Improved Methods to Calculate the Characters of the Symmetric Group

By Stig Comét

1. Introduction. It was shown by Bivins and others [1] how the characters χ_κ^ρ of the symmetric group of degree N may be calculated with the aid of an electronic computer. By repeated use of a recursion formula due to Murnaghan (ref. in [1]), the characters are finally expressed by means of special characters, with $\kappa = 1^N$, for which an explicit formula exists, due to Frobenius. The same principle was applied in a program written for the electronic computer BESK at Stockholm. As was indicated at an early stage [2], I have used the Nakayama version of the recursion formula in this program, thus avoiding some of the trials giving zero terms. Aiming at economy with respect to storage space, in order to permit as high values of N as possible, I tried different ways of representing partitions binarily; some of them were described in [3]. The one which was then called Second Method turned out to be most convenient. Some details of the program are described in §4 as an introduction to the use of this method. The general plan of the calculation is given in §3. Some observations from the runnings are reported and briefly discussed in §5.

It should be mentioned here that the program has been run at BESK, free of cost, during periods when machine time has been available. I am very grateful to Matematikmaskinnämnden, The Swedish Board for Computing Machinery, for this favor.

The employed notation for partitions is a useful tool even for certain theoretical considerations. This fact was developed recently [4], and we shall quote, in §6, some of the results obtained. Further, in §7 and §8, we shall deduce a few formulas, expected to simplify the character calculations.

2. Definitions, Notations, and Formulas. We denote by χ_κ^ρ a simple character of the symmetric group of degree N, remembering that this character is completely identified by two partitions, ρ and κ, of the integer N. One of them, written as a subscript, indicates the class of conjugate group elements to which the character corresponds, the other one, written as a superscript, determines the irreducible representation from which the character emanates, as the trace of the representing matrix. In particular, the partitions $\rho = (N)$ and $\rho = (1^N)$ specify the unit and the alternating representations, respectively.

The class $\kappa = (1^N)$ consists of one element only, namely the unit element of the group. In a representation indicated by the partition $\rho = (r_1, r_2, \cdots, r_m)$, where $r_1 \geqq r_2 \geqq \cdots \geqq r_m \geqq 0$, the corresponding character is obtained from the Frobenius formula

$$\chi_{(1^N)}^{\rho} = \frac{N!}{R_1!R_2!\cdots R_m!} \prod_{(i<j)} (R_i - R_j) \tag{1}$$

where

$$R_i = r_i + m - i; \qquad i = 1, 2, \cdots, m. \tag{2}$$

Received October 11, 1957; revised September 28, 1959.

There is no closed formula giving the characters of an unspecialized class $\kappa = (k_1, k_2, \cdots, k_n)$, but, with the aid of Murnaghan's recursion formula

$$\chi_\kappa^\rho = \sum_{(i)} \epsilon_i \chi_\kappa^{\rho^{(i)}} \tag{3}$$

they may be expressed by means of characters of a symmetric group of lower degree. In (3), the partition κ' appears, if one of the components of κ, say k_a, is deleted. The partitions $\rho^{(i)}$ are obtained from ρ following certain rules, also telling us whether the sign factor ϵ_i has the value $+1$ or -1 or 0.

The method of hook removals, introduced by Nakayama [5], simplifies the application of (3) in two respects. Firstly, by this method only those $\rho^{(i)}$ are calculated which correspond to an $\epsilon_i \neq 0$, and, secondly, the value of ϵ_i is obtained as

$$\epsilon_i = (-1)^{z_i} \tag{4}$$

where z_i is the so-called leg-length of a hook.

It is advantageous to use, for the representation partitions ρ and $\rho^{(i)}$ in (3), the notation introduced as Second Method in [3]. From now on, we shall give to this notation the name of *rim* of the partition, because of its geometrical meaning, as in a paper by Frame and others [6]. The rim R of a partition $\rho = (r_1, r_2, \cdots, r_m)$ can be defined as a sequence of binary digits, in which each component r_i is represented by a digit $= 0$, immediately followed by $r_i - r_{i+1}$ digits $= 1$, considering r_{m+1} to be $= 0$. For example, the rim of $\rho = (6.6.5.5.3.2) = (6^2.5^2.3.2)$ is

$$R = 001001101011. \tag{5}$$

Conversely, the partition ρ is easily reconstructed from R, the component r_i being equal to the total number of 1-valued digits to the right of the ith zero. Evidently, 1-valued digits to the left, and zeros to the right, do not influence this determination of r_i, nor do they affect the definition of R. Therefore, a sequence of digits like

$$11111001001101011000000$$

is considered to be a rim of the same partition ρ as (5). We will distinguish (5) as the normalized form of the rim. In a computer as BESK, where the basic operations, assuming fixed binary point, act on numbers with absolute value $\leqq 1$, the rims are conveniently inserted immediately after the binary point. Thus, the rim is represented by a number

$$R = 1 - (1 + \sum_{i=1}^{m} 2^{R_i}) \cdot 2^{-R_0} \tag{6}$$

where R_i has the same meaning as in (2) and R_0 is an arbitrarily chosen integer (within the scope given by the word-length) satisfying $R_0 > R_1$. The normalized form of R corresponds to $R_0 = R_1 + 1$.

