An algorithm for Weyl modules irreducibility

M.Barnabei (1), V.Frontini (2), F.Sgallari (1)

(1) Dipartimento di Matematica - Università di Bologna
Piazza di Porta S.Donato 5 - 40127 Bologna - Italy
(2) CINECA Computing Centre
via Magnanelli 15 - 40033 Casalecchio di Reno (Bologna) - Italy

1. Introduction

In the classical representation theory of general linear groups over fields of characteristic zero two classes of modules play a fundamental role, namely, Schur modules and Weyl modules relative to a given Young shape [13-15]. As well known, these are irreducible modules, and, for every Young shape λ, the Schur module relative to λ is isomorphic to the Weyl module relative to the conjugate shape $\tilde{\lambda}$. Recently, it has been recognized that the definitions of Schur and Weyl modules can be adapted in order to make sense over fields of arbitrary characteristic (see, e.g., [1-5,8,9]), giving rise to two classes of modules which are indecomposable but, in general, neither irreducible, nor isomorphic. Hence, the problem arises of deciding, for a given Young shape, in which characteristics the corresponding Weyl module is not irreducible. It has been shown (see [2,9]) that the solution of this problem is related to the rank of a matrix with integer entries, built up by considering standard Young tableaux of the given shape.

In the present paper we first exhibit some theoretical results, based on a new presentation of Weyl modules [2], which imply that a matrix of smaller size can be equivalently considered. Next, we present an algorithm which constructs such matrices and specifies in which characteristics there is no full rank. In particular, in §2 we recall the basic combinatorial facts about Young tableaux; in §3 we define Schur and Weyl modules, and summarize the fundamental results which yield the crucial property (Theorem 7) which the algorithm is based on. In §4 we describe the algorithm and its implementation in FORTRAN 77 on CRAY Y-MP8/432 of CINECA Computing Centre [6]. A sequential version has been devised, and numerical experiments are presented and discussed.

2. Young tableaux

Let $E=\{e_1,e_2,\ldots,e_n\}$ be an alphabet, namely, a finite, linearly ordered set, and let Mon(E) be the free monoid generated by E. Elements **w** of Mon(E) are called *words*; if $\mathbf{w} \in$ Mon(E), $\mathbf{w}=x_1x_2\ldots x_k$, $x_i \in E$, we say that **w** has length k. The *content* of a word $\mathbf{w} \in$ Mon(E) is the function cont(**w**) from E to the set of non-negative integers that assigns to each x in E the number cont(**w**;x) of its occurencies in **w**.

Example: if E={a,b,c,d} and **w**=abacc, then **w** has length 4, and cont(**w**;a)=2, cont(**w**;b)=1, cont(**w**;c)=2, cont(**w**;d)=0.

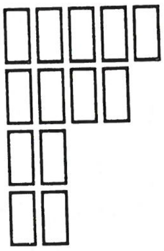
A word $\mathbf{w}$ will be said *simple* whenever cont($\mathbf{w}$;x)≤ 1 for every $x \in E$. If $\mathbf{v},\mathbf{w} \in$ Mon(E), set $\mathbf{v} \sim \mathbf{w}$ whenever cont($\mathbf{v}$)=cont($\mathbf{w}$).

Let $\mathbf{v},\mathbf{w} \in$ Mon(E); if $\mathbf{v},\mathbf{w}$ are simple words such that $\mathbf{v} \sim \mathbf{w}$, and if $\mathbf{v}=x_1x_2\ldots x_k$, then $\mathbf{w}=x_{\sigma(1)}x_{\sigma(2)}\ldots x_{\sigma(k)}$ for some permutation σ of $\{1,2,\ldots,k\}$; we define sign($\mathbf{v},\mathbf{w}$)=sign(σ). In all other cases, set sign($\mathbf{v},\mathbf{w}$)=0.

Example: if E={a,b,c,d}, $\mathbf{v}$=bcda, $\mathbf{w}$=abdc, then $\mathbf{v} \sim \mathbf{w}$, and sign($\mathbf{v},\mathbf{w}$)=1.

A *Young shape* λ is a finite sequence of positive integers $(\lambda_1,\lambda_2,\ldots,\lambda_k)$, such that $\lambda_i \geq \lambda_j$ whenever i<j, namely, a partition of the integer $n=\lambda_1+\lambda_2+\ldots+\lambda_k$; in symbols, $\lambda \vdash n$. Shapes will be ordered lexicographically from left to right.

