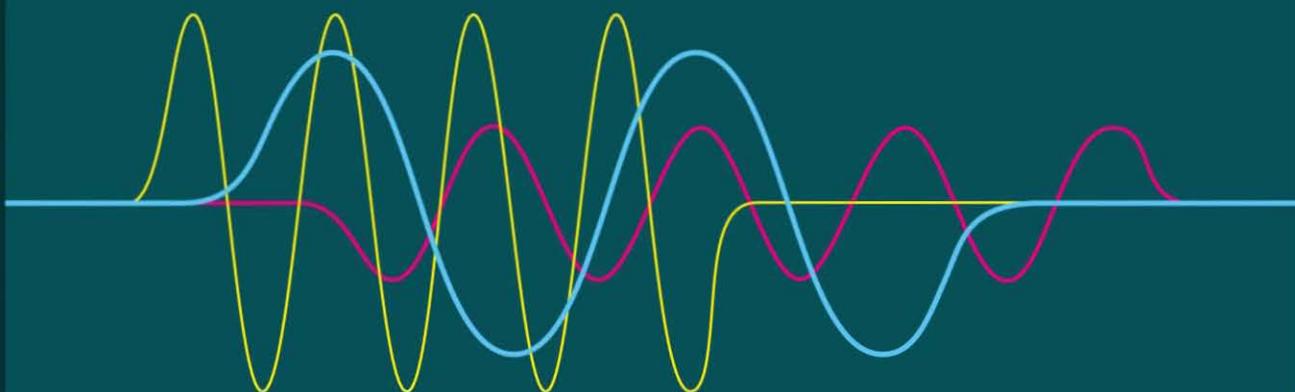




Ámbito Sonoro

Revista del Centro de Investigación
Musical Autónomo CIMA



estructura
caos
arte
totemismo
composición
forma
Deleuze
música
matrices
escucha
aborigen
idiolecto
creación
aerófono
arte sonoro
concepto



MATRIX: UNA LIBRERÍA DE COMPOSICIÓN ASISTIDA PARA EL ENTORNO PWGL, BASADA EN MATRICES COMBINATORIAS

DR. PABLO CETTA

FACULTAD DE ARTES Y CIENCIAS MUSICALES
UNIVERSIDAD CATÓLICA ARGENTINA
CETTA.PABLO@GMAIL.COM

Resumen

El empleo de matrices de grados cromáticos, como método de organización de la altura en la música no tonal, constituye un procedimiento eficaz para la generación de materiales para la composición. En este artículo se describen las propiedades de una librería especialmente creada para el entorno de composición asistida PWGL, destinada a la generación y transformación de matrices combinatorias y a la realización de operaciones con conjuntos de grados cromáticos.

Palabras clave: Composición asistida, PWGL, matrices combinatorias

Abstract

The use of pitch classes matrices as a method of pitch organization in atonal music, is an effective procedure for the generation of compositional materials. This paper describes the properties of a specially created library for the assisted composition environment PWGL, designed to create and transform combinatorial matrices and to perform some operations with pitch class sets.

Key words: assisted composition, PWGL, combinatorial matrices

1. INTRODUCCIÓN

La utilización conjunta de dos o más modos de una serie dodecafónica, obtenidos a través de operaciones de transformación derivadas de los principios de combinatoriedad e invariancia, ha sido el punto de partida para el posterior desarrollo de matrices combinatorias, tanto dodecafónicas como no dodecafónicas.

En la música serial, la superposición de dos modos cualquiera de la misma serie -en dos voces de una textura contrapuntística, por ejemplo- podía generar la repetición de grados idénticos entre voces, y en consecuencia la disminución de la variedad cromática. La aplicación del principio de combinatoriedad¹ se basaba en determinar qué operaciones de transformación de la serie producían modos que, al superponerse, garantizaban una distribución homogénea de los doce grados en el tiempo. Esta distribución, como sabemos, se producía al aplicar un operador mediante el cual las notas del primer hexacordio pasaran, en otro orden, al segundo hexacordio de la transformación, y las del segundo hexacordio al primero. Al comparar ambos modos podía observarse la presencia del total cromático tanto en los primeros hexacordios combinados, como en los segundos².

En el caso de las matrices no dodecafónicas, su utilización no persigue necesariamente la regulación de la variedad cromática. Su empleo se limita a garantizar cierta coherencia interválica entre el aspecto horizontal y el vertical de una configuración de alturas determinada. A diferencia de la concepción serial, el orden de las notas carece de importancia.

