CONDENSATION OF INDUCED REPRESENTATIONS AND AN APPLICATION: THE 2-MODULAR DECOMPOSITION NUMBERS OF Co_2

JÜRGEN MÜLLER AND JENS ROSENBOOM

ABSTRACT. We present an algorithm to condense induced modules for a finite group over a finite field. It is built on existing tools of computational group and representation theory, the MeatAxe for matrix algebras over finite fields and the Schreier-Sims methods for permutation groups, and has been implemented in GAP and as an extension of the MeatAxe. As an application we construct and analyze an induced module for the sporadic simple Conway group Co_2. The result of this analysis is used to complete the 2-modular Brauer character table of Co_2.

1. INTRODUCTION

In recent years, condensation has become one of the most valuable tools in computational representation theory of finite groups. Especially, the determination of many of the Brauer character tables of sporadic simple groups known today, see [9], would not have been possible without this powerful technique. As it turns out that many of the interesting modules for finite groups are too large to be constructed or analyzed by means of a computer directly, one tries to 'condense' these modules to smaller ones which still reflect enough of the original structure.

The functorial formalism to describe the theoretical correctness of the condensation method, which is introduced in Section 2, dates back at least to [7]. This formalism gives us a detailed description of which structural information is retained and which is lost under condensation. Under certain circumstances condensation turns out to induce an equivalence between the module categories of an arbitrary given algebra and of a Morita equivalent basic algebra. This explains why condensation has become a standard tool in the representation theory of finite-dimensional algebras.

But besides its theoretical value, condensation has become a valuable computational tool, as the formal recipe can be translated into a series of explicit computational steps which for suitable types of modules for finite groups can efficiently be performed by help of a computer. This has been implemented

Date: November 12, 2002.

1991 *Mathematics Subject Classification.* 20-04, 20C20, 20C34, 20C40.

e. g. for permutation modules [21, 18, 16, 4] and for tensor product modules over finite fields [22, 14, 17]. In this note we present an algorithm to condense arbitrary induced modules for a finite group over a finite field, thereby enlarging the set of types of modules which can be handled computationally using the condensation technique. In Section 3 we describe how the functorial description of the condensation method translates into group theoretical computational steps, in Section 4 we then show how these steps can be performed explicitly.

Our algorithm is built on existing tools of computational group and representation theory and we assume the reader to be familiar with the MeatAxe, see [15], and the Schreier-Sims method to handle permutation groups. The permutation group part of our algorithm, see 4.1, runs in GAP [19]. For the matrix group part, see 4.2 and 4.3, we have two implementations, one using the MeatAxe implementation [18], the other using a new finite field arithmetic [11] implemented in GAP. The uncondensation program has been implemented as extensions of the MeatAxe implementations [18] and [6]. Of course, all these programs are available from the authors on request.

In Section 5, we apply our algorithm to an induced module for the sporadic simple Conway group Co_2. A careful analysis of the resulting condensed module and an application of the uncondensation technique allows us to complete the determination of the 2-modular Brauer character table of this group begun in [20]. The table is given at the end of this note. This even completes the determination of all the Brauer character tables of Co_2 for all primes dividing its order.

Acknowledgments. The authors gratefully acknowledge discussions with Gerhard Hiss on the topic of this note. Both authors thank the Deutsche Forschungsgemeinschaft (DFG) for financial support in the framework of the joint research project 'Algorithmic number theory and algebra', to which this note is a contribution.

2. Condensation

2.1. Let k be a field, A be a finite-dimensional k-algebra, $e \in A$ be an idempotent and mod–A be the category of finitely generated unital right A-modules. Then the condensation functor with respect to e is given as

$$C_e := ? \otimes_A Ae : \text{mod–}A \longrightarrow \text{mod–}eAe\,.$$

Under this functor, $M \in \text{mod–}A$ is mapped to $M \otimes_A Ae \in \text{mod–}eAe$. The latter can be identified with the subset $Me \subseteq M$, let $\iota : M \otimes_A Ae \to Me \subseteq M$ denote the corresponding injection. Using this identification, a homomorphism $\alpha \in \mathrm{Hom}_A(M, N)$ is mapped to $\alpha|_{Me} \in \mathrm{Hom}_A(Me, Ne)$. We have $C_e \cong \mathrm{Hom}_A(eA, ?)$ as functors, hence C_e is an exact functor.

