THE BRAUER TREE OF THE PRINCIPAL 19-BLOCK OF THE SPORADIC SIMPLE THOMPSON GROUP

GENE COOPERMAN, GERHARD HISS, KLAUS LUX, AND JÜRGEN MÜLLER

ABSTRACT. This paper completes the construction of the Brauer tree of the sporadic simple Thompson group in characteristic 19. Our main computational tool to arrive at this result is a new parallel implementation of the DirectCondense method.

1. INTRODUCTION AND RESULTS

Let Th denote the sporadic simple Thompson group. In [6] the Brauer tree of the principal 19-block of Th has been determined up to two possibilities. In this note, we show which of these is the correct one, and we describe the new computational techniques which enabled us to decide between these two possibilities. We believe that the methods presented here will be powerful enough to solve even more difficult problems in the modular character theory of the sporadic groups.

As a general reference for the theory of blocks of cyclic defect, the interpretation of a Brauer tree and its planar embedding see the introduction of [6]. The planar embedded Brauer tree of the principal 19-block of Th is given in Table 1, it coincides with the tree given in [6, p. 277, Case I]. Its nodes are labelled by the numbers of the corresponding ordinary irreducible characters, where we use the notation for the ordinary irreducible characters of Th as is given in [3, p. 176], and can also be accessed in GAP [14]. In Table 2 we list the ordinary irreducible characters of G lying in the principal 19-block, plus some additional information concerning these. The column headed "CC" contains the entry "r" in rows corresponding to real valued characters. Otherwise it contains the number of the complex conjugate character. The last column of Table 2 contains the values of the characters on elements of class 19A. Characters which are connected on the Brauer tree must have unequal values on this class. The degrees of the irreducible Brauer characters are given in Table 3. A Brauer character corresponding to an edge of the tree connecting χ_i and χ_j, with $i < j$ is denoted by ϕ_i.

To obtain the result of this paper, we had to apply a new condensation technique to a module of dimension 976 841 775, the permutation module on the cosets of the third maximal subgroup of Th. The condensed module has dimension 1403 over $\mathbb{F}_{19}$ and can be analyzed with the MeatAxe [10], giving the result. The details are given in Section 2. We remark that in order to arrive at only two possibilities for the Brauer tree in [6], we had to rule out several other possibilities using sophisticated techniques involving Green correspondence. We have checked the results of the condensation against these other possible trees. None of them is consistent with the condensation results.

Table 1.   The Brauer Tree mod 19 of Th
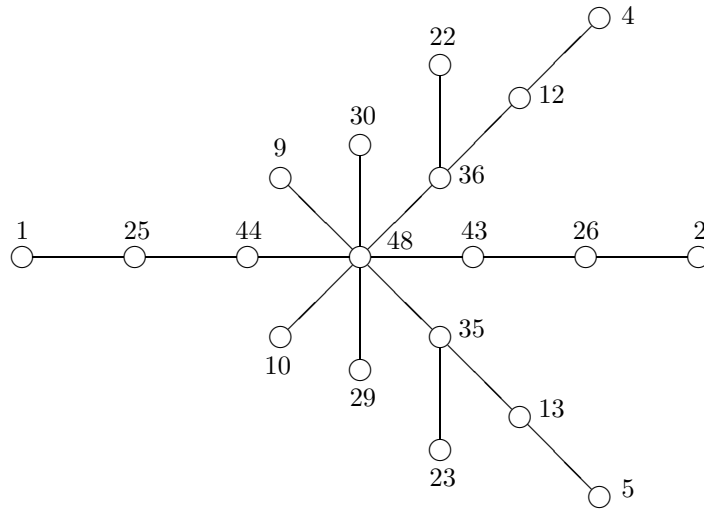
Table 2.   The Characters in the Principal 19-Block of Th

χ	Degree	CC	19A
1	1	r	1
2	248	r	1
4	27 000	5	1
5	27 000	4	1
9	85 995	10	1
10	85 995	9	1
12	767 637	13	−1
13	767 637	12	−1
22	4 096 000	23	−1