Las matrices no seriales están, en general, formadas por el mismo conjunto de grados cromáticos (PCS, por *pitch class sets*) tanto en sus filas como en sus columnas. De ahí que la lectura de las filas -líneas melódicas de la trama contrapuntística- o las columnas -acordes- recurren a las mismas clases interválicas. No obstante, cierto tipo de construcción, que luego veremos, puede presentar dos PCS, uno utilizado horizontalmente y el otro verticalmente.

Por tratarse de configuraciones de altura que presentan homogeneidad interválica en dos dimensiones, el empleo de estas matrices como punto de partida para la composición musical resulta de interés y ofrece múltiples posibilidades.

1 La combinatoriedad fue empleada inicialmente por Arnold Schönberg [1] y, posteriormente, desarrollada y clasificada por Milton Babbitt en diversos artículos.

2 Resulta obvio que la combinatoriedad más inmediata es la que se produce al superponer una serie cualquiera con su retrogradación.

2. MATRICES COMBINATORIAS NO SERIALES

A fin de caracterizar los distintos tipos de matrices y sus modos de construcción vamos a basarnos en la terminología empleada por Robert Morris en su artículo “Combinatoriality without the aggregate” [2]. Una descripción detallada de sus modos de construcción, con ejemplos musicales, puede hallarse en el libro *Elementos de Contrapunto Atonal*, de Pablo Cetta y Pablo Di Liscia [3], y también su aplicación a través de una librería especialmente programada para el entorno de procesamiento en tiempo real *Pure Data*³.

Una primera clasificación de las matrices resulta de la división en simples y complejas. En el primer caso, la construcción de la matriz no depende de las propiedades del PCS elegido como norma generatriz y el resultado tiende a ser una mera permutación de elementos, o bien una yuxtaposición de estructuras acórdicas. En esta categoría se inscriben las matrices conocidas como “cuadrado romano”, tipo “IA”, tipo “IB” y tipo “II”.

Las matrices complejas, por su parte, demuestran amplias posibilidades como materiales para la composición, y se dividen en matrices generadas a partir de cadenas de PCS y matrices creadas a partir de ciclos de un mismo operador.

2.1 Matrices simples

El “cuadrado romano” se construye a partir de las permutaciones circulares de los elementos de un conjunto cualquiera, ubicando cada permutación en una fila distinta. En el ejemplo siguiente, los grados del PCS elegido -numerados del 0 al 9 y con las letras A y B- se ubican de forma arbitraria en las posiciones de la primera fila de una matriz de 4 x 4. Las permutaciones circulares de las posiciones dan el siguiente resultado:

1B5	39		82
39		82	1B5
	82	1B5	39
82	1B5	39	

Según se observa, todas las filas y columnas tienen el mismo contenido, en distinto orden, por supuesto.

Las matrices tipo “IA”, “IB” y “II” constan de un único elemento por posición. La matriz tipo “IA” se obtiene ubicando el PCS elegido en la primera fila y en la primera columna. Las filas o columnas siguientes se calculan transponiendo el PCS

³ <https://puredata.info/>

sobre sus propios grados. En el ejemplo que sigue, el PCS (8, 0, 1, 7, 6) se transporta sobre 0, 1, 7 y 6:

8	0	1	7	6
0	4	5	B	A
1	5	6	0	B
7	B	0	6	5
6	A	B	5	4

Como se aprecia, se trata de una yuxtaposición de acordes de igual estructura, que se reconoce tanto armónica como melódicamente.

Las matrices "IB", por su parte, contienen el PCS original en las filas y su inversión en las columnas, como ocurre en el caso siguiente. La intervállica es la misma en todas las filas y columnas de la matriz (PCS 5-7):

0	4	5	B	A
8	0	1	7	6
7	B	0	6	5
1	5	6	0	B
2	6	7	1	0

Las matrices tipo "II", en cambio, se forman con dos PCS diferentes, uno para las filas y otro para las columnas. Por esta razón, dos PCS pueden dar como resultado matrices rectangulares, si elegimos a cada uno con distinto número de elementos, como en el ejemplo:

A	2	3	9	8
B	3	4	A	9
7	B	0	6	5

2.2 Matrices complejas

Las matrices generadas mediante cadenas de PCS enlazados resultan de gran interés debido a sus características especiales. Una cadena se construye a partir de las particiones de un conjunto dado. Para la realización del ejemplo siguiente tomamos el PCS 5-15 = {0, 1, 2, 6, 8} y sus particiones A: 01|268 con subconjuntos 2-1|3-8 y B: 16|028 con subconjuntos 2-5|3-8. El subconjunto común a ambas (3-8) es lo que nos permite construir la cadena, aplicando transposiciones, inversiones y retrogradaciones a las particiones elegidas:

01	268	A
268	07	T ₆ R(B)
07	15B	T ₁ I(B)
15B	67	T ₇ I R(A)
67	028	T ₆ (A)
028	16	T ₀ R(B)
16	57B	T ₇ I(B)
57B	01	T ₁ I(A)

A partir de esta disposición estamos en condiciones de crear la cadena, que es cerrada y apta para generar una matriz, dado que el primer elemento y el último son iguales:

01|268|07|15B|67|028|16|57B|01

De la mera observación del procedimiento -que no detallamos aquí por cuestiones de espacio- se desprende la necesidad de contar con una aplicación que nos asista en la realización de estas tareas. Una vez obtenida la cadena, la construcción de la matriz es relativamente simple:

01	268		
	07	15B	
		67	028
57 B			16

Las matrices generadas a partir de ciclos de un mismo operador también revelan amplias posibilidades al utilizarlas en la composición. Se construyen ubicando el PCS utilizado como norma en la primera posición y las transformaciones sucesivas a través de un mismo operador (T_n) a lo largo de la diagonal. Las dimensiones de la matriz dependen del operador elegido; en la tabla siguiente se describen estas relaciones:

Operador	Dimensiones de la matriz
T ₆ - T _n I	2 x 2
T ₄ - T ₈	3 x 3
T ₃ - T ₉	4 x 4
T ₂ - T _A	6 x 6

Para el siguiente ejemplo utilizaremos la técnica mencionada. El conjunto elegido es PCS 4-3 = {2, 1, 5, 4}, y el operador de transposición es T_3 :

2154			
	4578		
		78AB	
			12AB

En principio, puede parecer que tal matriz no tiene posibilidades de aplicación debido al escaso nivel de distribución de las notas en las posiciones de la matriz. Para mejorar esta situación podemos recurrir a un procedimiento de intercambio gradual de elementos entre posiciones (*swapping*). En las matrices siguientes, si se observan detenidamente las operaciones realizadas, podrá notarse que luego de cada intercambio todas las filas y columnas conservan el mismo PCS, en este caso el 4-3:

154			2
	4578		
		78AB	
2			1AB

15	4		2
4	578		
		78AB	
2			1AB

1	45		2
45	78		
		78AB	
2			1AB

1	45		2
45	8	7	
	7	8AB	
2			1AB

1	45		2
45	8	7	
	7	8B	A
2		A	1B

1	45		2
45	8	7	
	7	8	AB
2		AB	1

La utilización en orden sucesivo de las matrices anteriores puede emplearse para la composición de un proceso gradual, como el que se muestra a continuación:



Figura 1: fragmento musical que emplea una matriz generada por ciclos de un operador.

3. LA LIBRERÍA MATRIX

*Matrix*⁴ es una librería de objetos externos destinada al lenguaje visual de composición asistida PWGL (*PatchWork Graphic Library*)⁵, desarrollado por Mikael Laurson, Mika Kuuskankare, Vesa Norilo y Kilian Sprotte. PWGL se basa en Common Lisp, CLOS y OpenGL, y fue desarrollado en el entorno Lispworks⁶. Junto

4 La librería puede descargarse libremente de <http://www.pablocetta.com/matrix>

5 Para descargar PWGL y obtener documentación sobre este programa consultar <http://www2.siba.fi/PWGL/>

6 Lispworks es una plataforma para el desarrollo de software en lenguaje LISP. Para más información ver <http://www.lispworks.com/>

a OpenMusic⁷ forma parte de los entornos de programación para la composición más empleados en la actualidad.

Una vez instalada nuestra librería deberíamos ver el menú de objetos, de forma similar al que se muestra en la figura siguiente.

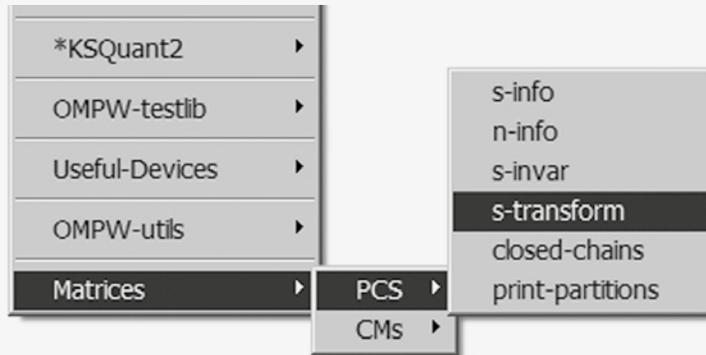


Figura 2: Menú de acceso a los objetos de la librería *Matrix*.