Let $e = \sum_S \sum_{j=1}^{n_S} e_S^{(j)} \in A$ be an orthogonal decomposition into primitive idempotents, where S runs through the isomorphism types of simple A-modules. For each summand we have $e_S^{(j)} A/\mathrm{Rad}(e_S^{(j)} A) \cong S$ and $n_S \in \mathbb{N}_0$. By $n_S = 0$ we indicate that this type of idempotent does not occur in the above decomposition. We have $\dim_k(Se) = n_S \cdot \dim_k(\mathrm{End}_A(S))$. By the Morita Theorems, see [5, Section 3.D.], C_e is an equivalence of categories if and only if $n_S > 0$ for all S; in this case e is called faithful.

2.2. **Duality.** Let $\Delta : \text{mod–}A \to A\text{–mod}$ be the contravariant duality functor which maps $M \in \text{mod–}A$ to $M^* := \mathrm{Hom}_A(M, k)$ and $\alpha \in \mathrm{Hom}_A(M, N)$ to $(\lambda \mapsto \alpha \cdot \lambda) \in \mathrm{Hom}_A(N^*, M^*)$. Furthermore, let σ be a k-algebra anti-automorphism of A. This defines a contravariant functor $\Sigma : \text{mod–}A \to A\text{–mod}$ which acts as the identity on homomorphism spaces and maps $M \in \text{mod–}A$ to $M^\sigma \in A\text{–mod}$, where the A-action on M^σ is given by $a \cdot m := m \cdot (a^\sigma)$ for $m \in M$, $a \in A$,

If additionally $e^\sigma = e$ holds, then this analogously defines a functor $\text{mod–}eAe \to eAe\text{–mod}$, again called Σ. It is easily seen that $\Sigma \cdot \Delta = \Delta \cdot \Sigma$ holds as equality of functors, both on mod–A and on mod–eAe. This defines duality auto-equivalences of mod–A and of mod–eAe. A module M is called self-dual, if $M \cong M^{\Sigma\Delta}$ holds. Note that the action of the duality auto-equivalence induces an anti-automorphism of the submodule lattice of a self-dual module. Finally, it is also easily seen that $(\Sigma \cdot \Delta) \cdot C_e \cong C_e \cdot (\Sigma \cdot \Delta)$ are naturally equivalent functors. Hence C_e maps a self-dual module to a self-dual module.

2.3. **Uncondensation.** The uncondensation functor with respect to e is given as

$$? \otimes_{eAe} eA : \text{mod–}eAe \longrightarrow \text{mod–}A.$$

Under this functor, $\tilde{M} \in \text{mod–}eAe$ is mapped to $\tilde{M} \otimes_{eAe} eA \in \text{mod–}A$ and $\tilde{\alpha} \in \mathrm{Hom}_{eAe}(\tilde{M}, \tilde{N})$ is mapped to $\tilde{\alpha} \otimes \mathrm{id} \in \mathrm{Hom}_A(\tilde{M} \otimes_{eAe} eA, \tilde{N} \otimes_{eAe} eA)$. If e is faithful, then this is the two-sided inverse functor to the condensation functor.

Uncondensation is applied in the following way. Let $N \in \text{mod–}A$ and $\tilde{N} := N \otimes_A Ae \in \text{mod–}eAe$ be its condensed module. The embedding $\iota : \tilde{N} \to N$ induces the A-homomorphism $\iota^A : \tilde{N} \otimes_{eAe} eA \to N : n \otimes ea \mapsto n \cdot a$ for $n \in \tilde{N}$, $a \in A$. Now let $\tilde{M} \leq \tilde{N}$ be an eAe-submodule, given by an embedding α. Then the image of the A-homomorphism $\alpha^A := (\alpha \otimes \mathrm{id}) \cdot \iota^A : \tilde{M} \otimes_{eAe} eA \to N$ is called the uncondensed module of $\tilde{M}$.

Note that in the applications ι is always known explicitly. Hence given a submodule $\tilde{M} \leq \tilde{N}$, which usually is found during the structural analysis of $\tilde{N}$, the image $\mathrm{Im}(\alpha \cdot \iota) \subseteq N$ can effectively be computed. Then $\mathrm{Im}(\alpha^A) \leq N$ equals the smallest A-submodule of N containing $\mathrm{Im}(\alpha \cdot \iota)$, hence can be found by standard techniques. This is a valuable tool to prove the existence of certain submodules of N, especially in face of the problem of generation we will discuss now.

2.4. **Generation problem.** Let A be generated as a k-algebra by the subset $\mathcal{A} \subseteq A$. We then let $\mathcal{C} := \langle eae; a \in \mathcal{A} \rangle_{k-\text{algebra}} \leq eAe$. Now the condensation subalgebra $\mathcal{C}$ is not necessarily equal to eAe, but may be a strict subalgebra of eAe. To prove equality in practice, only relatively weak criteria are known. They are stated and successfully used e. g. in [12], but unfortunately not practicable in the example we are envisaging. Hence, only the action of $\mathcal{C}$ on condensed modules can be computed explicitly, and so we are faced with the task to analyze condensed modules with respect to their structure as $\mathcal{C}$-modules and then to draw conclusions about their eAe-module structure from this analysis.