χ	Degree	CC	19A
23	4 096 000	22	−1
25	4 881 384	r	−1
26	4 936 750	r	−1
29	6 696 000	30	1
30	6 696 000	29	1
35	21 326 760	36	1
36	21 326 760	35	1
43	76 271 625	r	1
44	77 376 000	r	1
48	190 373 976	r	−1

The condensation method has originally been conceived by Parker and Thackray, see [15], and is also described in [13] and [8]. The idea of the new DirectCondense technique goes back to [11], and a fuller discussion is given in [5]. In Section 3 we give details on the implementation and special techniques for doing the condensation in practice.

2. Condensation

2.1. **General remarks.** Let A be finite dimensional algebra over the field F and $e \in A$ be an idempotent. Let mod-A denote the category of finitely generated right A-modules. Then the exact functor

$$? \otimes_A Ae : \mathrm{mod}\text{-}A \longrightarrow \mathrm{mod}\text{-}eAe : M \longmapsto Me$$

TABLE 3.   The Degrees of the Irreducible Brauer Characters

Char.	Degree	Char.	Degree
ϕ_1	1	ϕ_{23}	4 096 000
ϕ_2	248	ϕ_{25}	4 881 383
ϕ_4	27 000	ϕ_{26}	4 936 502
ϕ_5	27 000	ϕ_{29}	6 696 000
ϕ_9	85 995	ϕ_{30}	6 696 000
ϕ_{10}	85 995	ϕ_{35}	16 490 123
ϕ_{12}	740 637	ϕ_{36}	16 490 123
ϕ_{13}	740 637	ϕ_{43}	71 335 123
ϕ_{22}	4 096 000	ϕ_{44}	72 494 617

is called the *condensation functor* with respect to e, and Me is called the *condensation* of M. Condensation gives us a tool to analyse the submodule structure of M by looking at Me instead. We will apply this for F a field of prime characteristic p, $A = FG$, where G is a finite group, K the *condensation subgroup* of G having order $|K|$ prime to p, and

$$e = e_K = |K|^{-1} \cdot \sum_{g \in K} g \in FG.$$

If M is an FG-module, then Me is the subset of M consisting of the elements left fixed by K. Let ϕ denote the Brauer character of M, which is extended arbitrarily to a class function $\tilde{\phi}$ on G. Then we have

$$\dim_F(Me) = (\phi|_K, 1_K)_K = (\tilde{\phi}, 1_K^G)_G,$$

where $(\cdot, \cdot)_G$ denotes the scalar product for class functions. As each Brauer character can be written as a $\mathbb{Z}$-linear combination of ordinary characters restricted to the p-regular conjugacy classes, the computation of these scalar products can be carried out entirely in terms of ordinary characters. If the block under consideration is described by a Brauer tree, these linear combinations can directly be read off from the tree.

If P is an FG-permutation module on the finite set Ω, then the condensed module Pe can be described as follows. Let $\{O_i\}$ be the set of K-orbits on Ω, and $\bar{O}_i := \sum_{\omega \in O_i} \omega \in P$ be the orbit sums. Then $\{\bar{O}_i\}$ is an F-basis of Pe, and for $g \in G$ the action of $ege \in eFGe$ on Pe is given as

$$\bar{O}_i \cdot ege = \sum_j a_{ij}(g) \cdot |O_j|^{-1} \cdot \bar{O}_j, \text{ where } a_{ij}(g) = |\{\omega \in O_i : \ \omega g \in O_j\}|.$$

Hence to find the action of ege, we are reduced to find the K-orbits on Ω, their lengths, and the $a_{ij}(g)$.