3.1 Operaciones con conjuntos de grados cromáticos (PCS)

A la librería de matrices combinatorias hemos agregado algunos objetos relacionados con *pitch class sets*, a fin de integrarlos al entorno. A continuación describiremos sus características principales.

s-info: dado un conjunto cualquiera en forma de lista, representado por números (0 a 11) o de forma alfanumérica (0 a 9, A y B), devuelve su nombre, su forma prima y su vector interválico.

La notación entre paréntesis es propia del lenguaje LISP. El conjunto 0, 1, 4, 8, por ejemplo, puede representarse (0 1 4 8), o bien, (0 (1 4) 8). La distribución de paréntesis determinará la cantidad de elementos por posición en la construcción de las matrices combinatorias. En el primer caso, la notación hace referencia a cuatro posiciones (un elemento por posición), mientras que en el segundo caso se refiere a tres posiciones, de las cuales la segunda alberga dos elementos.

n-info: brinda la misma información que el objeto anterior, pero a partir de especificar el nombre del conjunto. En el ejemplo de la figura siguiente, al ingresar el texto 4-z15 y al evaluar (presionando la tecla “v” en PWGL, con el objeto seleccionado) la función devuelve la lista (|4-Z15| (0 1 4 6) (1 1 1 1 1)) en la ventana de salida.

⁷ OpenMusic fue desarrollado en el IRCAM por Carlos Agon, Gérard Assayag y Jean Bresson. Para más información consultar <http://repmus.ircam.fr/openmusic/home>



Figura 3: Objetos para obtener información sobre PCS.

El *patch* que veremos a continuación enumera una lista de nombres de PCS, obtiene su forma prima (segundo elemento de la lista que devuelve *n-info*), ubica las notas en la octava central (sumando 60 a cada grado) y representa los resultados en notación musical.

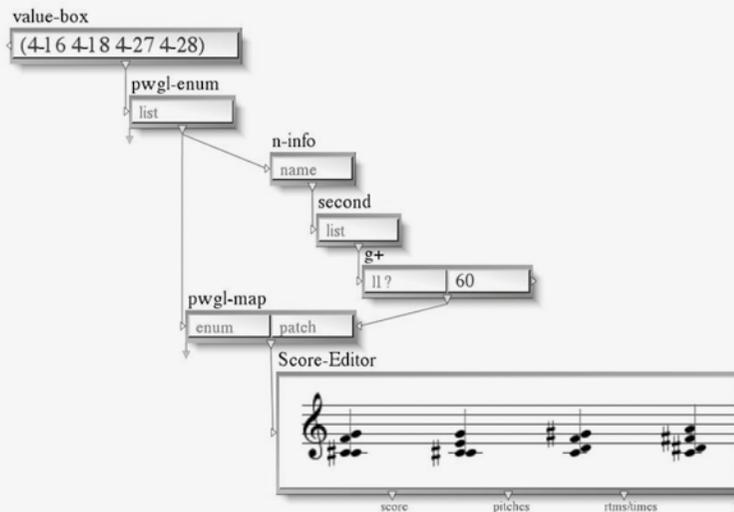


Figura 4: Utilización de *n-info* en la programación.

s-invar: devuelve la lista de invariantes por transposición e inversión de un conjunto dado. El conjunto (1 B 5 3 9 7), a modo de ejemplo, demuestra invariancia para los siguiente operadores ("T2" "T4" "T6" "T8" "TA" "T0I" "T2I" "T4I" "T6I" "T8I" "TAI").

s-transform: calcula el conjunto que resulta de aplicar una transformación por *TnI* a un conjunto dado. El conjunto (1 (2 3 A) 5) se convierte en (2 (3 4 11) 6) por T1, por ejemplo.

