QUANTUM THEORY PROJECT
FOR RESEARCH IN ATOMIC, MOLECULAR AND SOLID STATE
CHEMISTRY AND PHYSICS
UNIVERSITY OF FLORIDA, GAINESVILLE, FLORIDA

THE DIRAC GROUPS

By

Victor A. Dulock*

PROGRAM NOTE #8

25 July 1963

*Physics Department, University of Florida

ABSTRACT

A Dirac group is defined and some facts concerning the structure of Dirac groups are discussed. A method is then described for calculating products and inverses of the elements of a given Dirac group using MBLISP.

ACKNOWLEDGMENTS

The functions described in this note were prepared as exercises in an informal course in LISP programming presented by Harold V. McIntosh. It is a pleasure to acknowledge the interest displayed in programming languages and symbolic programming by Professor J. E. Maxfield and the committee directing the operation of the University of Florida computer center, who have granted the IBM 709 computer time used to prepare and verify the programs herein described.

Victor A. Dulock

Gainesville, 25 July 1963

THE DIRAC GROUPS

I. Properties:

Consider n quantites which satisfy a general exchange rule

$$1) \qquad \gamma_i \gamma_j = \omega_{ij} \gamma_j \gamma_i$$

where e.g. for Dirac matrices $\omega_{ij} = -1$. Assume that for each γ_i there exists some integer n_i (not necessarily the same for each different γ_i) such that

$$2) \qquad \gamma_i^{n_i} = \xi_i$$

where ξ_i is a scalar (in the case of Dirac matrices, a multiple of the unit matrix).

Now consider quantities of the form

$$3) \qquad \lambda_i \gamma_i$$

where the λ_i are scalars. Forming all possible products of the quantities in (3) we have

$$4) \qquad \prod_{i=1}^{m} \lambda_i \gamma_i = \lambda_1 \gamma_1 \cdots \lambda_m \gamma_m$$

Note that some of the γ_i may be repeated several times. By choosing an ordering for the γ_i we can write this product in a canonical form using properties (1) and (2) and the fact that the λ_i, ξ_i, and ω_{ij} are scalars,

$$5) \qquad \prod_{i=1}^{m} \lambda_i \gamma_i = \lambda_1 \lambda_2 \cdots \lambda_m \omega_{iI} \omega_{jI} \cdots \omega_{iII} \cdots \xi_I^k \xi_{II}^l \cdots \xi_{IV}^s \gamma_I^{a_I} \gamma_{II}^{a_{II}} \cdots \gamma_N^{a_N}$$

where $a_i < n_i$ the characteristic exponent of γ_i and k is an integer equal to the numbers of γ_1's originally present divided by n_1, etc. for 1,.... Since the λ_i, ξ_i, and ω_{ij} are scalars we have the form

6) $$\mu\, \gamma_I^{a_{II}}\, \gamma_{II}^{a_{II}} \cdots \gamma_N^{a_N}$$

where μ is a scalar coefficient and the γ's form an ordered product. Before proceeding we should note some properties which follow from eqs. (1):

First, if we take the determinant of both sides of (1) we see that ω_{ij} is one of the n roots of unity if the γ_i are n x n matrices. Now if we require the product of 2 of the forms (i.e. eq. (6)) to be of the same form (closure), then the λ_i and ξ_i are also roots of unity.

Note for the Dirac group all $n_i = 2$ are $N = 4$, and the number of distinct possible products (elements) is $2 \cdot 2^4 = 32$. In general it is $2 \cdot 2^N$, all $n_i = 2$.

From now on we drop the Roman numeral subscripts and use Arabic numerals and assume the order of γ's is 1, 2, ... Furthermore since the γ_i are assumed known and also have a particular ordering we can write eq. (6) as an n-tuple whose 1st element is the scalar coefficient and whose subsequent elements are the powers of the γ_i, i.e.,

7) $$\mu\, \gamma_1^{a_1}\, \gamma_2^{a_2} \cdots \gamma_n^{a_n} = (\mu\ a_1 a_2 \cdots a_n)$$

II. Product:

Consider the product of 2 of these elements

8) $$(\lambda\ a_1\ a_2 \cdots a_n)(\mu\ b_1\ b_2 \cdots b_n)$$

$$= (\lambda\mu(\prod_{\substack{i<j \\ i,j=1}}^{n} \omega_{ji}^{a_j b_i})(\prod_{i=1}^{n} \xi_i^{k_i})(a_1+b_1)_{\text{Mod } n_1} \cdots (a_n+b_n)_{\text{Mod } n_n})$$

where k_i is the number of times n_i divides a_i+b_i and n_i is defined in eq. (2). Note that there are $n(n-1)/2$ ω_{ij} each with its characteristic exponent.