The geometrical illustration of R is obtained, if the digits are represented by adjacent line segments of equal lengths, those labeled 0 being directed vertically downwards, those labeled 1 horizontally to the left. The broken line constructed in this way is precisely the right, lower rim of the Young diagram of the partition ρ. Historically, the notation R was first based on this geometrical figure.

Each pair of digits, formed by a 0 in a rim and a 1 situated k steps in the rim

to the right of this 0, is said to form a *hook* of length k in the rim or, shorter, a k-hook. It corresponds to the Nakayama hook, and, together with the intermediate digits, it is a binary representation of the rim-hook used by Frame and others [6]. The two digits which form the first mentioned pair will be called the end digits of the hook. The number of zeros *between* the end digits is $=z_i$ in (4). The removal of a hook is effectuated by simply interchanging the two end digits of the hook.

In terms of the expression (6), these considerations will read as follows. The jth 0 of R, if counted from the binary point, may be used as left end digit of a k-hook, if and only if $R_j - k$ is different from all the quantities R_i . If this condition is fulfilled, the hook will be removed by substituting $R_j - k$ for R_j in that term of (6) where $i = j$. Thus, the value of R will change to R_+ where

$$R_+ = R + (1 - 2^{-k}) \cdot 2^{R_j - R_0}. \tag{7}$$

The number z_j of zeros between the end digits is equal to the number of indices i for which $R_j > R_i > R_j - k$.

Another operation used in the calculation process is the passage to the rim $\tilde{R}$ of the partition $\tilde{\rho}$, conjugate to ρ. This rim $\tilde{R}$ is obtained by writing the digits of R in the reversed order, replacing each 1 by 0 and each 0 by 1. Thus, corresponding to the example (5) we get

$$\tilde{R} = 001010011011$$

being the rim of $\tilde{\rho} = (6^2 \cdot 5 \cdot 4^2 \cdot 2)$.

It should be noted that the essential part of our notation is the sum

$$\sum_{i=1}^{m} 2^{R_i} = r \tag{8}$$

which enters in the expression (6). The special form (6) was chosen in order to adapt it to the operation list of BESK. It is clear that the form may be modified so as to suit other computers.

3. The Actual Calculation Program. In order to avoid complication, I have restricted the calculation to those representations ρ which can be stored, by means of their rims, within one word in BESK. This means the following restriction for technical reasons,

$$r_1 + m \leqq 38, \tag{9}$$

r_1 being the greatest component of ρ, as before. This condition indicates the limit of the capacity of the program in its present form.

If ρ varies through *all* partitions of N, we have $r_1 + m \leqq N + 1$. Hence, N must in such cases be limited to

$$N \leqq 37. \tag{10}$$

For an isolated ρ, satisfying (9), N may be greater. For instance, if $\rho = (19^{19})$, we get $N = 19^2 = 361$, the greatest value of N that can appear in the present version of the program.

The partition κ is stored in a different way (see below), not using the rim, but requiring that the components, k_a , of κ satisfy the condition

$$k_a \leqq 37 \quad \text{for all} \quad a = 1, 2, \cdots, n. \tag{11}$$

In fact, if a component of κ is $\geqq r_1 + m$, the value of χ_κ^ρ is 0, and there is no need for calculation. Thus, the condition (11) means no further restriction, from a practical point of view.

Originally, the program was not intended to calculate complete character tables, but only to give individual characters, thus serving as a substitute for a character table. A set of compiling routines, activated by different start operations, has later been added, permitting the program to work in the following variants:

(i) Calculate all χ_κ^ρ for a given N, under the heading of one ρ at a time, either for strictly all partitions ρ or for only one in each pair of ρ and $\tilde{\rho}$. Stop after the completion of an N, and then, after start, continue with the next, $N + 1$.

(ii) As in (i), but under the heading of one κ, instead of ρ, at a time. In the case when χ_κ^ρ is not calculated because $\chi_\kappa^{\tilde{\rho}}$ has already been, the letter C is printed (for "conjugate").

(iii) Calculate all χ_κ^ρ for one given ρ.

(iv) Calculate all χ_κ^ρ for one given κ.

(v) Calculate χ_κ^ρ for one given pair of ρ and κ.

(vi) As in (iv) but, instead of giving out the values of χ_κ^ρ, select the one having the greatest absolute value. Print this χ_κ^ρ together with the corresponding ρ and κ.

The central, common part of these programs will now be outlined, and a few points of it will be treated in greater detail. Suppose that ρ and κ are already represented in the computer, ρ by means of its rim R, inserted, in its normalized form, in a storage cell, while κ is represented in the following way. Let us write

$$\kappa = (k_1, k_2, \cdots, k_n, 1^q) \tag{12}$$

where

$$k_1 + k_2 + \cdots + k_n + q = N.$$

The order of succession of the components k_a ($a = 1, 2, \cdots, n$) must be fixed before the calculation, not necessarily a monotonous order although it is appropriate to begin with the greatest components. There may be some $k_a = 1$, but in most cases all the 1-valued components are comprised in 1^q. Now, each of the components k_a will be stored as represented by a number 2^{-40+k_a}. For the storage of these numbers a set of cells are at disposal, which we will call H_a. The number q is separately stored.