2.5. **Fixed point condensation.** We will apply condensation in the following situation. Let G be a finite group and $A := k[G]$ be its group algebra over k. Let $K \leq G$ be a subgroup such that $|K| \neq 0 \in k$; in the sequel K is called the condensation subgroup. Then

$$e = e_K := \frac{1}{|K|} \cdot \sum_{g \in K} g \in k[K] \subseteq k[G]$$

is the centrally primitive idempotent of $k[K]$ belonging to the trivial K-module. We have $e \cdot k[G] \cong (1_K)^G$, where $(1_K)^G$ is the permutation module afforded by the permutation action of G on the cosets of K. Hence by the adjointness of tensor product and homomorphism functors, see [5, Theorem 2.19.], we have $Me \cong \mathrm{Hom}_G((1_K)^G, M) \cong \mathrm{Hom}_K(1_K, M_K) \cong \mathrm{Fix}_M(K)$ as vector spaces, where M is a $k[G]$-module, M_K denotes its restriction to $k[K]$ and $\mathrm{Fix}_M(K) \subseteq M$ consists of the elements of M being fixed by K. Hence this method is called fixed point condensation. As mod–$k[K]$ is a semisimple category, the dimension of the condensed module of M can be computed as the character theoretic scalar product of the trivial K-character and the restriction to K of the Brauer character of M.

Note that by letting $\sigma : k[G] \to k[G] : g \mapsto g^{-1}$ for $g \in G$, the duality $\Sigma \cdot \Delta = \Delta \cdot \Sigma$, see 2.2, specializes to the operation of taking contragradient modules.

3. Condensation of induced modules, theoretically ...

3.1. Let $H \leq G$ be a subgroup and $V \in$ mod–$k[H]$. Let $\{g_i; i \in I\}$ for a suitable index set I be a set of representatives for the H-K double cosets in G and for each $i \in I$ let $\{k_{ij}; j \in I_i\}$ for a suitable index set I_i be a set of representatives for the $H^{g_i} \cap K$ right cosets in K. By the Mackey Decomposition Theorem we have $V^G = V \otimes_{k[H]} k[G] = \oplus_{i \in I} \oplus_{j \in I_i} V \otimes g_i k_{ij}$ as a vector space and hence $(V^G)_K \cong \oplus_{i \in I} ((V^{g_i})_{H^{g_i} \cap K})^K$ as $k[K]$-modules, where $V^{g_i} \in$

mod–$k[H^{g_i}]$ is defined by $v \cdot h^{g_i} := v \cdot h$ for $v \in V$, $h \in H$. This means we have

$$\begin{aligned} V^G e &\cong \mathrm{Hom}_K(1_K, (V^G)_K) \\ &\cong \bigoplus_{i \in I} \mathrm{Hom}_K(1_K, ((V^{g_i})_{H^{g_i} \cap K})^K) \\ &\overset{\star}{\cong} \bigoplus_{i \in I} \mathrm{Hom}_{H^{g_i} \cap K}(1_{H^{g_i} \cap K}, (V^{g_i})_{H^{g_i} \cap K}) \\ &\cong \bigoplus_{i \in I} \mathrm{Hom}_{H \cap {}^{g_i}K}(1_{H \cap {}^{g_i}K}, V_{H \cap {}^{g_i}K}) \\ &\cong \bigoplus_{i \in I} \mathrm{Fix}_V(H \cap {}^{g_i}K), \end{aligned}$$

where the isomorphism '$\star$' maps $\varphi \in \mathrm{Hom}_{H^{g_i} \cap K}(1_{H^{g_i} \cap K}, (V^{g_i})_{H^{g_i} \cap K})$ by [1, Lemma III.8.6.] to

$$\left(\lambda \mapsto \sum_{j \in I_i} \varphi(\lambda k_{ij}^{-1}) k_{ij} = \varphi(\lambda) \cdot \sum_{j \in I_i} k_{ij}\right) \in \mathrm{Hom}_K(1_K, ((V^{g_i})_{H^{g_i} \cap K})^K).$$

Hence we explicitly obtain in terms of the Mackey decomposition of V^G given above:

$$V^G e = \bigoplus_{i \in I} \left(\mathrm{Fix}_V(H \cap {}^{g_i}K) \otimes g_i \sum_{j \in I_i} k_{ij}\right) = \bigoplus_{i \in I} \left(\mathrm{Im}_V(e_i) \otimes g_i \sum_{j \in I_i} k_{ij}\right),$$

where $e_i := e_{H \cap {}^{g_i}K}$ denotes the idempotent belonging to $H \cap {}^{g_i}K \leq G$.

