with Lapped Transforms.* Artech House, 1992.

[43] Maslen, D., und Rockmore, D. Generalized FFTs – a survey of some recent results. *Proceedings of IMACS Workshop in Groups and Computation 28* (1995), 182–238.

[44] Mertins, A. *Signaltheorie.* Teubner Verlag, 1996.

[45] Minkwitz, T. *Algorithmensynthese für lineare Systeme mit Symmetrie.* Diss., Universität Karlsruhe, 1993.

[46] Minkwitz, T. Algorithms Explained by Symmetry. *Lecture Notes on Computer Science 900* (1995), 157–167.

[47] Minkwitz, T. Extension of Irreducible Representations. *Applicable Algebra in Engineering, Communication and Computing 7* (1996), 391–399.

[48] Minkwitz, T. On the generation of algorithms for linear transforms and systems with symmetry. Tech. Rep., EISS, 1997.

[49] Ore, O. Theory of monomial groups. *Trans. Am. Math. Soc. 51* (1942), 15–64.

[50] Passman, D. *Permutation Groups.* Benjamin/Cummings, 1968.

[51] Pennebaker, W., und Mitchell, J. *JPEG: still image data compression standard.* Van Nostrand Reinhold, 1993.

[52] Pichler, F. *Mathematische Systemtheorie; Dynamische Konstruktionen.* de Gruyter, Berlin, 1975.

[53] Rader, C. M. Discrete fourier transforms when the number of data samples is prime. *Proceedings of the IEEE 56* (1968), 1107–1108.

[54] Rao, K., und Hwang, J. *Techniques & Standards for Image, Video and Audio Coding.* Prentice Hall PTR, 1996.

[55] Rockmore, D. Efficient computation of Fourier inversion for finite groups. *Assoc. Comp. Mach. 41(1)* (1994), 31–66.

[56] Rockmore, D. Some applications of generalized FFT's. *Proceedings of DIMACS Workshop in Groups and Computation 28* (1995), 329–370.

[57] Schönert, M. et al. *GAP — Groups, Algorithms and Programming*, fifth ed., 1995.

[58] Serre, J. *Linear Representations of Finite Groups.* Springer, 1977.

[59] Sturmfels, B. *Algorithms in Invariant Theory.* Springer, 1993.

[60] Wielandt, H. *Finite Permutation Groups.* Academic Press, 1964.

Index