The calculation method consists in a repeated use of the recurrence formula (3), removing successively hooks of length $k_1, k_2, \cdots, k_n$ by the rule (7). After this, it remains to make calculations using (1) with q for N. The rims of the representation partitions appearing after the hook removals will be stored in a set of cells, called W_a ($a = 1, 2, \cdots, n$), where W_a will contain a rim from which a k_a-hook is the next to be removed. Its sign position will contain the sign factor resulting from the preceding hook-removals.

Designating by χ a quantity intended to get the end-value χ_κ^ρ, by P and P_+ two partition rims under consideration, and by e and e_+ certain corresponding sign factors, we describe the calculation:

1. Insert initial value 0 for χ.

   If $n = 0$, *i.e.* if $\kappa = (1^N)$, put R for P_+ and pass directly to point 6 below.

   If $n > 0$, insert R into W_1, put 1 for a, and arrange such an order modification that point 3 will not be neglected. Pass to point 2.

2. The contents of W_a are examined: The sign digit of this word replaces e, the rest of it replaces P. Pass to point 3 or 4 according to latest order modification.
3. The positions j of the left end-digits of all possible k_a-hooks in P are determined. Digits = 1, placed in the found positions, are called *hook indicators* and are inserted into H_a (together with the number 2^{-40+k_a} already stored there). Pass to point 4.
4. The contents of H_a are examined:
   If there are no hook indicators, change a to $a - 1$.
   If the new $a > 0$: After modification so as to neglect point 3, pass to point 2.
   If the new $a = 0$: Calculation ended. Pass to output routine.
   If there are one or more hook indicators: The most significant of them is removed from H_a. Corresponding to the same indicator, the k_a-hook is removed from P, giving P_+. The number, z, of zeros within this hook is counted, and $e \cdot (-1)^z$ gives e_+.
   If $a < n$: Pass to point 5.
   If $a = n$: Pass to point 6.
5. Insert e_+ into the sign position and P_+ into the other positions of W_{a+1}. Change a to $a + 1$. After order modification, assuring that point 3 will not be neglected, pass to point 2.
6. Calculate $\chi^{\pi}_{(1^p)}$ using formula (1) for the partition π corresponding to the rim $P_+ . -NB$. For all p such that $3 \leqq p \leqq M$ $(= 13)$, these characters are precalculated for all π and permanently stored.
   Replace χ by $\chi + e_+ \cdot \chi^{\pi}_{(1^p)}$. Pass to point 4.

4. Some Details of the Program. It should be mentioned that great values of p may cause $\chi^{\pi}_{(1^p)}$ and χ to exceed the word-length. Multiple-word routines are then automatically called into action, permitting, in the actual lay-out of the program, character values of at most 576 decimal digits.

Further, if $p > M' \geqq M$ $(M' = 19)$, the calculation of $\chi^{\pi}_{(1^p)}$ is not executed in point 6, but the rim P_+ is stored and then the program passes to point 4. For the storage of different P_+ and for accumulating coefficients of them, a set of working storage cells is available. When it occurs that this set is fully occupied, and always before passing to the output routine, the calculations of point 6 are executed for all the stored rims, and the value of χ is correspondingly determined. After clearing the set of working cells, the program may continue, if necessary.

Some of the points which have been programmed in a specific way will be described in the sequel.

a. Determination of the indicators (*point* 3). By shifting the rim P to the left k_a steps (and dropping the digits passing the binary point), a number Q is obtained. By changing all zeros of P into 1, and all ones into 0, one form of complement, $\bar{P}$, is obtained. The "logical product" of Q and $\bar{P}$ gets a 1 in those positions (only) where both Q and $\bar{P}$ have a 1. Thus, this logical product gives directly all the indicators of k_a-hooks in P, and it is inserted into H_a.

b. Test on hook indicators in H_a (*point* 4). The negative contents (modulo 2) of H_a are normalized. This means that a number, represented by binary digits, is shifted to the left, until unequal digits appear on either side of the binary point. In the actual case, the normalization will give a 1 to the left, a 0 to the right of the

binary point. If there were no indicators in H_a, there would be only zeros to the right of the binary point after the mentioned normalization. This case is very easily distinguished from the opposite one.

c. Removal of an indicator and a hook (*point* 4). The normalization process includes counting the shift steps. Thus, the number, v, of shift steps during the normalization under point *b* above determines the position of the left end-digit of the first k_a-hook, this position being the $(v + 1)$st after the binary point. The number $h = 2^{-v-1}$, obtained from 2^{-1} by shifting v steps to the right, is subtracted from the contents of H_a, thus removing the first indicator.

Further, if the negative number $-h$ is shifted k_a steps to the right and then h is added, we obtain

$$h' = h - h \cdot 2^{-k_a} = 2^{-v-1} - 2^{-k_a-v-1}.$$

According to (7), the rim, appearing when the hook in question is removed from P, is obtained as $P_+ = P + h'$.

d. The number of zeros within the hook (*point* 4). The logical product of the numbers h' and P consists of those digits of P which appertain to the hook, and, outside the hook, zeros. By means of a $v + 1$ steps shift to the left, it is brought to such a position that the interior of the hook begins at the binary point. In the present version of the program, the number, z, of zeros within the hook is now obtained by repeated normalizations, accumulating only the number of those shift steps when zeros pass the binary point. It would have been better to examine the hook one digit at a time, because such a process is more rapid for hooks of short length. Removals of short hooks have to be repeated considerably more times than those of longer hooks.