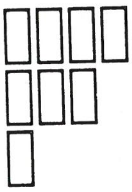
A Young shape is usually seen as an array consisting of λ_1 boxes in the first row, λ_2 boxes in the second row, and so on; for example, the diagram



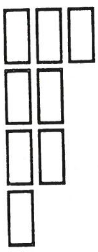
represents the shape (5,4,2,2).

The *conjugate shape* $\tilde{\lambda}=(\tilde{\lambda}_1,\tilde{\lambda}_2,\ldots,\tilde{\lambda}_h)$ of λ is defined by setting $\tilde{\lambda}_j$ to be equal to the number of entries λ_i of λ such that $\lambda_i \geq j$.

Example: if λ=(4,3,1):



the conjugate shape of λ is $\tilde{\lambda}$=(3,2,2,1):



Given a Young shape λ, a *Young tableau of shape* λ relative to the alphabet {a,b,c,d,...} is any way of filling the boxes of the shape with symbols of the set; for example,

a	b	c	a	b
b	c	c	a	
e	f			

is a Young tableau of shape (5,4,2).

Formally, a *Young tableau of shape* $\lambda=(\lambda_1,\lambda_2,...,\lambda_k)$ *over the alphabet* E is a finite sequence of words $T=(w_i)$, $i=1,2,...,k$, where $w_i \in \text{Mon}(E)$ is of length λ_i. We shall write $sh(T)=\lambda$. The *content* of the Young tableau $T=(w_i)$ is the function

$$\text{cont}(T) = \sum_i \text{cont}(w_i).$$

Let $T=(w_i)$ be a Young tableau of shape $\lambda=(\lambda_1,\lambda_2,...,\lambda_k)$; writing

$$w_1=x_{11}\, x_{12}\, \ldots\ldots$$
$$w_2=x_{21}\, x_{22}\, \ldots\ldots$$
$$\ldots$$
$$w_k=x_{k1}\, x_{k2}\, \ldots\ldots$$

set

$$\tilde{w}_i=x_{1i}\, x_{2i}\, \ldots x_{ki} \qquad \text{for } i=1,2,...,\lambda_1.$$

The Young tableau $\tilde{T}=(\tilde{w}_i)$, of shape $\tilde{\lambda}$, is called the *conjugate tableau* of T.

<u>Example</u>: given the tableau $T=(w_i)$, i=1,2,3,4, with

w_1 = b a c e d
w_2 = a c c c
w_3 = b e
w_4 = c b,

its conjugate tableau is $\tilde{T}=(w'_i)$, i=1,2,3,4,5, with

w'_1 = b a b c
w'_2 = a c e b
w'_3 = c c
w'_4 = e c
w'_5 = d.

A Young tableau $T=(w_i)$ is said to be *standard* if, for every i, writing $w_i=x_1x_2...x_p$, we have $x_h<x_k$ whenever h<k, and, writing $\tilde{w}_i=y_1y_2...y_q$ for the i-th word of the conjugate tableau $\tilde{T}$ of T, we have $y_h \leq y_k$ whenever h<k.

<u>Example</u>: the tableau

T =

a	b	d	e
b	c	e	
c	d		

is standard, while

$$T' = \begin{array}{|c|c|c|c|} \hline c & e & d & b \\ \hline b & c & e \\ \cline{1-3} a & d \\ \cline{1-2} \end{array}$$

is not standard.

Given a shape $\lambda=(\lambda_1,\lambda_2,\ldots,\lambda_k)$ with $\lambda_1 \leq |E|$, the *Deruyts tableau* of shape λ over E will be the (standard) tableau $Der(\lambda)=(w_i)$, with

$$w_i = e_1 e_2 \ldots e_{\lambda_i} \quad , i=1,2,..,k.$$

For example, the Deruyts tableau of shape (4,3,2) over the alphabet {a,b,c,d,...} is

$$\begin{array}{|c|c|c|c|} \hline a & b & c & d \\ \hline a & b & c \\ \cline{1-3} a & b \\ \cline{1-2} \end{array} .$$

A Young tableau is said to be *co-standard* whenever its conjugate tableau is standard. For example, the tableau

$$\begin{array}{|c|c|c|c|} \hline a & a & c & d \\ \hline c & c & e \\ \cline{1-3} d & e \\ \cline{1-2} \end{array}$$

is co-standard.