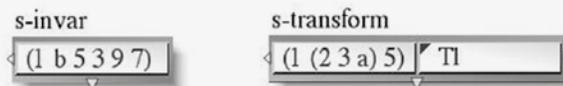


Figura 5: Invariancia de un conjunto y transformación por *TnI*.

closed-chains: calcula una lista de cadenas cerradas, producidas por medio de las particiones de un conjunto dado. Los conjuntos dato (norma de la cadena) deben tener 5 o 6 elementos, dado que son los más empleados en la construcción de matrices combinatorias mediante esta técnica. El PCS (0 1 2 7 9), por ejemplo, genera la siguiente lista resultado: (((0 9) (1 2 7) (0 5) (3 4 10) (5 8) (3 9 10) (2 9) (0 1 7) (2 5) (0 6 7) (5 10) (3 8 9) (0 9)) ((1 2) (0 7 9) (2 8) (4 7 9) (2 3) (1 8 10) (1 7) (3 6 8) (1 2))). Según se observa, la lista comprende dos sublistas (dos cadenas generadas), y a su vez cada cadena está formada por sublistas que albergan a las particiones enlazadas. Obsérvese además que cada cadena comienza y termina con la misma partición -(0 9) en la primera cadena y (1 2) en la segunda- lo cual pone de relieve la igualdad entre comienzo y fin, propia de las cadenas cerradas.

Dado que la estructura en árbol de las cadenas cerradas lo permite, vamos a observar el resultado en notación musical. Para ello, emplearemos un objeto que forma parte de nuestra librería, y que luego analizaremos con mayor detalle: **cm-score**. Haciendo doble *click* sobre el objeto *Score-editor* accedemos a la partitura completa. El primer pentagrama corresponde a la primera cadena y el segundo a la cadena restante, lo cual no significa que deban ejecutarse simultáneamente. Se trata simplemente de un modo de visualización.



Figura 6: Generación de cadenas cerradas y representación en notación musical.

print-partitions: imprime en la ventana de salida las particiones de un conjunto, especificando los PCS que las componen y sus características.

3.2 Operaciones con matrices combinatorias

Los objetos vinculados con matrices se dividen de acuerdo a su función: generación, transformación e impresión:

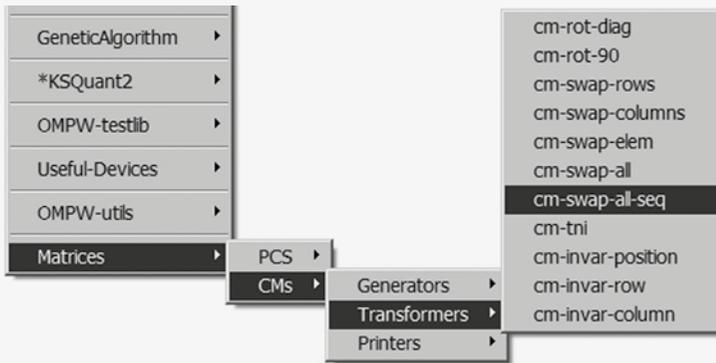


Figura 7: Menú de objetos relacionados con matrices, ordenados según su función.

3.2.1 Generación de matrices

cm-roman: construye una matriz del tipo cuadrado romano. El conjunto de partida puede tener solo un nivel de sublistas, ya que varios grados cromáticos pueden compartir la misma posición en la matriz. También puede contener el símbolo *nil*, que va a determinar una posición vacía. El conjunto (6 (2 4) nil 8), por ejemplo, genera una matriz en la cual la primera fila está formada por el grado 6 en la primera columna, los grados 2 y 4 en la segunda columna, *nil* en la tercera y el grado 8 en la cuarta. La posición con *nil* equivale a una posición vacía (silencio). El resto de la matriz, según ya sabemos, se genera a partir de permutaciones circulares de la primera fila. Obsérvese que la notación rítmica obedece simplemente a la distribución de grados por posición, y no a la forma que deba tener en la composición:

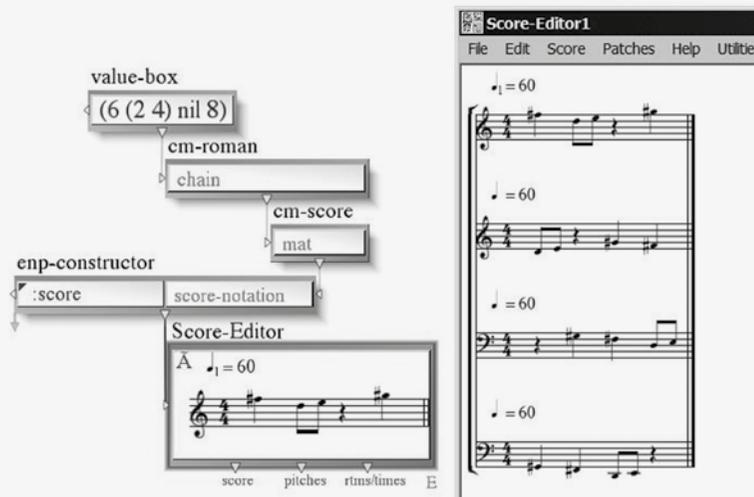


Figura 8: Cuadrado romano obtenido con la función *cm-roman*.