Point 4 is the most frequently repeated sequence of the program in the phase of applying the recursion formula (3). The programming methods just described have resulted from my efforts to reduce the execution time for this sequence as far as possible. The weakest point is still the determination of z.

With respect to the definition (12), removal of hooks of length = 1 need not occur in a standard running of the program, unless such hooks are used for special purposes, e.g. checking. Instead of the removal of 1-hooks, the Frobenius formula (1) is used which, fortunately, works considerably faster.

e. The calculation of $\chi^{\pi}_{(1^p)}$ (*point* 6). Only a few remarks on the programming of formula (1) are to be made. The calculated characters, which are all integers, may have very large values requiring arrangements for multiple-word calculation. Further, the numerator, being still larger, should not be explicitly evaluated, and, finally, operations which might introduce round-off errors should be avoided. The following strategy will meet these demands. Each factor entering in (1) is factorized into a product of prime powers. In a set of storage cells, one cell for each prime number, the exponents of the prime powers are accumulated, i.e. added if they come from the numerator, subtracted if they come from the denominator. Afterwards, the character is obtained by successive multiplications of the prime numbers, with due regard to the final values of the exponents, which will never be negative. When multiple word-length is required, the number of storage cells is successively adapted to the actual demand. As an intermediate stage, as many of the prime factors as possible without exceeding one word-length are multiplied, thus reducing the required number of multiple-word operations.

5. Observations from the Computing Work. The version (i) of the program (see §3) has been run by the regular staff of BESK-operators, when computer time has been available. Complete character tables have been calculated for all $N \leqq 20$. The results were produced by the machine punched on paper tapes, and these have also been printed out on paper sheets [8]. In each pair of conjugate representations, ρ and $\tilde{\rho}$, only one has been treated. The chosen representation partition has been written out as a common heading for the characters in this representation. These have then been listed following the class partitions, κ, which succeed each other in lexicographical, non-increasing order, beginning with $\kappa = (N)$ and ending with $\kappa = (1^N)$. The calculation time, including the tape punching of the results, was for the table $N = 16$ about 105 minutes, for the table $N = 20$ about 51 hours, of course scattered on many partial runnings.

The versions (ii) through (v) have not been systematically employed for table calculations. Yet, many series of experiments have been made, giving some information on the differences in complexity between the calculation schemes in different cases, *e.g.* for different classes. The results can not be reported here.

The variant (vi) was prepared in view of giving an experimental answer to the question proposed in Note 9 of the paper [1] by Bivins and others. The case of $\kappa = (1^N)$ has been treated for all $N \leqq 30$, and we will list here those representation partitions which were found to give the greatest character for each of the $N = 3, 4, \cdots, 30$: (2.1), (3.1), (3.1^2), $(3.2.1)$, $(4.2.1)$, $(4.2.1^2)$, $(4.2^2.1)$, $(4.3.2.1)$, $(5.3.2.1)$, $(5.3.2.1^2)$, $(5.3.2^2.1)$, $(6.4.2.1^2)$, $(5.4.3.2.1)$, $(6.4.3.2.1)$, $(6.4.3.2.1^2)$, $(7.4.3.2.1^2)$, $(7.5.3.2.1^2)$, $(7.5.3.2^2.1)$, $(7.5.3.2^2.1^2)$, $(7.5.4.3.2.1)$, $(7.5.4.3.2.1^2)$, $(8.5.4.3.2.1^2)$, $(8.6.4.3.2.1^2)$, $(8.5.4.3.2^2.1^2)$, $(8.6.4.3.2^2.1^2)$, $(8.6.4.3^2.2.1^2)$, $(8.6.5.4.3.2.1)$, $(8.6.5.4.3.2.1^2)$.

Giant character values may appear already at rather small degrees N. As an example, for $N = 30$, the maximum character is a number with 16 decimal digits. The greatest value calculated by the version (v), under the condition (9), is expressed by 325 decimal digits; it corresponds to $N = 361$, $\rho = (19^{19})$, $\kappa = (1^{361})$. Another character, corresponding to the same N and ρ, but $\kappa = (7.5.1^{349})$, is a 313-digit, negative number, *etc.*

There is no particular checking sequence in the program. The reason is that the main purpose of the program is to calculate individual characters, as in version (v), and then the normal checking method, based on orthogonality, is of no use. Therefore, in order to check a calculated value, χ_κ^ρ, one should recalculate it in introducing the components $k_1, k_2, \cdots$ of κ in an *altered order*, this leading to a check if they are not all equal. If they are all $= 1$, *i.e.* $\kappa = (1^N)$, the recalculation is made with one or more $k_a = 1$ in (12), where q is adequately diminished. Thus, the only unchecked cases are those of $\kappa = (p^q)$ with $p > 1$. Concerning these cases, see, however, §7.

Most of the tables calculated by version (i) being unchecked, it is still possible to subject their values, as they are punched on paper tape, to the orthogonality test by means of a separate program. However, it will be neither possible nor necessary to carry out this test to its full extent.

6. Deductions for Further Improvements. Although it is interesting that, theoretically, the entire character calculation could be built up by hook removals.

it is important that, in practice, more effective routines can be introduced, reducing the number of hook removals. For such a purpose, the Frobenius formula (1) was included in the program described above, where it could replace the removals of all hooks of length = 1. In an improved program, this formula is to be generalized to the case of $\kappa = (p^q)$ where p may be > 1.