We recall the definitions of two equivalence relations over the set of all Young tableaux of a given shape over E, which will be frequently used in the sequel:

Let $S=(v_i)$, $T=(w_i)$ be Young tableaux over E; we say that S is *row-equivalent* to T, in symbols

$$S \sim_r T,$$

whenever $sh(S)=sh(T)$ and $v_i \sim w_i$ for every i.
Similarly, we say that S is *column-equivalent* to T, in symbols

$$S \sim_c T,$$

whenever $\tilde{S} \sim_r \tilde{T}$.

Example: the tableaux

$$S = \begin{array}{|c|c|c|} \hline a & b & e \\ \hline c & d \\ \cline{1-2} b & c \\ \cline{1-2} \end{array} \qquad T = \begin{array}{|c|c|c|} \hline b & e & a \\ \hline d & c \\ \cline{1-2} c & b \\ \cline{1-2} \end{array}$$

are row-equivalent, while

$$S = \begin{array}{l} \boxed{a}\boxed{b}\boxed{e} \\ \boxed{c}\boxed{d} \\ \boxed{b}\boxed{c} \end{array} \qquad\qquad T' = \begin{array}{l} \boxed{c}\boxed{b}\boxed{e} \\ \boxed{b}\boxed{c} \\ \boxed{a}\boxed{d} \end{array}$$

are column-equivalent.

If $S=(v_i)$, $T=(w_i)$ are Young tableaux of the same shape, set

$$\text{sign}\,(S,T) = \prod_i \text{sign}\,(v_i,w_i).$$

Needless to add, $\text{sign}\,(S,T)\neq 0$ if and only if S and T are row-equivalent.

Let S,T be Young tableaux over E, with sh(S)=sh(T); set

$$J(\boxed{S}\,|\,\boxed{T}) = \sum_{X\sim_c S}\ \sum_{Y\sim_c T} \text{sign}(X,Y).$$

it is immediately seen that the integer $J(\boxed{S}\,|\,\boxed{T})$ is non zero only if the tableaux S and T have the same content.

Example: let

$$S = \begin{array}{l} \boxed{a}\boxed{b}\boxed{e} \\ \boxed{c}\boxed{d} \end{array} \qquad\qquad T = \begin{array}{l} \boxed{a}\boxed{c}\boxed{e} \\ \boxed{b}\boxed{d} \end{array} \ ;$$

there are four tableaux which are column-equivalent to S, namely:

$$S = \begin{array}{l} \boxed{a}\boxed{b}\boxed{e} \\ \boxed{c}\boxed{d} \end{array}, \qquad S_1 = \begin{array}{l} \boxed{a}\boxed{d}\boxed{e} \\ \boxed{c}\boxed{b} \end{array}, \qquad S_2 = \begin{array}{l} \boxed{c}\boxed{b}\boxed{e} \\ \boxed{a}\boxed{d} \end{array}, \qquad S_3 = \begin{array}{l} \boxed{c}\boxed{d}\boxed{e} \\ \boxed{a}\boxed{b} \end{array}$$

and four tableaux which are column-equivalent to T, namely:

$$T = \begin{array}{l} \boxed{a}\boxed{c}\boxed{e} \\ \boxed{b}\boxed{d} \end{array}, \qquad T_1 = \begin{array}{l} \boxed{a}\boxed{d}\boxed{e} \\ \boxed{b}\boxed{c} \end{array}, \qquad T_2 = \begin{array}{l} \boxed{b}\boxed{c}\boxed{e} \\ \boxed{a}\boxed{d} \end{array}, \qquad T_3 = \begin{array}{l} \boxed{b}\boxed{d}\boxed{e} \\ \boxed{a}\boxed{c} \end{array} \ ;$$

among these, the only pairs of row-equivalent tableaux are the following:

$$S_1 \sim_r T_1 \qquad \text{and} \qquad S_2 \sim_r T_2;$$

moreover, $\text{sign}(S_1, T_1) = -1 = \text{sign}(S_2, T_2)$; hence,

$$J(\boxed{S}\,|\,\boxed{T}) = -2.$$

3. Schur modules and Weyl modules

Let K be an infinite field of any characteristic. Let $A=\{a_1,a_2,...,a_p\}$, $B=\{b_1,b_2,...,b_q\}$ be two alphabets. We create "double variables" $(a_i|b_j)$, $i=1,2,...,p$, $j=1,2,...,q$, which can be organized in a $p\times q$ matrix **M**, and consider the polynomial algebra $S_K[(a_i|b_j)]$ over K generated by these variables. To each word in Mon(A) $w=x_1x_2...x_k$ we associate the polynomial pol(w) in $S_K[(a_i|b_j)]$ obtained by taking the determinant of the minor of **M** consisting of the first k columns and the rows of indices $x_1,x_2,...,x_k$. In other words,

$$\mathrm{pol}(w) = \sum_{\substack{y_1y_2...y_k\in \mathrm{Mon}(A)\\ y_1y_2...y_k\sim w}} \mathrm{sign}(y_1y_2...y_k,w)\,(y_1|b_1)(y_2|b_2)...(y_k|b_k).$$

Example:

$$\mathrm{pol}(a_1a_3) = (a_1|b_1)(a_3|b_2) - (a_3|b_1)(a_1|b_2) =$$

$$= \det\begin{pmatrix}(a_1|b_1) & (a_1|b_2)\\ (a_3|b_1) & (a_3|b_2)\end{pmatrix}.$$

Now, to a Young tableau $T=(w_i)$, $i=1,2,...,n$, over A we associate the polynomial in $S_K[(a_i|b_j)]$

$$\mathrm{pol}(T) = \mathrm{pol}(w_1)\,\mathrm{pol}(w_2)\,...\,\mathrm{pol}(w_n).$$

Given a shape λ, the *Schur module* Schur(λ) relative to λ is defined as the linear span over K of pol(T), as T ranges over the set of all Young tableaux of shape λ over the alphabet A.

The general linear group GL(p,K) acts on the ring $S_K[(a_i|b_j)]$ by left multiplication, and it can be shown that Schur(λ) is invariant under this action, namely, it is a GL(p,K)-module.[see, e.g.,1,2]

We now recall two fundamental results about Schur modules.

Theorem 1 The set P={pol(T); T standard tableau of shape λ over A} is a basis of the module Schur(λ). More specifically, if T is any tableau of shape λ over A, the polynomial pol(T) can be uniquely expressed as a linear combination with integer coefficients of elements of P.

Proof: see, e.g., [1,2].

Theorem 2 The module Schur(λ) has a unique minimal GL(p,K)-submodule, namely, the cyclic submodule Cyc(λ) generated by the polynomial pol(Der(λ)). Moreover, the module Cyc(λ) is linearly spanned by the set of elements

$$\sum_{X\sim_c S} \mathrm{pol}(X),$$

as S ranges over the set of all standard tableaux of shape λ over A.

In particular, if K is of characteristic zero, for every shape λ the module Schur(λ) is irreducible, namely, it has no proper submodule; hence, in characteristic zero, Cyc(λ)=Schur(λ).

Proof: see, e.g., [2].

We now introduce Weyl modules, that can be described as the analogs of Schur modules in the exterior letterplace algebra $\Lambda_K[<a_i|b_j>]$, which is the exterior algebra over K generated by the double variables $<a_i|b_j>$, $i=1,2,...,p$, $j=1,2,...,q$, here seen as skew-symmetric variables. In this algebra, we can again associate a polynomial to every Young tableau, as follows: first of all, to each word in Mon(A) $w=x_1x_2...x_k$ we associate an element of $\Lambda_K[<a_i|b_j>]$, which we shall denote by skewpol(w) (polynomial in skew-symmetric variables), defined as:

$$\text{skewpol}(w) = \sum_{\substack{y_1y_2...y_k \in \text{Mon}(A) \\ y_1y_2...y_k \sim w}} <y_1|b_1><y_2|b_2>...<y_k|b_k>.$$

Example: we have

$$\text{skewpol}(a_1a_1) = <a_1|b_1><a_1|b_2>$$

and

$$\text{skewpol}(a_1a_2) = <a_1|b_1><a_2|b_2> + <a_2|b_1><a_1|b_2> = \text{skewpol}(a_2a_1).$$

Now, to a Young tableau over A, $T=(w_i)$, $i=1,2,...,n$, can be associated the polynomial in $\Lambda_K[<a_i|b_j>]$

$$\text{skewpol}(T) = \text{skewpol}(w_1)\ \text{skewpol}(w_2)\ ...\ \text{skewpol}(w_n).$$

Given a shape λ, the *Weyl module* Weyl(λ) relative to λ is defined as the linear span over K of skewpol(T), as T ranges over the set of all Young tableaux of shape λ over the alphabet A.