cm-type1a: genera una matriz del tipo 1a. La matriz debe tener un solo elemento por posición, por lo cual el conjunto de partida no debe contener sublistas. Ejemplo (0 3 A 7 5).

cm-type1b: genera una matriz del tipo 1b a partir de un conjunto dado. Aquí también, la lista que especifica el conjunto dato no debe contener sublistas.

cm-type2: genera una matriz del tipo 2 a partir de dos conjuntos, ambos especificados por medio de listas sin sublistas. El primer conjunto (1 9 B, en el ejemplo de la figura) determina la norma horizontal, mientras que el segundo conjunto (2 A 4 5) establece la norma vertical de la matriz:

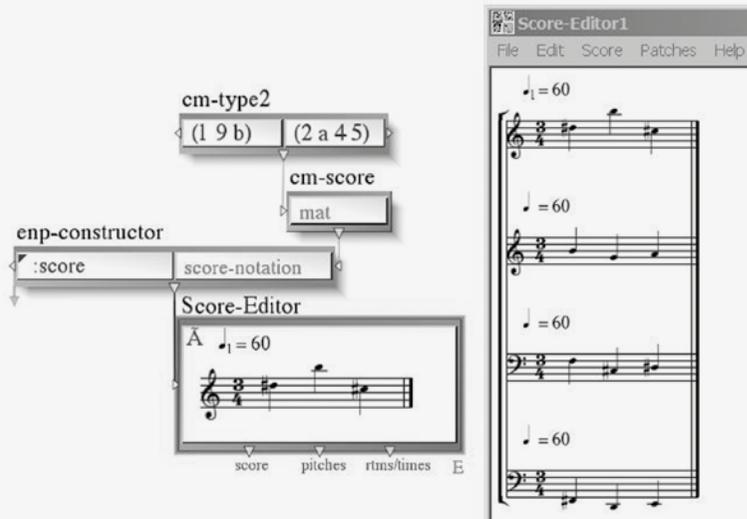


Figura 9: Matriz rectangular obtenida con *cm-type2*.

cm-op-cycles: genera una matriz por ciclos de un mismo operador. Para ello, utilizamos un conjunto de partida cuyos elementos ocupan la primera posición de la matriz (fila 1, columna1). Este conjunto de partida se especifica con doble paréntesis, dado que todos sus grados cromáticos pertenecen a la misma posición. Ejemplo: ((1 7 A B)). Como segundo argumento se debe determinar el operador de transformación que genera la matriz. Los operadores T6 y Tn1 producen una matriz de 2x2; los operadores T4 y T8 una matriz de 3x3; los operadores T3 y T9 una matriz de 4x4 y los operadores T2 y TA, una matriz de 6x6. En la figura que sigue se observa el empleo del operador T9, y la matriz cuyas dimensiones son 4x4.

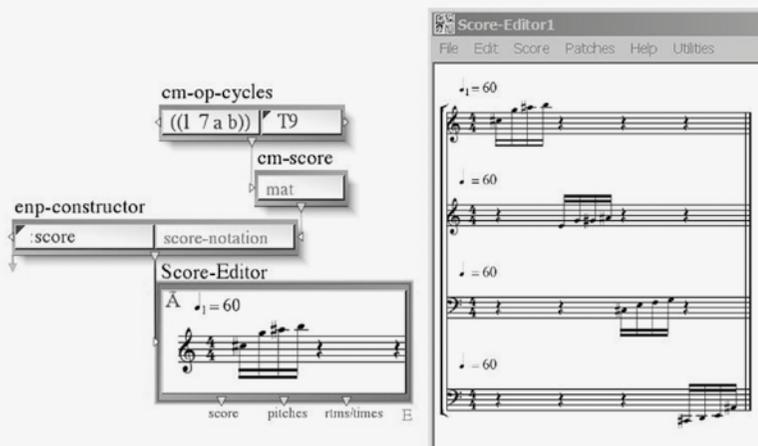


Figura 10: Generación de una matriz por ciclos de un operador.

cm-chains: construye una serie de matrices por cadenas. El conjunto dato es el que servirá de norma de las particiones a generar, y debe declararse a través de una lista simple, por ejemplo (0 1 2 7 9). Dado que el programa emplea todas las particiones útiles del conjunto especificado, las matrices generadas se ubican dentro una superlista que las contiene. Para acceder a cada una de ellas se puede recurrir al objeto *nth*, que devuelve la *n*ésima sublista (matriz) deseada. La figura siguiente lo ilustra.