Another way of reducing the number of hook removals to be executed is to observe that one and the same rim may be obtained several times when sequences of hooks of length $k_1, k_2, \cdots, k_{n'}$ $(2 \leqq n' < n)$ are removed. Instead of calculating the contribution to χ of that rim each time it occurs, one should multiply the contribution by an appropriate coefficient. Already in the described version of the program, this procedure was used at a certain point, as indicated in the beginning of §4 ($\cdots$ if $p > M' \geqq M \cdots$). It should, however, be used more systematically.

The realization of these desiderata requires some new formulas and rules. These can be based on the rim notation. In [4], detailed foundations and deductions are given, and, in the present paragraph, we shall quote from [4] what is needed for the developments in §7 and §8.

The essential part of the rim notation, *i.e.* the sum (8):

$$r = \sum_{i=1}^{m} 2^{R_i} \tag{8}$$

will now be called, as in [4], the *binary model* of the partition ρ. If (8) is written down in binary digits, to each 1-valued digit is associated a quantity called its *weight*, defined as the total number of zeros to the right of it up to the binary point. The weight of r is defined as the sum of the weights of its 1-valued digits. Calling it $N(r)$ we get, with regard to (2),

$$N(r) = r_1 + r_2 + \cdots + r_m . \tag{13}$$

Thus, the weight of r is equal to the number N of which ρ is a partition. A special, binary model called u, with $R_i = m - i$, has the weight

$$N(u) = 0; \qquad u = \sum_{i=1}^{m} 2^{m-i}. \tag{14}$$

The digits, 1 and 0, of a binary model r could be considered as indicating the presence or absence, respectively, of m movable objects at the locations represented by the digit positions of the number r.

The removal of a k-hook from the rim R, as in (7), is equivalent to an operation called *reduction* of the binary model r to another one, r', such that

$$r' = r + 2^{R_j - k} - 2^{R_j}. \tag{15}$$

It is easily proved (see [4] for this and other details) that

$$N(r') = N(r) - k,$$

i.e., k is the decrease in weight. We say that (15) is a reduction of r by the amount k. This reduction could be described by an operation on the jth object of r consisting in moving it k steps to the right.

A sequence of successive reductions by the amounts $k_1, k_2, \cdots, k_n$ is called a *chain* of reductions. If $k_1 + k_2 + \cdots + k_n = N$, the chain will transform r to a

binary model u of the form (14). The number of existing chains of reductions by the indicated amounts, leading from a given r to u, is not independent of the order of succession of the reduction amounts. Therefore, we have introduced the *value* of a chain as $(-1)^{z_1+z_2+\cdots+z_n}$, where z_j is the number of objects (1-valued digits) passed by the moved object during the reduction by the amount k_j. As shown in [4], the *sum of the values* of all chains of reductions by the amounts $k_1, k_2, \cdots, k_n$ is the same as the sum obtained, if another order of succession is chosen for these amounts. It follows that this sum is characterized by the *partition* κ of N having the components $k_1, k_2, \cdots, k_n$. It will be denoted by $\chi_\kappa(r)$ and sometimes called the characteristic value of κ determined by r. It is, in fact, identical with the character χ_κ^ρ, but it may be convenient to keep the notation $\chi_\kappa(r)$.

A generalization is obtained in considering those chains which reduce r, not to u, but to r', where r' is the binary model of a partition ρ' of a number $N(r') < N(r)$. The sum of the values of all chains of reductions leading from r to r' and consisting of reductions by the amounts indicated by the components of a certain partition κ'' of the number $N(r) - N(r')$, will be denoted by $\chi_{\kappa''}(r; r')$ and called the characteristic value of κ'' determined by the reduction of r to r'. If κ' is a partition of the number N', we get the fundamental formula

$$\chi_{\kappa''\kappa'}(r) = \sum_{r'} \chi_{\kappa''}(r; r') \cdot \chi_{\kappa'}(r') \tag{16}$$

where the binary models r' correspond to partitions ρ' of the number N'. The demonstration depends mainly on the consideration of the chains leading from r to r' and of those leading from r' to u. Concerning the details, the reader is referred to [4]. It should be observed that (3) is contained in (16) as a special case.

We shall show, in §7, how (16) may simplify the calculation of χ_κ^ρ if special properties of the components of κ can be utilized, and, in §8, how (16) may lead to relations independent of κ, but depending on different ρ.

7. Utilization of Properties of κ. If κ is a partition of N, let κ'' be the partition formed by those components of κ which have a common factor p, where p is an integer > 1. Then, the components of κ'' may be written $pq_1, pq_2, \cdots, pq_h$. This κ'' is a partition of a number, say pM, also containing the factor p. The numbers $q_1, q_2, \cdots, q_h$ are the components of a partition of M, which we will call ξ. Finally, we put

$$N - pM = N'$$

being, in (16), the weight of r'. It is clear that κ' is a partition of this N', as it is formed by all those components of κ which are not comprised in κ''.