2.2. **The Thompson group.** Now we let $G := Th$. For the necessary group theoretic information we refer the reader to [3]. We start our constructions with the irreducible representation D_{248} over $\mathbb{F}_2$ which is the 2-modular reduction of the irreducible ordinary representation of the same degree. Matrices for the action of two generators $A, B \in G$ on the module underlying D_{248} can be found in the group

representation library [16]. Here, A is a $2A$ element, B is a $3A$ element, and their product AB is a $19A$ element. This is a rationally rigid triple for G, see [17, 9]. The following computations are carried out using D_{248} with the help of the MeatAxe and GAP.

We now let $P_1 := AB$, $P_2 := AB^2$, and $H := \langle H_1, H_2 \rangle$, where

$$H_1 := (P_1 P_2 P_1 P_2^2 P_1 P_2)^2 P_2, \quad H_2 := (H_1^{18} P_1^{-3} A P_1^3)^{18}.$$

Furthermore, we let $K := \langle K_1, K_2 \rangle$, where $K_1 := H_1$, $K_2 := Q_2 Q_1 Q_2^{-1}$, and

$$Q_1 := H_1 H_2 H_1 H_2^2 (H_1 H_2)^2 (H_1 H_2^2)^2, \quad Q_2 := (H_2 H_1)^2 H_1^3 (H_2 H_1)^3.$$

We are going to choose K as our condensation subgroup. Hence we have to show that $|K|$ is not divisible by 19, and we have to find the scalar products $(\chi, 1_K^G)_G$ for all irreducible ordinary characters of G.

We find that H_1 is of order 36, and that both H_1, H_2 centralize H_1^{18}. Hence $H \leq C_G(H_1^{18}) \cong 2_+^{1+8} \cdot A_9$. We will show, that H equals that centralizer. First we collect a few elements in H generating a normal subgroup $O \leq H$ of order 2^9. Then it turns out that $D_{248}|_H$ has two different constituents of dimension 8, $8a$ and $8b$ say, where $8b$ restricts irreducibly to K, whereas $8a$ has a fixed space of dimension 1. Now the vector fixed by K yields an orbit of length 120 under the action of H, yielding a permutation representation P_{120} of H. This permutation group turns out to be of order $181\,440 = |A_9|$. As O is contained in the kernel of P_{120}, we conclude that $H = C_G(H_1^{18})$.

By construction, we know that K is contained in a subgroup $2_+^{1+8}.L_2(8) : 3$ of H, which is of index 120 in H. Using the action of $D_{248}|_K$ on an orbit of a suitable vector, we find that $|K| \geq 774\,144$, hence we have $K \cong 2_+^{1+8}.L_2(8) : 3$.

To compute the scalar products $(\chi, 1_K^G)_G$, we first observe that $(\chi, 1_K^G)_G = (\chi_H, 1_K^H)$ holds. The character tables of G and H are accessible in GAP, and it turns out that the fusion map of the conjugacy classes of H into those of G is uniquely determined up to table automorphisms of G. Furthermore, as the normal subgroup O of H is contained in K, it is enough to find the fusion map from $L_2(8) : 3$ to A_9, which is induced by the fusion map from K to H. Hence we choose a factor fusion map from the character table of H to that of A_9. Having fixed such a map, the admissible table automorphisms of A_9 are those which leave the chosen fusion map invariant. It now turns out that there are exactly two possible fusion maps from $L_2(8) : 3$ to A_9, and these are not conjugate under the action of the admissible table automorphisms. A look at the classes of A_9 which are the images of the classes of elements of order 36 under the factor fusion map shows, that these classes in turn are in the image of exactly one of the candidate fusion maps from $L_2(8) : 3$ to A_9. Now the scalar products $(\chi, 1_K^G)_G$ for all irreducible ordinary characters of G can be computed. They are given in Table 4.