The general linear group GL(p,K) acts on the ring $\Lambda_K[<a_i|b_j>]$ by left multiplication, and it is easily seen that Weyl(λ) is invariant under this action, namely, it is a GL(p,K)-module (see, e. g.,.[1,2,3]).

The main properties of Weyl modules are summarized in the following theorems:

Theorem 3 The set Q={skewpol(T); T co-standard tableau of shape λ over A} is a basis of the module Weyl(λ). More specifically, if T is any tableau of shape λ over A, the polynomial skewpol(T) can be uniquely expressed as a linear combination with integer coefficients of elements of Q.

Proof: see, e.g., [2].

Theorem 4 The module Weyl(λ) has a unique maximal GL(p,K)-submodule, which is denoted by CL(λ). In particular, if K is of characteristic zero, for every shape λ the module Weyl(λ) is irreducible; hence, in characteristic zero, CL(λ)=(0).

Proof: see, e.g., [2,3].

Our next goal is to examine the connections between Schur and Weyl modules. To this aim, given a shape λ, we can define a map

$$\phi : \mathrm{Weyl}(\lambda) \rightarrow \mathrm{Schur}(\tilde{\lambda})$$

by setting, for every co-standard tableau T of shape λ over A,

$$\phi(\mathrm{skewpol}(T)) = \sum_{X \sim_c S} \mathrm{pol}(X)$$

where S is the conjugate tableau of T, and extending by linearity. It is easily seen that ϕ is a GL(p,K)-module homorphism, and, by preceding results, its image is $\mathrm{Cyc}(\tilde{\lambda})$, the minimal submodule of $\mathrm{Schur}(\tilde{\lambda})$. Moreover, we have:

Theorem 5 The kernel of the map ϕ is precisely $\mathrm{CL}(\lambda)$, the unique maximal submodule of $\mathrm{Weyl}(\lambda)$; hence,

$$\mathrm{Cyc}(\tilde{\lambda}) \cong \mathrm{Weyl}(\lambda) / \mathrm{CL}(\lambda) .$$

Proof: see [2].

The preceding results have the following immediate consequences:

1. The module $\mathrm{Weyl}(\lambda)$ is irreducible over a given field K whenever its maximal submodule $\mathrm{CL}(\lambda)$ is the zero module.
2. The irreducibility of $\mathrm{Weyl}(\lambda)$ depends only on the characteristic of the field.
3. Since the bases of the modules $\mathrm{Schur}(\tilde{\lambda})$ and $\mathrm{Weyl}(\lambda)$ have the same cardinality, $\mathrm{Schur}(\tilde{\lambda})$ is irreducible whenever $\mathrm{Weyl}(\lambda)$ is irreducible.

We now state a result which allows us to verify, given a shape λ and a prime p, whether $\mathrm{Weyl}(\lambda)$ is irreducible over a field of characteristic p, or not.

Theorem 6 Let $\{T_1, T_2, \ldots, T_k\}$ be the set of all standard tableaux of shape $\tilde{\lambda}$ over A. The codimension of the module $\mathrm{CL}(\lambda)$ equals the rank over K of the matrix

$$J(\lambda) = (J(\boxed{T_i} \mid \boxed{T_j})), \ i,j=1,2,\ldots k.$$

In particular, the module $\mathrm{Weyl}(\lambda)$ is irreducible if and only if $\det J(\lambda) \neq 0$.

Proof: see [2].

The matrix $J(\lambda)$ has, in general, a very large size; the next result shows that it can be replaced by a matrix of smaller size.

Theorem 7 If the module $CL(\lambda)$ is non zero, then it must contain an element of the kind skewpol(T), where T is a co-standard tableau whose content equals cont(Der(μ)) for some shape $\mu \vdash n$ such that $\mu > \tilde{\lambda}$.

Proof: see [2].

The previous result yields immediately the following consequence, which is the theoretical underpinning of the algorithm presented in the next section.