At the determination of $\chi_{\kappa''}(r; r')$, the objects of r are moved to the right by multiples of p steps. Therefore, the binary model r may be split into p binary models $r^{(1)}, r^{(2)}, \cdots, r^{(p)}$ in such a way that each object remains within one and the same model when moved. In order to get suitable and unambiguous rules, we shall assume that both the number of 1-valued digits of r and the total number of digits of r are divisible by p. In fact, we can always adjoin ones to the right of r and zeros to the left, because this does not change the weight of r. Now, the splitting of r is most easily accomplished in writing the digits of r in successive, vertical columns of

height p, as in a matrix. The digits found in horizontal rows then form the binary models $r^{(j)}$, $j = 1, 2, \cdots, p$.

Each chain of reductions corresponding to κ'', executed on r, is represented by chains of reductions by the amounts $q_1, q_2, \cdots, q_h$ executed on the binary models $r^{(j)}$. Thereby, the reduction amounts on $r^{(j)}$ will be denoted $q_1^{(j)}, q_2^{(j)}, \cdots, q_{h_j}^{(j)}$, which are the components of a partition $\xi^{(j)}$ of a number $M^{(j)}$. It is necessary that

$$M^{(j)} \leqq N(r^{(j)}). \tag{17}$$

Because $M^{(1)} + M^{(2)} + \cdots + M^{(p)} = M$, it follows that necessarily

$$\sum_{j=1}^{p} N(r^{(j)}) \geqq M, \tag{18}$$

as soon as there are chains corresponding to κ''.

On the contrary, if

$$\sum_{j=1}^{p} N(r^{(j)}) < M, \tag{19}$$

there can not be any chains corresponding to κ'', and, hence, the sum in (16) is empty. Consequently, if (19) holds, the character $\chi_{\kappa''\kappa'}(r) = 0$.

Returning to the case when (18) is satisfied, we have to distribute the components of ξ, in all possible ways, to partitions $\xi^{(j)}$ of numbers $M^{(j)}$ satisfying (17). For each of these distributions we subject the binary models $r^{(j)}$ to chains of reductions by the amounts given by the appropriate $\xi^{(j)}$. We denote the resulting binary models by $r'^{(j)}$. One particular system of such binary models is obtained by means of chains of reductions which correspond to certain chains of reductions executed on r. The binary model r', obtained by this reduction of r, can be written down very easily, if the digits are read column by column in the matrix where the $r'^{(j)}$ are rows.

Further we may conclude, similarly to the discussion in [4] (p. 105), that the value of the chain from r to r' differs from the product of the values of the chains from $r^{(j)}$ to $r'^{(j)}$ only by a factor $=$ sgn $Q \cdot$ sgn Q', where sgn Q is $+1$ or -1 depending on r and the $r^{(j)}$ only, whereas sgn Q' is $+1$ or -1 depending on r' and the $r'^{(j)}$ only. For the sums of these values we get

$$\chi_{\kappa''}(r; r') = \text{sgn } Q \cdot \text{sgn } Q' \sum \prod_{j=1}^{p} \chi_{\xi^{(j)}}(r^{(j)}; r'^{(j)}) \tag{20}$$

where the summation is extended over all distributions of the components of ξ to such $\xi^{(j)}$ which lead to the actual set of $r'^{(j)}$. If some of the components of ξ are equal, it may happen that one set of $\xi^{(j)}$ is obtained several times. Let a component, valued i, be present τ_i times in ξ and τ_{ji} times in $\xi^{(j)}$. This set of $\xi^{(j)}$ is obtained so many times as given by the expression

$$\varphi(\xi^{(1)}, \xi^{(2)}, \cdots, \xi^{(p)}) = \prod_{i=1}^{M} \frac{\tau_i !}{\prod_{j=1}^{p} \tau_{ji} !} \tag{21}$$

We can now write (20) in the following form:

$$\chi_{\kappa''}(r; r') = \text{sgn } Q \cdot \text{sgn } Q' \cdot \sum \varphi(\xi^{(1)}, \cdots, \xi^{(p)}) \prod_{j=1}^{p} \chi_{\xi^{(j)}}(r^{(j)}; r'^{(j)}) \tag{22}$$

where the summation is extended over all *different* distributions of ξ to $\xi^{(j)}$. When inserting (22), with (21), into (16), we observe that the factor sgn Q, but not the factor sgn Q', may be put outside the summation sign (over r').

The sign factor sgn Q is determined by means of the number ω of inversions of the objects of r introduced by replacing r by the set $r^{(j)}$. A method to obtain ω, suitable for a computer program, could be sketched as follows: The method consists in counting over the objects column by column in the matrix formed by $r^{(j)}$. Let, at a certain moment, $m^{(j)}$ be the number of still uncounted objects in the jth row. If, then, the next object is encountered in the Jth row, the following two operations are made:

(i) The accumulated value of ω is augmented by $m^{(1)} + m^{(2)} + \cdots + m^{(J-1)}$.

(ii) $m^{(J)}$ is replaced by $m^{(J)} - 1$.

When all objects have been counted, we get

$$\operatorname{sgn} Q = (-1)^{\omega}. \tag{23}$$

Of course, sgn Q' is obtained similarly from $r'^{(j)}$.