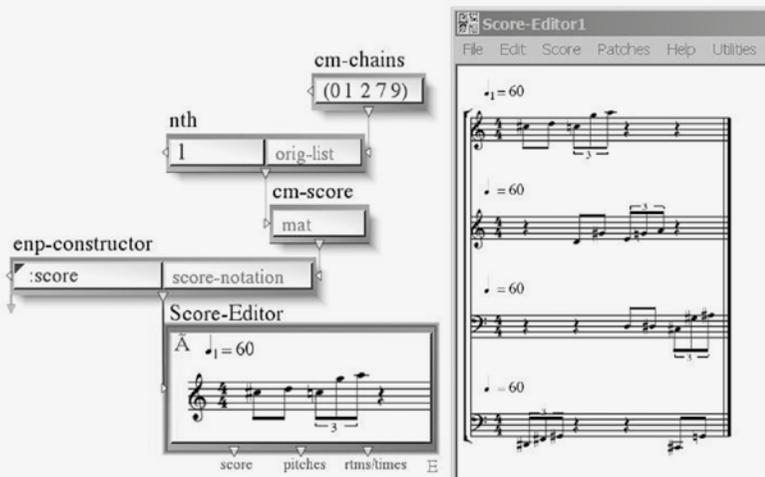


Figura 11: Obtención de matrices por cadenas y selección de la *n*ésima con *nth*.

La norma empleada en el ejemplo produce dos matrices distintas. Utilizando el argumento 0 en *nth* obtenemos la primera de ellas, mientras que con 1 obtenemos la segunda matriz. La cantidad total de matrices generadas por el objeto *cm-chains* se informa a través de la ventana de salida de PWGL.

3.2.2 Transformación de matrices

Los siguientes objetos sirven para realizar transformaciones sobre las matrices generadas. Los procesos comprenden rotaciones, intercambio de elementos entre posiciones, transposiciones, inversiones, y conservación de elementos por invariancia.

cm-rot-diag: rota la matriz por sus diagonales. Se puede especificar la diagonal deseada a través de las opciones *Diag1* o *Diag2*.

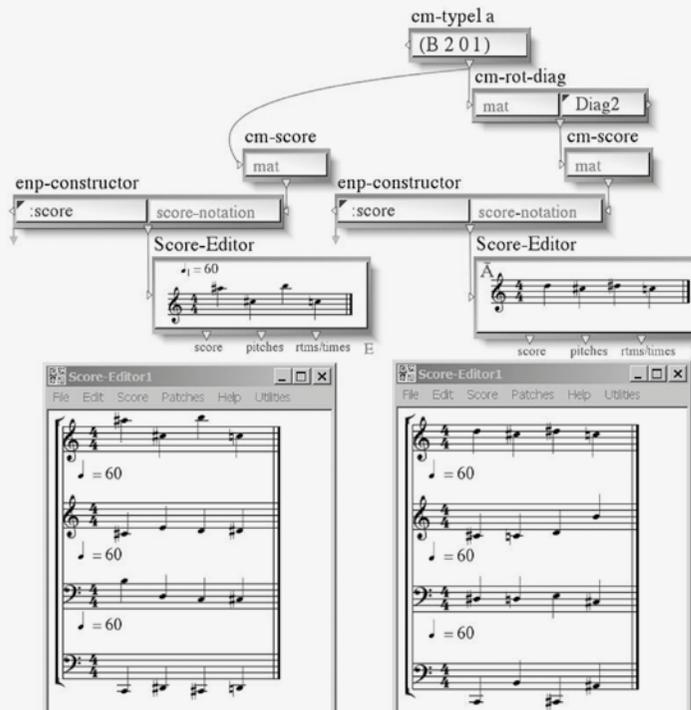


Figura 12: Comparación entre una matriz tipo 1a y su rotación por la diagonal.

cm-rot-90: rota la matriz 90 grados a la izquierda o a la derecha.

cm-swap-rows: intercambia las filas de una matriz. Los argumentos 1 y 3, por ejemplo, determinan que la primera fila ocupe el lugar de la tercera, y viceversa.

cm-swap-columns: intercambia las columnas de una matriz.

cm-swap-elem: intercambia elementos iguales entre posiciones. Aquí se debe determinar el elemento (grado cromático a intercambiar), la fila y la columna en la que se encuentra el primer elemento, y la fila y columna del segundo elemento.