The module we are going to condense is the FG-permutation module P on the cosets of H, where $F := \mathbb{F}_{19}$. It is found as the 19-modular reduction of the corresponding $\mathbb{Z}G$-permutation module. The latter in turn is found as the action of G on the orbit of a nontrivial vector in the module underlying D_{248} which is fixed by H. Using the MeatAxe, such a vector is found to exist and to be uniquely determined. The orbit is of length $976\,841\,775$, which is the index of H in G. From Table 4, where the scalar products $(\chi, 1_H^G)_G$ are given, we compute $\dim_F(Pe) = 1403$. A star in the second column of Table 4 indicates a character in the principal block of FG.

TABLE 4. Two Permutation Characters of G

χ	pb	H	K	χ	pb	H	K	χ	pb	H	K
1	$\star$	1	1	17		.	.	33		2	31
2	$\star$	.	.	18		.	.	34		.	25
3		.	.	19		1	9	35	$\star$	.	27
4	$\star$	.	.	20		.	2	36	$\star$	.	27
5	$\star$	.	.	21		1	10	37		1	43
6		.	.	22	$\star$	.	5	38		1	43
7		1	3	23	$\star$	.	5	39		2	62
8		1	2	24		1	8	40		2	70
9	$\star$	.	.	25	$\star$	2	15	41		1	73
10	$\star$	.	.	26	$\star$	.	8	42		1	86
11		1	3	27		.	6	43	$\star$	.	95
12	$\star$	.	2	28		.	6	44	$\star$	2	111
13	$\star$	.	2	29	$\star$	.	6	45		.	100
14		.	.	30	$\star$	.	6	46		1	119
15		.	.	31		.	15	47		1	147
16		.	1	32		2	21	48	$\star$	1	236

TABLE 5. Result of Condensation

Dim.	Mlp.	Dim.	Mlp.	Dim.	Mlp.
$1a$	3	$10a$	1	$62a$	2
$2a$	1	$14a$	4	$70a$	2
$3a$	1	$20a$	1	$73a$	1
$3b$	1	$20b$	1	$86a$	1
$6a$	1	$21a$	2	$87a$	1
$6b$	1	$31a$	2	$97a$	3
$8a$	1	$43a$	1	$119a$	1
$9a$	1	$43b$	1	$147a$	1

Let $A, B \in G$ denote the elements introduced above. In Section 3 we will describe how the action of A, B on the orbit of the seed vector and matrices M_{eAe} and M_{eBe} giving the action of eAe and eBe on Pe are computed actually. We consider the subalgebra

$$\mathcal{K} := \langle eAe, eBe \rangle \leq eFGe.$$

In fact there is no reason why $\mathcal{K}$ should not be equal to $eFGe$, but we do not know whether this is the case. Using the MeatAxe, with the two matrices M_{eAe} and M_{eBe}, we find the $\mathcal{K}$-constituents of Pe. Their dimensions and their multiplicities are given in Table 5.

TABLE 6.   Degrees of Irreducible Brauer Characters: Case I

Char.	Degree	d_i	Char.	Degree	d_i
ϕ_1	1	1	ϕ_{23}	4 096 000	5
ϕ_2	248	.	ϕ_{25}	4 881 383	14
ϕ_4	27 000	.	ϕ_{26}	4 936 502	8
ϕ_5	27 000	.	ϕ_{29}	6 696 000	6
ϕ_9	85 995	.	ϕ_{30}	6 696 000	6
ϕ_{10}	85 995	.	ϕ_{35}	16 490 123	20
ϕ_{12}	740 637	2	ϕ_{36}	16 490 123	20
ϕ_{13}	740 637	2	ϕ_{43}	71 335 123	87
ϕ_{22}	4 096 000	5	ϕ_{44}	72 494 617	97