Corollary Let λn be a shape; set

$$\Lambda = \{\mu \vdash n;\ \mu > \tilde{\lambda}, \mu_1 \leq p\}.$$

For every $\mu \in \Lambda$, let $\{S_1, S_2, \ldots, S_h\}$ be the set of all standard tableaux of shape $\tilde{\lambda}$ over A, with cont(S_i)=cont(Der(μ)) for every i; set

$$J(\lambda;\mu) = (J(\boxed{S_i} \mid \boxed{S_j})),\ i,j=1,2,\ldots h.$$

We have:

$$\det J(\lambda) \neq 0 \Leftrightarrow \prod_{\mu \in \Lambda} \det J(\lambda;\mu) \neq 0.$$

Example: let A={a,b,c} and $\lambda=(2,1)=\tilde{\lambda}$; the standard tableaux of shape $\tilde{\lambda}$ over A are:

$$S_1 = \begin{matrix} a\ b \\ c \end{matrix} \quad S_2 = \begin{matrix} a\ c \\ b \end{matrix} \quad S_3 = \begin{matrix} a\ b \\ a \end{matrix} \quad S_4 = \begin{matrix} a\ c \\ a \end{matrix}$$

$$S_5 = \begin{matrix} a\ b \\ b \end{matrix} \quad S_6 = \begin{matrix} a\ c \\ c \end{matrix} \quad S_7 = \begin{matrix} b\ c \\ b \end{matrix} \quad S_8 = \begin{matrix} b\ c \\ c \end{matrix} .$$

In this case, the set Λ consists only of the shape $\mu=(3)$, and the standard tableaux whose content equals cont(Der(μ)) are S_1 and S_2; hence, by the preceding result, instead of considering the whole matrix $J(\lambda)$, which has size 8, we can consider the 2×2 matrix $J(\lambda;\mu)$.

4. The algorithm and its implementation

We start now to describe the steps of the algorithm:

1. Fix an integer n and a Young shape $\lambda \vdash n$.
2. Construct all shapes $\mu \vdash n$ such that $\mu > \lambda$.
3. For every such shape μ, repeat:
   - 3.1. generate all standard tableaux $\{T_i\,;\ i=1,2,..\}$ over the alphabet $\{1,2,\ldots,n\}$ of shape λ and content = cont(Der(μ));
     - 3.1.1. for every standard tableau T_i, construct all tableaux T' which are column-equivalent to T;
   - 3.2. for every pair of standard tableaux T_i, T_j compute the integer $J(\boxed{T_i} \mid \boxed{T_j})$;
   - 3.3. compute the determinant of the matrix $J(\boxed{T_i} \mid \boxed{T_j})$, $i,j=1,2,\ldots$;

3.4. determine the prime divisors of the determinant.

The implementation of Steps 1 and 2 requires only standard algorithms [10] to determine all the partitions of the integer n, in descending order, until the shape λ is obtained. In Step 3.1 all the standard tableaux of given shape and content are generated by taking into account several combinatorial properties to avoid violations of standardness, hence minimizing computations and storage requirements. For instance, in the first position of each column of a standard tableau we must place the smallest symbol of the alphabet not yet used; a symbol x may occupy a box b only if the number of unused symbols greater than or equal to x is sufficient to fill the boxes lying under and to the right of the box b, and so forth. In Step 3.1.1 we construct all tableaux T' which are column equivalent to a given standard tableau T; all column permutations of the tableau T are obtained by standard algorithms [12]. For every such tableau T' a test is performed to verify whether in the rows of T' there are repeated symbols; in this case, T' is discarded. Otherwise, the rows of T' are sorted in increasing order, and the tableau T'' obtained henceforth is memorized together with sign(T',T''). In Step 3.2, each integer $J(\boxed{T_i} \mid \boxed{T_j})$ is obtained by comparing all the memorized tableaux which are column equivalent to T_i to those relative to T_j. In Step 3.3 the determinant of the matrix $J(\boxed{T_i} \mid \boxed{T_j})$ is computed by a standard Gauss factorization algorithm [11].

The algorithm was implemented in FORTRAN 77, and some experimentations were made on the CRAY Y-MP8/432 supercomputer at CINECA Computing Centre. This is a vector computer with specialized pipelined functional units, vector access to memory using indirect memory addressing is completely supported by hardware features. The four CPU's (6 ns cycle time) are tightly coupled via the shared main memory and five identical groups of registers, called clusters, are used in common by all processors.