Special cases occur if (18) reads as an equality. In these cases, the relations (17) must be equalities, too, and it follows that all the binary models $r'^{(j)}$ have the weight $= 0$. Therefore, there is only one set $r'^{(j)}$ and one single r'. This r' is the binary model of the partition called by Littlewood [7] the p-residue, and by Nakayama [5] the p-core of ρ. Observing that, in these cases, the characteristic values $\chi_{\xi^{(j)}}(r^{(j)}; r'^{(j)})$ become veritable characters, the formula (16), with regard to (22) and (21), gives the following expression:

$$\chi_{\kappa''\kappa'}(r) = \operatorname{sgn} Q \cdot \operatorname{sgn} Q' \cdot \chi_{\kappa'}(r') \cdot \sum \varphi \prod_{j=1}^{p} \chi_{\xi^{(j)}}(r^{(j)}). \tag{24}$$

This will be still more simplified if *all* components of κ have the common factor p. In this case κ'' may be chosen $= \kappa$ so that κ' will vanish. If all the binary models $r^{(j)}$ contain equal numbers of 1-valued digits, the character $\chi_\kappa(r)$ is given by (24), where $\chi_{\kappa'}(r')$ has to be put $= 1$. If the $r^{(j)}$ do not satisfy this condition (although the number of 1-valued digits in r is divisible by p), the inequality (19) holds (see [4]) and, consequently, the character $\chi_\kappa(r)$ is $= 0$.

Finally, if $\kappa = (p^M)$, the last considerations may be completed by giving the explicit expressions for $\chi_{\xi^{(j)}}(r^{(j)})$. Because $\xi^{(j)} = (1^{M^{(j)}})$, the Frobenius formula (1) is applicable. Writing, in analogy to (2),

$$R_h^{(j)} = r_h^{(j)} + m - h \qquad (h = 1, 2, \cdots, m)'$$

where m is the number of 1-valued digits in each $r^{(j)}$, we get for the quantities in (21):

$$\tau_1 = M; \qquad \tau_{j1} = M^{(j)}; \qquad \tau_i = \tau_{ji} = 0 \qquad \text{for } i \neq 1.$$

Observing that there is only one set of $\xi^{(j)}$, we obtain from (24) the non-zero characters:

$$\chi_{(p^M)}(r) = \operatorname{sgn} Q \cdot \operatorname{sgn} Q' \cdot M! \prod_{j=1}^{p} \left(\frac{\prod_{h<i} (R_h^{(j)} - R_i^{(j)})}{R_1^{(j)}! \cdots R_m^{(j)}!} \right) \tag{25}$$

In summarizing, we find that the utility of the described method may appear in two respects. Firstly, the binary model r is split into the models $r^{(i)}$ which, corresponding to partitions of smaller numbers than the original one, will need considerably less computing work. Secondly, the calculation of the factor φ by means of (21) will shorten the computing work as soon as it replaces the treatment of several equal terms in a sum. The utility of the inequality (19) for the detection of zero-valued characters is evident.

8. Relations Independent of κ. Turning our attention to the binary models r and r' in $\chi_{\kappa''}(r; r')$ of (16), we write, as in (8),

$$r = \sum_{i=1}^{m} 2^{R_i} \quad \text{and} \quad r' = \sum_{i=1}^{m} 2^{R_i'} \tag{26}$$

where, as in (2),

$$R_i = r_i + m - i \quad \text{and} \quad R_i' = r_i' + m - i, \tag{27}$$

the number of 1-valued digits being the same, m, in both models.

Preliminarily, let us assume that r and r' satisfy the conditions:

$$\begin{cases} r_i \geqq r_i' & \text{for } i = 1, 2, \cdots, m, \\ r_i' \geqq r_{i+1} & \text{for } i = 1, 2, \cdots, m-1. \end{cases} \tag{28}$$

By (27), this means

$$R_i \geqq R_i' \quad \text{and} \quad R_i' > R_{i+1}, \text{ respectively.}$$

It follows that, by each chain of reductions leading from r to r', the ith object of r will become the ith object of r', *i.e.* the objects will not be permuted. The values of such chains, corresponding to the components of κ (written instead of κ''), will be $+1$, and the sum, $\chi_\kappa(r; r')$, of these values will be equal to the number of the chains.

If the components of κ are $k_1, k_2, \cdots, k_n$, with

$$k_1 + k_2 + \cdots + k_n = N,$$

each chain is constructed by distributing these components to sequences $k_1^{(i)}, k_2^{(i)}, \cdots, k_{n_i}^{(i)}$ $(i = 1, 2, \cdots, m)$, the ith sequence indicating the movements of the ith object, thus

$$k_1^{(i)} + k_2^{(i)} + \cdots + k_{n_i}^{(i)} = R_i - R_i' = r_i - r_i'. \tag{29}$$

The chains depend merely on κ and on the differences $r_i - r_i' = d_i$, but not on the components r_i and r_i' themselves, provided that they satisfy (28). The number of chains is equal to the number of distributions of κ satisfying (29). If another pair of binary models r and r' satisfies (28) and gives the same set of differences d_i, which may, however, come in a different order, the system (29) will still have the same number of solutions. Therefore, denoting by δ the partition of N with the components $d_1, d_2, \cdots, d_m$, we find that the number of chains leading from r to r' depends merely on κ and δ. We will write this number ψ_κ^δ, and for binary models r and r' satisfying (28) we have

$$\chi_\kappa(r; r') = \psi_\kappa^\delta$$

In the general case, the condition (28) does not hold but only

$$r_i \geqq r_i' \quad \text{for} \quad i = 1, 2, \cdots, m, \tag{30}$$

expressing that the objects never move to the left during a reduction. The objects may be permuted by a reduction chain. Let us consider those chains by which the ith object of r becomes the p_ith object of r' ($i = 1, 2, \cdots, m$). Here, the numbers $p_1, p_2, \cdots, p_m$ form a permutation P of the integers $1, 2, \cdots, m$. Not all permutations P are possible, because it is required that

$$R_i - R'_{p_i} \geqq 0 \quad \text{for all} \quad i = 1, 2, \cdots, m. \tag{31}$$