2.3. **Proof of the result.** We have to decide between two possible Brauer trees for the principal 19-block of G. The first possibility, Case I, is the tree given in the introduction, which will turn out to be the correct one. The second possibility, Case II, is given below.
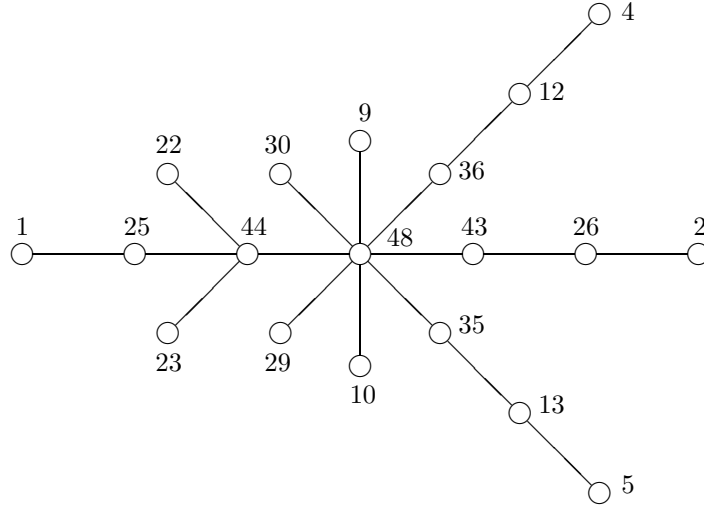
The degrees of the irreducible Brauer characters ϕ_i in the principal block and the dimensions d_i of the corresponding condensed modules for Case I are given in Table 6. These can be computed from the Brauer tree and Table 4. For example, in Case I, $\phi_{44} = \chi_{44} - \chi_{25} + \chi_1$ on 19-regular elements of G. Hence $d_{44} = 111 - 15 + 1 = 97$. In Case II, we have $\phi_{35} = \phi_{36} = 20\,586\,123$, $\phi_{44} = 64\,302\,617$, $d_{35} = d_{36} = 25$, and $d_{44} = 87$. All other degrees and dimensions remain unchanged.

By Tables 4 and 6, the $eFGe$-module Pe has unique composition factors of degrees 147 and 119, respectively. These correspond to the two defect 0 characters χ_{46} and χ_{47}. All other $eFGe$-composition factors of Pe have a smaller dimension, in fact at most 97. It now follows from Table 5, that the irreducible $eFGe$-modules of dimensions 147 and 119 restrict irreducibly to $\mathcal{K}$. Finally, Table 5 shows that

there is an $eFGe$-composition factor of Pe of dimension at least 97, different from the ones of dimensions 147 and 119. This implies that Case I is correct.

3. COMPUTATIONS

The key idea, which goes back to [11], of the condensation method applied here is to use the interpretation of the abstract set Ω, where G acts on, as a set of vectors in a space, where G acts on linearly. This gives us a compact and efficient way to compute the K-orbits and the action of group elements $g \in G$. A fuller discussion of this method is contained in [5]. The program from that description was used as the base, but performance enhancements, modifications for robustness and ease of use, and checkpointing facilities had to be added to make it work for the case of $G = Th$ in its 248-dimensional representation D_{248} over $\mathbb{F}_2$. This section concentrates on these implementation issues.

3.1. **Algorithm.** The vectors in $\mathbb{F}_2^{248}$ are stored with eight entries per Byte. So, a vector requires 31 Bytes. Since the machines having been used have 64-bit words, the vectors are in fact stored in 32 Bytes each. Matrix-vector multiplication over $\mathbb{F}_2$ is implemented using logical bit operations. A lookup table of all linear combinations of each set of four adjacent rows under $\mathbb{F}_2$ is kept for each matrix. Such a lookup table was first used by Arlazarov et al. [1, 2, p. 245], and was popularized by Parker [10] in his software for the MeatAxe.

The natural algorithm would have been to maintain a hash table for all the 976 841 775 vectors. Each vector would be stored in the table, along with an index indicating which orbit it comes from. At 32 Bytes per vector, plus hash table overhead, this would imply the use of more than 32 Gigabytes, which would be far from possible at our site.