3.2. The action of $ege \in ek[G]e$ for some $g \in G$ on $V^G e$ is derived from the action of e and that of g on V^G in terms of its Mackey decomposition. For $v \in V$, we have $(v \otimes g_i k_{ij}) \cdot g = vh' \otimes g_{i'} k_{i'j'}$, where the indices $i' \in I$, $j' \in I_{i'}$ and $h' \in H$ are uniquely determined by $g_i k_{ij} g = h' g_{i'} k_{i'j'}$. For $v \in \mathrm{Fix}_V(H \cap {}^{g_i}K)$ we hence have

$$(v \otimes g_i \sum_{j \in I_i} k_{ij}) \cdot eg = \sum_{j \in I_i} (v \otimes g_i k_{ij}) \cdot g = \sum_{j \in I_i} vh' \otimes g_{i'} k_{i'j'},$$

where the indices i', j' and $h' \in H$ of course depend on i and j.

The action of e on V^G is described as follows. For each $i \in I$ we have the decomposition

$$e = \frac{|H^{g_i} \cap K|}{|K|} \cdot e_i \cdot \sum_{j \in I_i} k_{ij} \in k[K],$$

and so for $v \in V$ this gives

$$(v \otimes g_i k_{ij}) \cdot e = (v \otimes g_i) \cdot e = \frac{|H^{g_i} \cap K|}{|K|} \cdot ve_i \otimes g_i \sum_{j \in I_i} k_{ij}.$$

4. ... and practically

4.1. Collecting coset information. Let $\mathcal{G}$ be a generating set for G given as set of permutations acting on the right cosets of H in G, i. e. we assume the permutation module $(1_H)^G$ to be given explicitly. Let us further assume K also to be given as a set $\mathcal{K}$ of permutation generators acting on the right cosets of H in G. In this paragraph we describe the part of the computations which are entirely in the context of permutation groups.

First we compute a Schreier chain for G with respect to this permutation representation. As in our application we already know the group order $|G|$, this can be done using a randomized Schreier-Sims algorithm. Such an algorithm is already available in GAP. But since we have to sift many elements of G through the Schreier chain, we have to make this easily accessible and more flexible and so we use our own GAP implementation of a randomized Schreier-Sims algorithm instead.

Now the H-K double cosets in G are in bijection with the K-orbits on $(1_H)^G$, hence suitable $\{g_i; i \in I\}$ and their decomposition into $\mathcal{G}$ are found by the Schreier-Sims algorithm. Since $H = \mathrm{Stab}_G(H \cdot 1)$ this also gives us a generating set $\mathcal{H}$ and a Schreier chain for H. Furthermore, we have $H^{g_i} \cap K = \mathrm{Stab}_K(H \cdot g_i)$, hence suitable $\{k_{ij}; j \in I_i\}$, a generating set $\mathcal{K}_i$ and a Schreier chain for $H^{g_i} \cap K$ are also found by a run of the Schreier-Sims algorithm. Conjugating gives a generating set $\mathcal{H}_i := {}^{g_i}\mathcal{K}_i$ for $H \cap {}^{g_i}K$, and sifting the elements of $\mathcal{H}_i$ through the Schreier chain for H, gives us decompositions of these elements into $\mathcal{H}$.

4.2. Precondensation. Now let k be a finite field and $V \in \text{mod–}k[H]$ be given in terms of the generating set $\mathcal{H}$ for H as a set of representing k-matrices. In this paragraph we describe those computations using V which as a precondensation step can be done once and for all elements of G to be condensed. The necessary matrix computations are done using the MeatAxe [18] or using the new finite field arithmetic [11] implemented in GAP.

By the formulae developed in 3.1, a basis of $V^G e$ is given in terms of bases for the subspaces $\mathrm{Fix}_V(H \cap {}^{g_i}K) = \mathrm{Im}_V(e_i) \leq V$. Note that we have already computed the decompositions of the elements of $\mathcal{H}_i$ into $\mathcal{H}$, hence we can find representing matrices for the elements of $\mathcal{H}_i$, too. Now, to find a basis for $\mathrm{Fix}_V(H \cap {}^{g_i}K)$, we could compute the intersection of the fixed spaces for the action of the elements of $\mathcal{H}_i$ on V, which would be a standard application of the MeatAxe. But since we do need the action of e_i on V anyway, we instead proceed as follows. To compute a representing matrix for e_i, we use the Schreier chain for $H^{g_i} \cap K$, which gives us a factorization of e_i, analogous to the factorization of e given at the end of 3.2, and allows us to enumerate the summands.