Since all of the scalar coefficients are roots of unity, we know that there exists a root of unity ρ such that

9a) $\rho^{c_i} = \lambda_i$

b) $\rho^{r_{ij}} = \omega_{ij}$ where c_i, r_{ij}, and s_i are integers.

c) $\rho^{s_i} = \xi_i$

For example, if we had three roots of unity

$\lambda = e^{2\pi i/M}$ $\omega = e^{2\pi i/L}$ $\xi = e^{2\pi i/R}$

then $\rho = e^{2\pi i/K}$

where K = least common multiple of the product MLR.

One of our problems will be to find ρ.

III. Some Properties of ω_{ij}:

From eq. (1) $\gamma_i\gamma_j = \omega_{ij}\gamma_j\gamma_i$

$$\gamma_j\gamma_i = \omega_{ji}\gamma_i\gamma_j$$

10) hence $\omega_{ij} = \omega_{ji}^{-1}$. However, since the ω_{ij} are roots of unity (hence in general complex)

$$\omega_{ji}^{-1} = \omega_{ji}^*$$

Therefore

11) $$\omega_{ij} = \omega_{ji}^*$$

also $$\gamma_i\gamma_i = \omega_{ii}\gamma_i\gamma_i$$

12) Therefore $$\omega_{ii} = 1$$

From eqs. (11) and (12) we deduce that the matrix formed by the ω_{ij} is hermitian.

Also from (1), if $\gamma_j^{n_j} = \xi_j$ then

13) $$\omega_{ij}^{n_j} = \omega_{ij}^{n_i} = 1$$

At this point we have the following scalars:

a matrix of ω_{ij}'s

a vector of ξ_i's

and whatever λ_j's we care to use.

IV. Restating the Problem in Order to Program it in LISP:

We would like to write a LISP program which would multiply the elements of our group. To do this we must write things in a different form.

Define a matrix W with elements w_{ij} such that

14) $$\omega_{ij} = e^{(2\pi i/K)\, w_{ij}}$$

where $\rho = e^{2\pi i/K}$. For LISP W has the form of a list of the columns

$$W = ((w_{21}\ w_{31} \ldots w_{n1})(w_{32} \ldots w_{n2})(w_{43} \ldots w_{n3}) \ldots (w_{n,n-1}))$$

From the definition of the w_{ij} and the hermitian character of ω_{ij} we see that the W matrix can be made antisymmetric.

Neglecting the multiplicative factor $2\pi i/K$ the product of ω_{ij}'s in eq. (8) in terms of the w's is

15) $$\sum_{\substack{i\ j \\ i,j=1}}^{n} a_j w_{ji} b_i$$

which is almost aWb if we could set all elements above the main diagonal = 0 in W.

To perform the summation of products in (15) we define a LISP function, Quadratic Form - QF which performs this task QF(W X Y) where W is as in (14) and

16) $$X = (a_1 \ldots a_n)$$

17) $$Y = (b_1 \ldots b_n)$$

where the powers to which the γ's are raised in each element of the product are X and Y. We assume positive integers only.

```
(QF (LAMBDA (W X Y)
      (IF (NULL W)
          (DEC (QUOTE 0)
          (I+ (I* (CAR Y)
                  (IP (CDR X)(CAR W)))
              (QF (CDR W)(CDR X)(CDR Y))))))
```

QF uses IP (inner product). IP is a function which calculates the inner product of two vectors, i.e., if the vectors are $A = (a_1 \ldots a_n)$ and $B = (b_1 \ldots b_n)$, IP calculates the scalar $\sum_{i=1}^{n} a_i b_i$.*

Another auxiliary function which we will need is one which, when given the two lists (eqs. (16) and (17)) above and a list of the form

18) $$N = (n_1\ x_1\ n_2\ x_2 \ldots n_n\ x_n)$$

where the n_i are the powers for which $\gamma_i^{n_i} = \xi_i$ and the x_i are the powers to which ρ must be raised to give ξ_i, i.e.,

$$\xi_i = \rho^{x_i} = e^{(2\pi i/K)x_i} ,$$

will give the final powers of the various γ_i mod n_i and a scalar coefficient due to the products of the ξ_i. We call this function PREPRODUCT and define it as follows:

19) (PREPRODUCT (LAMBDA (N X Y)(PREPRODUCT* (DEC (QUOTE 0))(LIST) N X Y)))