The solution is to store only $1/m$ of the vectors, where $m = 64$ in our case, thus allowing the entire computation to proceed within the available virtual memory. This is accomplished by applying a second hash-like function, and by saving in the hash table only those vectors for which the function returns zero modulo m. There are a few small orbits that do not have such vectors. In the case of such orbits, all of the vectors are stored in the hash table. Since K has order 774 144, it is always feasible to place all vectors of a single orbit into a temporary hash table of size less than 32 Megabytes.

This decision to store only some of the vectors affects another part of the natural algorithm. At several stages in the algorithm, one must determine if a new vector, ωg, has previously been encountered, and if it has been encountered, what is the index of the orbit to which it belongs. The new solution requires one to do a local search of vectors in the orbit $\omega g K$ until one either finds a vector whose second hash value is 0 modulo m, or else until all vectors of the orbit $\omega g K$ have been found. At this point, one has found a vector ω' in the same orbit as ωg, which either is contained in the main hash table, and one can look up the index of the orbit, or one adds ω' to a queue of representatives of new orbits to be explored. Note that one is able to determine a vector which should be contained in the hash table even without having access to the hash table. This is an important point in section 3.2, where the distributed version of the algorithm is described.

The other complication of the implementation was the decision whether to first store the vectors in the hash table, and then compute the matrix entries a_{ij}, see 2.1, or else to compute both at once. For efficiency reasons, it was decided to compute

the two at the same time. Thus, when an image vector ωg is computed, one next determines whether a vector from the corresponding orbit is in the hash table. If this is the case, one increments the appropriate a_{ij} entry. But if it was not part of a known orbit, when does one increment an appropriate a_{ij}? Further, one must be careful that the number of images ωg in the queue representing new orbits does not grow too large. This is especially worrisome at the beginning, when many points of new orbits are found, and for an unknown orbit there may be a large number of member vectors waiting in the queue. Ensuring proper accounting and memory restrictions is what leads to complications.

The same code is then used to condense the second and further matrices, maintaining the orbit data structures found during the condensation of the first matrix.

3.2. **Parallelization.** The parallelization was carried out through a master-slave architecture. The code was considerably simplified by the use of the STAR/MPI software [4], a transaction-oriented parallel LISP. The computation used GCL Common LISP. This parallel tool is based on the MPICH implementation of MPI. At Northeastern University, a small "homegrown" subset of MPI has been implemented by one of us (G. C.) and R. Kyzas as a teaching tool, which replaced MPICH in part of the computations.

This software helped to handle such concurrency issues as when two distinct slaves were exploring the same orbit with two distinct orbit representatives from the queue of new orbits.

3.3. **Timing.** The computations were done using eight 75 MHz Alpha 3000/300 workstations at Northeastern University. While the seven slave workstations had 64 Megabytes and 300 Megabytes of virtual memory, the master had 192 Megabytes and 1 Gigabyte of virtual memory, Timing tests indicate that a matrix-vector multiplication costs about 23 microseconds.

The computation took about one month on eight workstations. The calculation was roughly divided into 10 days to determine the orbits, 10 days for the condensation of the first generator with respect to those orbits, and 10 days for the second generator. The first two parts of the computation were mixed together, as described in section 3.1, although the majority of the first 10 days was still spent in building new orbits and the majority of the second 10 days in computing the condensation of the first generator.

There are two constraints on the speed of the computation: the number of slaves, or equivalently, the CPU speed of the slaves, and semiconductor memory on the master. Determination of the orbits during the first 10 days was primarily constrained by the available CPU power. This is because the number of matrix-vector multiplications in the computation of the points of an orbit O_i is roughly proportional to $|O_i|k$, where k is the size of the generating set for K. Yet, $|O_i|/m$ points are needed to be stored on the master. Thus, the slave carries out km matrix-vector multiplications for each hash access on the master. In the second phase, condensation of a generator with respect to those orbits, requires on average approximately $|O_i|m$ matrix-vector multiplications, while $|O_i|$ points need to be accessed on the master from the hash array. Hence the slave carries out m matrix-vector multiplications for each hash access on the master. In addition, in the second phase, the hash table is almost full and one must also update the matrix associated with the condensation. Thus, there are still more demands on memory.