Let $V = Ve_i \oplus V(1 - e_i)$ be the vector space decomposition of V with respect to e_i, with corresponding projection π_i onto Ve_i and injection ι_i, fulfilling $\pi_i \cdot \iota_i = e_i$ and $\iota_i \cdot \pi_i = \mathrm{id}$. We have $\mathrm{Fix}_V(H \cap {}^{g_i}K) = \mathrm{Im}_V(e_i) = \mathrm{Im}(\iota_i)$, a basis of which can be found as a standard application of the MeatAxe. The matrix of basis vectors can be interpreted as the matrix of the linear map ι_i. As another standard application of the MeatAxe we obtain the matrix of π_i. We remark that the technique of precomputing the ι_i and π_i has also been used in [14].

4.3. **Condensation of group elements.** Let $g \in G$; to find the action of $ege \in ek[G]e$ on $V^G e$ we proceed as follows. Having fixed an index $i \in I$, for each index $j \in I_i$ we compute the corresponding indices i' and j' and $h' \in H$ describing the action of g as defined in 3.2. The indices are found using the i-th Schreier chain of K obtained in 4.1, this then gives us the element h' explicitly, which is decomposed into $\mathcal{H}$ using the Schreier chain of H, giving a representing matrix η'. The action of h' on $\mathrm{Fix}_V(H \cap {}^{g_i}K)$ is then described by $\iota_i \cdot \eta'$. The action of e, namely projection to the i'-th component of $V^G e$, is given by another multiplication of the latter matrix with $\pi_{i'}$.

4.4. **Remarks.** Firstly, the above computations can be simplified, if we have $H^{g_i} \cap K = \{1\}$ for some $i \in I$, i. e. if the i-th orbit is a regular K-orbit. In this case we have $\mathrm{Fix}_V(H \cap {}^{g_i}K) = V$, and hence $\iota_i = \pi_i = \mathrm{id}$. As usually K is small compared to the index $[G : H]$, regular orbits are in fact quite frequent.

Secondly, we make use of the permutation module $(1_H)^G$ because this makes the determination of H-K double cosets in G or the identification, to which H-K double coset in G or $H^{g_i} \cap K$ right coset in K an element belongs, particularly easy. This also could be done in any other representation of G where this information can be extracted in a computationally tractable manner.

4.5. **Uncondensation.** Given an $ek[G]e$-submodule $\tilde{M} \leq V^G e$, see 2.3, we use the explicitly known injections ι_i from 4.2 to find its embedding as a subspace of V^G. Using the description of V^G and the action of G thereon in terms of the Mackey decomposition, see 3.2 and 4.3, we can use the standard MeatAxe 'spinning' algorithm to find the uncondensed module of $\tilde{M}$. This has been implemented as extensions of the MeatAxe implementations [18] and [6].

5. An application: The 2-modular character table of Co_2

From now on let $G := Co_2$ be the second sporadic simple Conway group, where we follow the notation used in [3]. The determination of its 2-modular Brauer character table has been begun in [20], where 10 of the 13 irreducible 2-modular Brauer characters of Co_2 have been determined, and the remaining three characters were conjectured to have degrees 36 938, 83 948 and 156 538. In [20], too, a condensation technique has been used to find Brauer characters. The main obstacle to give a complete proof was the problem of generation of the algebra $ek[G]e$, see 2.4, as even with todays computers it seems to be impossible to do the necessary uncondensation within the permutation module examined in [20]. Now, with our new condensation algorithm at hand, we are able to circumvent the problem of generation by looking at a suitable induced module, which allows a detailed structural analysis, due to its much simpler structure, and where uncondensation can be performed efficiently, due to the much smaller dimensions involved and the powerful parallel computers available today. Finally, the above mentioned conjecture from [20] turns out to be true. Thus this completes the computation of the 2-modular Brauer character table of G.

5.1. Let $\varphi_1, \dots, \varphi_9, \varphi_{13}$ denote the known irreducible Brauer characters as given in the Brauer character table printed at the end of this note. Let ξ_{10}, ξ_{11} and ξ_{12} denote the remaining three irreducible Brauer characters. It has already been shown in [20] that the class functions φ_{10}, φ_{11} and φ_{12} are Brauer atoms and that they can be written as $\xi_i = \varphi_i + \sum_{j=1}^{9} \lambda_{ij}\varphi_j$ for $i \in \{10, 11, 12\}$ and some $\lambda_{ij} \in \mathbb{N}_0$. So our aim is to prove that all $\lambda_{ij} = 0$. For more details on the notion of basic sets and atoms we refer the reader to [8].