---

*H. V. McIntosh, "Program Note No. 6." This Note contains a detailed description of IP and also of other useful arithmetic functions.

```
20)  (PREPRODUCT* (LAMBDA (D L N X Y)(IF (NULL X)
            (CONS D (REVERSE L))
            ((LAMBDA (U V)(PREPRODUCT* (I+ (I* U (CADR N)) D)
                                       (CONS V L)
                                       (CDDR N)
                                       (CDR X)
                                       (CDR Y)))
             ($DIVIDE (2NDVAL ($PLUS (CAR X)(CAR Y)))(CAR N))) )))
     (LIST (LAMBDA L L))
     (REVERSE (LAMBDA (L) (REVERSE* (LIST L))))
     (REVERSE* (LAMBDA (M L)
                  (IF (NULL L)
                      M
                      (REVERSE* (CONS (CAR L) M)
                                (CDR L)))))
```

It should be noted that the lists X and Y must be of equal length, hence if one of the factors in the product has e.g. γ_i missing, then we must explicitly write 0 for a_i or b_i whichever is the case.

We will also need a function which will add an arbitrary number of terms which we now define,

```
21)  (+ (LAMBDA L (IF (NULL L)(DEC (QUOTE 0))(++ L))))
22)  (++ (LAMBDA (L)(IF (NULL L)(DEC (QUOTE 0))
                              (I+ (CAR L)(++(CDR L))) )))
```

Defining l to be that power to which ρ must be raised to give λ in eq. (8) and similarly m for μ, we can now define a function, which we will call DP (Dirac Product) that will give the product of 2 elements as in (8). The element lists will be of the form

23) X $(1\ a_1\ a_2\ a_3\ \ldots\ a_n)$

24) Y $(m\ b_1\ b_2\ b_3\ \ldots\ b_n)$

```
25)     (DP (LAMBDA (X Y) ((LAMBDA (Z) (CONS (REM (+ (CAR X)
             (CAR Y) (QF (W) (CDR X) (CDR Y))
             (CAR Z)) (K)) (CDR Z)))
                (PREPRODUCT (N) (CDR X) (CDR Y)))))
```

where K, the integer which identifies ρ ($\rho = e^{2\pi i/K}$) is defined by

```
26)     (K(LAMBDA ( ) (DEC (QUOTE K))))
```

W is the matrix W defined by

```
27)     (W (LAMBDA ( ) (NUMBETHERE (QUOTE (            )))))
```

and N the alternating list given in (18) is defined by

28) (N (LAMBDA ( ) (NUMBETHERE (QUOTE $(n_1 x_1 n_2 x_2 \ldots)$)))))

K, W, and N are to be given for a particular problem.

We now wish to have a means of finding the inverse of one of the elements, $(\lambda\ a_1\ a_2\ \ldots\ a_n)$. The inverse will be of the form $(\mu\ b_1\ b_2\ \ldots\ b_n)$. We see that

29) $$b_j = n_j - a_j \quad \text{where} \quad \gamma_j^{n_j} = \xi_j.$$

Thus to find the correct powers of the γ's in the inverse we define (DINV* N X*) where X* = $(a_1\ a_2\ \ldots\ a_n)$ = CDR X where X = $(\lambda\ a_1\ a_2\ \ldots\ a_n)$ and N = $(n_1\ x_1\ n_2\ x_2\ \ldots)$ given in eq. (28).

```
30) (DINV* (LAMBDA (N X*) (IF (NULL X*)
        (CONS (REM (I-(CAR N) (CAR X*)) (K)) (DINV*
              (CDDR N) (CDR X*))))))
```

To produce unity the element and its inverse must satisfy

31) $\mu + \lambda + QF + x_1 + x_2 \ldots = nK, \qquad n = 0, 1, 2, \ldots$

32) $\mu = (K - (\lambda + QF \;\; x_1 \;\; x_2 \ldots)_{\text{mod } K})\text{mod } K$

Now only those X_i will contribute for which the corresponding $a_i \neq 0$, i.e., the corresponding γ_i is present in X.

Hence we define

33)
```
(DINV (LAMBDA (X) (IF (NULL X)
      X
      ((LAMBDA (Z) (CONS (REM (I- (K) (REM (+
         (CAR X) (QF (W) (CDR X) Z)(XI (N) (CDR X)))
         (K))) (K)) Z))(DINV* (N) (CDR X))))))
```

where

34)
```
(XI (LAMBDA (N X*) (COND ((NULL N) (DEC (QUOTE O))))
     ((EQ (CAR X*) (DEC (QUOTE O))) (XI (CDDR N) (CDR X*)))
     ((AND) (I+ (CADR N) (XI (CDDR N) (CDR X*)))))))
```