The memory demands for the hash array are proportional, of course, to $|\Omega|/m$. The factor $m = 64$ was chosen precisely to balance these competing demands of memory and CPU time. It was found empirically that it was advantageous to choose m such that the hash array was larger than RAM, and accept a certain paging penalty. In our example, the hash array occupied 610 Megabytes.

References

[1] A. V. Aho, J. E. Hopcroft and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974.

[2] V. L. Arlazarov, E. A. Dinic, M. A. Kronrod and I. A. Faradzev, On Economical Construction of the Transitive Closure of a Directed Graph, *Dokl. Nauk SSSR* **194**:3 (1970), pp. 487–488, English translation *in: Soviet Math. Dokl.* **11**:5 (1970), pp. 1209–1210.

[3] J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker, and R. A. Wilson, Atlas of Finite Groups, Oxford University Press, London, 1985.

[4] G. Cooperman, STAR/MPI: Binding a Parallel Library to Interactive Symbolic Algebra Systems, *Proc. of International Symposium on Symbolic and Algebraic Computation (ISSAC '95)*, ACM Press, pp. 126–132.

[5] G. Cooperman and M. Tselman, New Sequential and Parallel Algorithms for Generating High Dimension Hecke Algebras using the Condensation Technique, *Proc. of International Symposium on Symbolic and Algebraic Computation (ISSAC '96)*, ACM Press, 1996, pp. 155–160.

[6] G. Hiss and K. Lux, Brauer Trees of Sporadic Groups, Oxford University Press, Oxford, 1989.

[7] C. Jansen, K. Lux, R. A. Parker, and R. A. Wilson, An Atlas of Brauer Characters, Oxford University Press, 1995.

[8] K. Lux, J. Müller, and M. Ringe, Peakword condensation and submodule lattices: an application of the MeatAxe, *J. Symb. Comp.* **17** (1994), 529–544.

[9] H. Pahlings, Realizing finite groups as Galois groups, *Bayreuth. Math. Schr.* **33** (1990), 137–152.

[10] R. A. Parker, The computer calculation of modular characters (the Meat-Axe), *in:* Computational Group Theory, Academic Press, London, 1984, pp. 267–274.

[11] R. A. Parker and R. A. Wilson, Private communication.

[12] M. Ringe, The C-MeatAxe, Release 2.2.0., Handbuch, Lehrstuhl D für Mathematik, RWTH Aachen, Germany, 1994.

[13] A. J. E. Ryba, Computer condensation of modular representations, *J. Symbol. Comput.* **9** (1990), 591–600.

[14] M. Schönert et al., GAP — Groups, Algorithms, and Programming, Lehrstuhl D für Mathematik, RWTH Aachen, Germany, 1994.

[15] J. G. Thackray, Modular representations of finite groups, Ph. D. Thesis, Cambridge University, 1981.

[16] R. A. Wilson, http://www.mat.bham.ac.uk/atlas/, University of Birmingham, UK, 1996.

[17] R. A. Wilson, Standard generators for sporadic simple groups, *J. Algebra* **184** (1996), 505–515.

G.C.: College of Computer Science, Northeastern University, Boston, MA 02115, USA

G.H., J.M.: IWR der Universität Heidelberg, Im Neuenheimer Feld 368, 69120 Heidelberg, Germany

K.L.: Lehrstuhl D für Mathematik, RWTH Aachen, Templergraben 64, 52062 Aachen, Germany

G.C.: e-mail: gene@ccs.neu.edu

G.H.: e-mail: hiss@euterpe.iwr.uni-heidelberg.de

K.L.: e-mail: klux@Math.RWTH-Aachen.DE

J.M.: E-MAIL: JMUELLER@EUTERPE.IWR.UNI-HEIDELBERG.DE