We now let $H := U_6(2).2$, which is the largest maximal subgroup of G and has index $[G : H] = 2300$, let $k := \mathbb{F}_2$ and $V \in \text{mod--}k[H]$ be the simple module of dimension 140, see [10] or the remark at the beginning of 5.3. We denote the Brauer character of V by ψ. Using the library of character tables and the character theoretic algorithms accessible in GAP, we find that the induced character ψ^G, which is of degree 322 000, decomposes into the set of Brauer atoms as

$$\psi^G = 10 \cdot \varphi_1 + 12 \cdot \varphi_2 + 6 \cdot \varphi_3 + 4 \cdot \varphi_4 + 4 \cdot \varphi_5 + 2 \cdot \varphi_{10} + \varphi_{11} + \varphi_{12} \, .$$

5.2. We begin our constructions by accessing the permutation representation P of G on 2 300 points from [24]. It is given there for a pair $\{a, b\}$ of standard generators for G as defined in [23].

As our condensation subgroup we choose $K := 3_+^{1+4} = O_3(N_G(3A)) \lhd N_G(3A) = 3_+^{1+4}{:}2_-^{1+4}.S_5 < G$, the maximal 3-normal subgroup of the normalizer in G of an element in the $3A$ conjugacy class. Hence this defines an unique conjugacy class of subgroups of G of order 243. A representative can be constructed as follows, using the algorithms dealing with permutation groups and soluble groups implemented in GAP. We first find a 3-Sylow subgroup $S < G$, which is of order 3^6, and then it turns out that S contains a unique normal subgroup K of isomorphism type 3_+^{1+4}.

Using GAP again we find the distribution of the elements of K into the conjugacy classes of G. Then we compute the scalar products of the restrictions to K of the φ_i and the trivial K-character. This by 2.5 gives us the condensed dimensions of the φ_i. These are given in the forth column of Table 1, where the degrees and the behavior under complex conjugation are also indicated. Especially, we conclude that the condensation idempotent $e = e_K$ is faithful.

5.3. Let P_H be the restriction of P to H. We find that V is a constituent of the 2-modular reduction of P_H, hence is obtained by a standard application of the MeatAxe. Note that this provides an alternative approach to find ψ: Let V

TABLE 1. Dimensions of condensed modules

φ	Degree	CC	Dim.
1	1	r	1
2	22	r	4
3	230	r	6
4	748	5	4
5	748	4	4
6	3 520	r	24
7	5 312	r	24
8	8 602	9	16
9	8 602	8	16
10	36 938	r	100
11	83 948	r	288
12	156 538	r	566
13	1 835 008	r	7 424

be the unique constituent of the 2-modular reduction of P_H of dimension 140, and compute its Brauer character using the MeatAxe.

We now have prepared all necessary input data to run our condensation program. It turns out that, using the new finite field arithmetic [11] implemented in GAP, the permutation group part (4.1) of our algorithm takes $10s$, the precondensation (4.2) takes $40s$ and the condensation (4.3) of one group element takes $498s$ of CPU time on a SPARC 20 machine. As condensation algebra we take

$$\mathcal{C} := \langle eae, ebe, eabe, ebae \rangle_{k-\text{algebra}} \leq ek[G]e,$$

and we let $M := V^G e|_{\mathcal{C}}$ denote the restriction of the $ek[G]e$-module $V^G e$ to $\mathcal{C}$. The $\mathcal{C}$-module M has dimension 1 180 and the MeatAxe finds the following constituents:

$$1a^{10}, 4a^{12}, 4b^4, 4c^4, 6a^6, 100a^2, 288a^1, 566a^1,$$

where we denote the constituents by their dimensions and lower case letters, and multiplicities by superscripts. The dimensions and multiplicities coincide with the decomposition of ψ^G into Brauer atoms given in 5.1 and Table 1, so we can already be quite confident that the all the Brauer atoms are in fact Brauer characters. In order to prove this, we are now going to analyze the $\mathcal{C}$-module M in more detail, using the ideas and algorithms developed in [13], which we assume the reader to be familiar with.

5.4. **Ad φ_{10}.** Using the techniques developed in [13], we find that M contains exactly two $100a$-local submodules, L^1_{100a} and L^2_{100a}. Furthermore, for each local submodule $L \leq M$, we find that $L^1_{100a} \leq L \leq L^2_{100a}$ holds. Hence we have $\mathrm{Soc}(M) < \mathrm{Rad}(M)$ and $\mathrm{Soc}(M) \cong M/\mathrm{Rad}(M) \cong 100a$ is simple.

Let S denote the condensed $ek[G]e$-module of ξ_{10}. Table 1 shows that $100a$ occurs with multiplicity 1 as a constituent of $S|_{\mathcal{C}}$. Hence $S|_{\mathcal{C}}$ is indecomposable, and $100a$ is a socle and a radical quotient constituent of $S|_{\mathcal{C}}$. This means we have $S|_{\mathcal{C}} \cong 100a$, hence $\varphi_{10} = \xi_{10}$.