The experimentations reported here were run on COS 1.17 operating system, with CFT77 3.0 compiler, the time measurements were done, not in dedicated mode, using the SECOND function. This because, actually, the CRAY runs two operating system: UNICOS 5.0 and COS 1.17. The first uses 3 CPU's and 20 Megawords of memory, the second has 1 CPU and 7 Megawords of memory. The runs were done on COS because UNICOS was not yet completely released.

Future implementations will comprehend a parallel version, developed using the autotaskin facility of the CFT77 compiler under UNICOS, and the investigation of the maximum possible dimemnsions that "shape" can reach.

5. Some numerical experiments

In the following table we present some results of our tests. For every given shape λ in column 1, we give:

- the total number of standard tableaux of shape λ and content = cont(Der(μ)) for some $\mu \in \Lambda$ (column 2);
- the factors of the determinants of the matrices $J(\boxed{T_i} \mid \boxed{T_j})$ (column 3);

- the time used for generating all the partitions $\mu \in \Lambda$ and the standard tableaux of shape λ and content = cont(Der(μ)) (column 4);
- the time used to construct the matrices $J(\boxed{T_i} \mid \boxed{T_j})$, compute their determinants, and determine their respective prime divisors (column 5).

shape	stand. tabl.	factors	time1	time2
2 2 1 1	29	2 5 7	1.913 exp(-3)	2.106
3 2 1	32	2 3 5 54983 1274267	1.106 exp(-3)	1.984
4 1 1 1	40	2 3 7 11 443 194481 6168209959	1.1732 exp(-3)	2.085
3 2 2	47	2 3 5 47 11765106383	5.59 exp(-3)	2.352
6 2	25	2 3 7	1.15 exp(-3)	1.75

The above few examples show how the complexity of the problem is influenced by both the total number of boxes and the shape λ. In particular, the more the shape is far from a "hook" (exactly one row of length greater than 1), the more is the the time required for constructing the matrices $J(\boxed{T_i} \mid \boxed{T_j})$.

References

1. Akin, K., Buchsbaum, D.A. and Weyman, J., Schur functors and Schur complexes, *Adv. in Math.* **44** (1982), 207-278.
2. Barnabei, M., Schur modules, Weyl modules, and Capelli operators, *Adv. in Math.*, in press.
3. Carter, R.W. and Lusztig, G., On the modular representations of the general linear and symmetric groups, *Math. Zeit.* **136** (1974), 193-242.
4. Clausen, M., Letterplace algebras and a characteristic-free approach to the representation theory of the general linear and symmetric groups, I, *Adv. in Math.* **33** (1979), 161-191.
5. Clausen, M., Letterplace algebras and a characteristic-free approach to the representation theory of the general linear and symmetric groups, II, *Adv. in Math.* **38** (1980), 152-177.
6. CRAY Y-MP and CRAY X-MP Multitasking Programmer's Manual, SR-022 E, June 1988.
7. Doubilet, P., Rota, G.-C. and Stein, J., On the foundations of combinatorial theory IX. Combinatorial methods in invariant theory, *Stud. Appl. Math.* **53** (1974), 185-216.

8. Green, J.A., *Polynomial Representations of* GL_n, Lecture Notes in Math., vol 830, Springer-Verlag, 1980.

9. James, G.D. and Kerber, A., *The Representation Theory of the Symmetric Group*, Encyclopedia of Math. Appl., vol. 16, Addison-Wesley, 1981.

10. Knuth, D.E., *The Art of Computer Programming*, vol.3, Addison-Wesley, 1973.

11. Modi, J.J., *Parallel Algorithms and Matrix Computation*, Claredon Press, 1988.

12. Reingold, E.M., Nievergelt, J. and Deo, N., *Combinatorial Algorithms: Theory and Practice*, Prentice-Hall, 1977.

13. Schur, I., *Über eine Klasse von Matrizen, die sich einer gegebenen Matrix zuordenen lassen*, Dissertation, Berlin, 1901.

14. Schur, I., Über die rationalen Darstellungen der allgemeinen linearen Gruppe, *Sber. der Akad. zu Berlin* (1927), 58-75.

15. Weyl, H., *The classical groups*, Princeton Univ. Press, 1946.

16. Young, A., On quantitative substitutional analysis (third paper), *Proc. London Math. Soc.* 2, **28** (1927), 255-292.