We write

$$R_i - R'_{p_i} = d_i,$$

which are, as before, the components of a partition δ, now depending on P. The number of the chains considered is $\leqq \psi_\kappa^\delta$. It will be $< \psi_\kappa^\delta$ as soon as a distribution like (29) would cause two objects to coincide after some reduction steps, which is not permitted. By a reasoning, made in detail in [4] (see p. 99 ff.), we find that, if sgn P is defined as $+1$ for all even and -1 for all odd permutations, the sum of all sgn $P \cdot \psi_\kappa^\delta$ will be equal to the sum of all values of the chains leading from r to r'. Thus,

$$\chi_\kappa(r; r') = \sum_{(P)} \operatorname{sgn} P \cdot \psi_\kappa^\delta \tag{32}$$

where P ranges over all permutations of $1, 2, \cdots, m$ compatible with (31). This formula holds for all partitions κ of the number N, so that we may leave out the index κ. By $\chi(r; r')$ and ψ^δ we could mean vectors or 1-column matrices, simplifying (32) to

$$\chi(r; r') = \sum_{(P)} \operatorname{sgn} P \cdot \psi^\delta \tag{33}$$

We note that, for $r' = u$, the quantity $\chi_\kappa(r; r')$ means the character χ_κ^ρ. Thus, the vector χ^ρ is, by means of (33), expressed as a linear combination of the vectors ψ^δ, both ρ and δ being partitions of N. Now, as the different vectors χ^ρ are linearly independent, and as the number of the vectors ψ^δ is equal to that of χ^ρ, we conclude that the vectors ψ^δ also form a linearly independent system.

At the practical application of these formulas we have to calculate the quantities ψ_κ^δ. This could be done either by (29) or by counting chains of reductions from an r to an r', arbitrarily chosen but in accordance with (28). It is essential that no determinations of the sign $+$ or $-$ are required at this stage of the process. This means that the most time-consuming point in the calculation program (see §4, d) is avoided. At the insertion in (33), the factor sgn P is determined independently of the κ.

In certain situations one may prefer expressing the ψ^δ in terms of χ^ρ. This is done by solving the system (33), with $r' = u$, with respect to ψ^δ. The obtained expressions also present a theoretical interest. In fact, ψ_κ^δ is a compound character for the class κ of the symmetric group in a representation obtained in the following way. Let S be the group of all permutations of a set of N objects. Dividing this set of objects in subsets containing $d_1, d_2, \cdots, d_m$ objects, according to the partition

δ, we consider the subgroup S^δ formed by those permutations which only permute the objects within the subsets. The group S is divided in complexes, $S_1, S_2, \cdots, S_h$, induced by the subgroup S^δ. For each element $s \in S$ we have

$$s \cdot S_i = S_{i'}$$

Thus, $S_1, \cdots, S_h$ form a basis for a representation of S by matrices. The trace of the matrix representing s can be shown to be $= \psi_\kappa{}^\delta$ where κ is the class to which s belongs. The demonstration must be omitted here. The above mentioned expression of ψ^δ in terms of χ^δ indicates directly the structure of the described representation.

Returning our attention to the computing work required by (16), we see how the formulas (32) and (33) permit the calculation to be performed in "blocks". This will be the more necessary, the more the degree of the symmetric group increases. An improved computer program for character calculation should be planned so as to combine in the most efficient way the methods described in §7 and §8. This program has not yet been constructed, and, in fact, it is necessary first to establish the purpose for which the program is to be used, because this has a great influence on the decision concerning efficiency.

Drömstigen 13
Bromma (Sweden)

1. R. L. Bivins, N. Metropolis, P. R. Stein & M. B. Wells, "Characters of the symmetric groups of degree 15 and 16," *MTAC*, v. 8, 1954, p. 212.
2. S. Comét, "On the machine calculation of characters of the symmetric group," *Comptes rendus*, 12th Congrès Math. Scand., Lund, 1953, p. 18.
3. S. Comét, "Notations for partitions," *MTAC*, v. 9, 1955, p. 143.
4. S. Comét, "Über die Anwendung von Binärmodellen in der Theorie der Charaktere der symmetrischen Gruppen," *Numerische Math.*, v. 1, 1959, p. 90.
5. T. Nakayama, "On some modular properties of irreducible representations of a symmetric group," *Jap. Jn. Math.*, v. 17, 1941, p. 165, 411.
6. J. S. Frame, G. de B. Robinson & R. M. Thrall, "The hook graphs of the symmetric group," *Canadian Jn. Math.*, v. 6, 1954, p. 316.
→ D. E. Littlewood, "Modular representations of symmetric groups," Roy. Soc., London, *Proc.*, ser. A, v. 209, 1951, p. 333.
8. Character tables for $N \leqq 20$, punched on paper tapes and printed on paper, kept in the library of Matematikmaskinnämnden, P.O. Box 6131, Stockholm 6, Sweden, telex 1613 ("besk sth").