5.5. **Ad φ_{11}.** Let $L_{288a} < M$ denote the unique $288a$-local submodule. It turns out to have dimension 439 and has the following constituents:

$$1a^3, 4a^5, 4b^3, 4c^1, 6a^2, 100a^1, 288a^1.$$

Hence, to prove that $\varphi_{11} = \xi_{11}$ holds, it is sufficient to show that the uncondensed module of L_{288a} in V^G has dimension 124 451 and the uncondensed module of $\mathrm{Rad}(L_{288a})$ has dimension 40 503, since this means that the uncondensed module of L_{288a} has a simple radical quotient module of dimension 83 948.

The latter conditions have been checked using the uncondensation algorithm and the 'spinning' algorithm for induced modules incorporated into the MeatAxe implementation [6]. The computations have taken a total of about 1 000 hours of CPU time using between four to eight nodes on the SP/2 of the computing center of the University of Essen and indeed resulted in two invariant subspaces of the expected dimensions. This proves that $\varphi_{11} = \xi_{11}$ holds.

5.6. **Ad φ_{12}.** Let L_{566a} denote the unique $566a$-local submodule of M. It turns out to have dimension 705. Unfortunately, the analysis of the constituents of L_{566a} shows that the uncondensed module of L_{566a} in V^G would have dimension at least 196 249. To avoid the explicit construction of this uncondensed module we instead consider the submodule

$$N := \sum\{L \leq M \text{ local}; L_{566a} \not\leq L\} < M.$$

It turns out that N has dimension 475 and the following constituents:

$$1a^7, 4a^9, 4b^3, 4c^2, 6a^4, 100a^1, 288a^1.$$

Hence N is a $ek[G]e$-submodule of V^Ge if and only if its uncondensed module has dimension 125 751. Since we have $L_{288a} \leq N$, the same inclusion must hold for the corresponding uncondensed modules. Again using the parallel MeatAxe implementation [6], we have therefore computed the uncondensed module of N by extending the uncondensed module of L_{288a} found in 5.5. Again we have found an invariant subspace of the dimension we have expected, hence N is a $ek[G]e$-submodule of V^Ge.

Using this information we now proceed as follows. From the list of constituents of N we see that $L_{566a} \not\leq N$ holds; note that this could have been deduced in advance using Remark 5.7. By the structure theory of modular lattices, see [2, 13], it follows that $(L_{566a}+N)/N = \mathrm{Soc}(M/N) \cong 566a$ holds. Next we note that V is a self-dual $k[H]$-module, hence V^G is a self-dual $k[G]$-module and V^Ge is a self-dual $ek[G]e$-module, see 2.2. By the above analysis, L_{566a} is

TABLE 2. The character table of Co_2 mod 2

	ind	1A	3A	3B	5A	5B	7A	9A	11A	15A	15B	15C	23A	23B
χ_1	+	1	1	1	1	1	1	1	1	1	1	1	1	1
χ_2	+	22	-5	4	-3	2	1	1	0	-1	0	0	-1	-1
χ_3	+	230	14	5	5	0	-1	-1	-1	0	-1	-1	0	0
χ_4	o	748	-8	1	-2	-2	-1	-2	0	1	b15	⋆⋆	-b23	⋆⋆
χ_5	o	748	-8	1	-2	-2	-1	-2	0	1	⋆⋆	b15	⋆⋆	-b23
χ_6	+	3520	-44	10	-5	0	-1	1	0	0	1	1	1	1
χ_7	+	5312	20	2	12	-3	-1	-1	-1	-3	0	0	-1	-1
χ_8	o	8602	43	-20	2	-3	-1	1	0	0	-b15	⋆⋆	0	0
χ_9	o	8602	43	-20	2	-3	-1	1	0	0	⋆⋆	-b15	0	0
χ_{10}	?	36938	-79	-52	13	-2	-1	2	0	-2	1	1	0	0
χ_{11}	?	83948	-22	-58	-2	-12	-3	-1	-4	-3	-2	-2	-2	-2
χ_{12}	?	156538	100	-80	-12	-2	-3	-2	-3	-5	0	0	0	0
χ_{13}	+	1835008	-128	-128	8	8	0	-2	-1	2	2	2	-1	-1

the image of N under the duality anti-automorphism of the $ek[G]e$-submodule lattice of $V^G e$. Hence L_{566a} is a $ek[G]e$-submodule of $V^G e$. Finally, we have $\mathrm{Rad}(L_{566a}) = L_{566a} \cap N$. Hence $\mathrm{Rad}(L_{566a})$ also is an $ek[G]e$-submodule of $V^G e$ and thus $\varphi_{12} = \xi_{12}$ holds.

5.7. **Concluding remarks.** The fact $L_{566a} \not\leq N$ used in 5.6 could have been deduced in advance from the following statement, we only give without proof.

Let $M \in$ mod–A, let $L \leq M$ be a local submodule and $M_1, M_2 \leq M$ such that $L \not\leq M_1, M_2$, but $L \leq M_1 + M_2$. Then there are local submodules $L_1 \leq M_1$, $L_2 \leq M_2$ such that $L+L_1 = L+L_2 = L_1+L_2$ holds, i. e. there is a dotted-line, see [2, 13], through L connecting M_1 and M_2.

Finally, it now remains to determine the 2-modular Frobenius-Schur indicators, see [9, Section 9], for the irreducible Brauer characters now proven to exist. We expect the methods developed here should be sufficient to solve this problem, too; this, however, is work still under progress.

References

[1] J. Alperin: Local representation theory, Cambridge University Press, 1986.

[2] D. Benson, J. Conway: Diagrams for modular lattices, J. Pure Appl. Alg. **37**, 1985, 111–116.

[3] J. Conway, R. Curtis, S. Norton, R. Parker, R. Wilson: Atlas of finite groups, Clarendon Press, 1985.

[4] G. Cooperman, M. Tselman: New sequential and parallel algorithms for generating high dimension Hecke algebras using the condensation technique, *in*: Proc. of International Symposium on Symbolic and Algebraic Computation (ISSAC '96), ACM Press, 1996, 155–160.

[5] C. Curtis, I. Reiner: Methods of representation theory I, Wiley, 1981.

[6] P. Fleischmann, G. Michler, P. Roelse, J. Rosenboom, R. Staszewski, C. Wagner, M. Weller: Linear algebra over small finite fields on parallel machines, Vorlesungen aus dem Fachbereich Mathematik der Universität GH Essen **23**, 1995.
[7] J. Green: Polynomial representations of GL_n, Lecture Notes in Mathematics **830**, Springer, 1980.
[8] G. Hiss, C. Jansen, K. Lux, R. Parker: Computational modular character theory, unpublished manuscript.
[9] C. Jansen, K. Lux, R. Parker, R. Wilson: An atlas of Brauer characters, Clarendon Press, 1995.
[10] N. N.: An atlas of Brauer characters, part II, in preparation.
[11] N. Kim: Implementierung der MeatAxe in das Computeralgebra-System GAP unter besonderer Berücksichtigung einer schnellen Vektorarithmetik, Diplomarbeit, RWTH Aachen, 1997.
[12] K. Lux: Algorithmic methods in modular representation theory, Habilitationsschrift, RWTH Aachen, 1997.
[13] K. Lux, J. Müller, M. Ringe: Peakword condensation and submodule lattices: an application of the MeatAxe, J. Symb. Comp. **17**, 1994, 529–544.
[14] K. Lux, M. Wiegelmann: Condensing tensor product modules, *in*: R. Wilson (ed.): The Atlas Ten Years On, Birmingham, 1995, Proceedings, to appear.
[15] R. Parker: The computer calculation of modular characters (The MeatAxe), *in*: M. Atkinson (ed.): Computational Group Theory, 1984, 267–274.
[16] R. Parker, R. Wilson: Unpublished.
[17] A. Ryba: Condensation programs and their application to the decomposition of modular representations, J. Symb. Comp. **9**, 1990, 591–600.
[18] M. Ringe: The C–MeatAxe, Manual, RWTH Aachen, 1994.
[19] M. Schönert et. al.: GAP — Groups, Algorithms and Programming, Manual, RWTH Aachen, 1995.
[20] I. Suleiman and R. Wilson: The 2-modular characters of Conway's group Co_2, Math. Proc. Camb. Philos. Soc. **116**(2), 1994, 275–283.
[21] J. Thackray: Modular representations of some finite groups, PhD thesis, Cambridge University, 1981.
[22] M. Wiegelmann: Fixpunktkondensation von Tensorproduktmoduln, Diplomarbeit, RWTH Aachen, 1994.
[23] R. Wilson: Standard generators for sporadic simple groups, J. Algebra **184**, 1996, 505–515.
[24] R. Wilson: ATLAS of finite group representations, URL <http://www.mat.bham.ac.uk/atlas>.

Lehrstuhl D für Mathematik, RWTH Aachen, Templergraben 64, D-52062 Aachen, Germany
E-mail address: juergen.mueller@math.rwth-aachen.de

IEM, University of Essen, Ellernstrasse 29, 45326 Essen, Germany
E-mail address: jens@exp-math.uni-essen.de