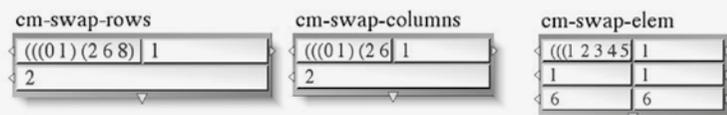


Figura 13: Objetos de intercambio de filas, columnas y elementos.

cm-swap-all: intercambia elementos entre posiciones hasta obtener una distribución lo más homogénea posible en la matriz. En la figura siguiente se observa una matriz construida mediante ciclos de un operador, en la cual los grados cromáticos se ubican en la diagonal. Luego de la transformación puede verse la distribución alcanzada:

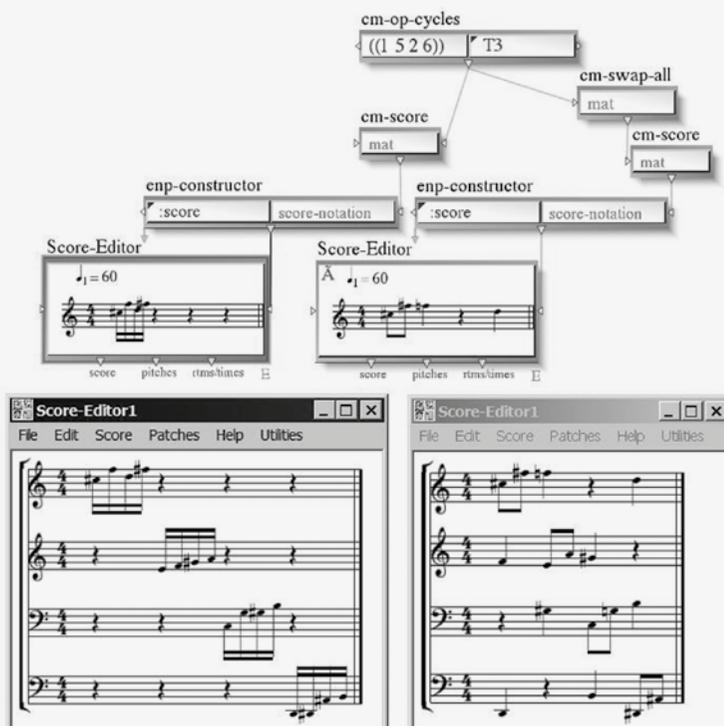
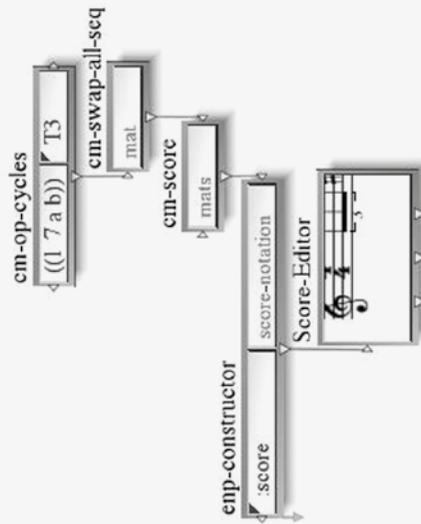


Figura 14: Objeto *cm-swap-all* para distribuir uniformemente los elementos en las posiciones.

cm-swap-all-seq: intercambia elementos entre posiciones hasta obtener una distribución homogénea en la matriz, pero en este caso, guardando las matrices generadas a lo largo del proceso de *swapping*. El resultado es una lista de matrices, por lo cual, para acceder a cada una de ellas se deben utilizar los objetos *first*, *second*, *third*, etcétera, o bien *nth*. También puede emplearse directamente el objeto *cm-score* para representarlas, dado que éste reconoce que se trata de una lista de matrices de iguales dimensiones y las imprime.



Score-Editor1
File Edit Score Patches Help Utilities

Figura 15: Objeto *cm-swap-all-seq* y representación de la secuencia con *cm-score*.

cm-invar-position: genera una serie de matrices transpuestas en las cuales sólo una posición permanece invariante. La posición elegida se determina especificando en el objeto el número de fila y el número de columna. Para acceder a cada una de las matrices de la serie debemos usar *first*, *second*, u otros objetos de selección.

cm-invar-row: genera una serie de matrices transpuestas en las cuales sólo la fila especificada permanece invariante (ver figura).

cm-invar-column: genera una serie de matrices transpuestas en las cuales sólo la columna especificada permanece invariante.

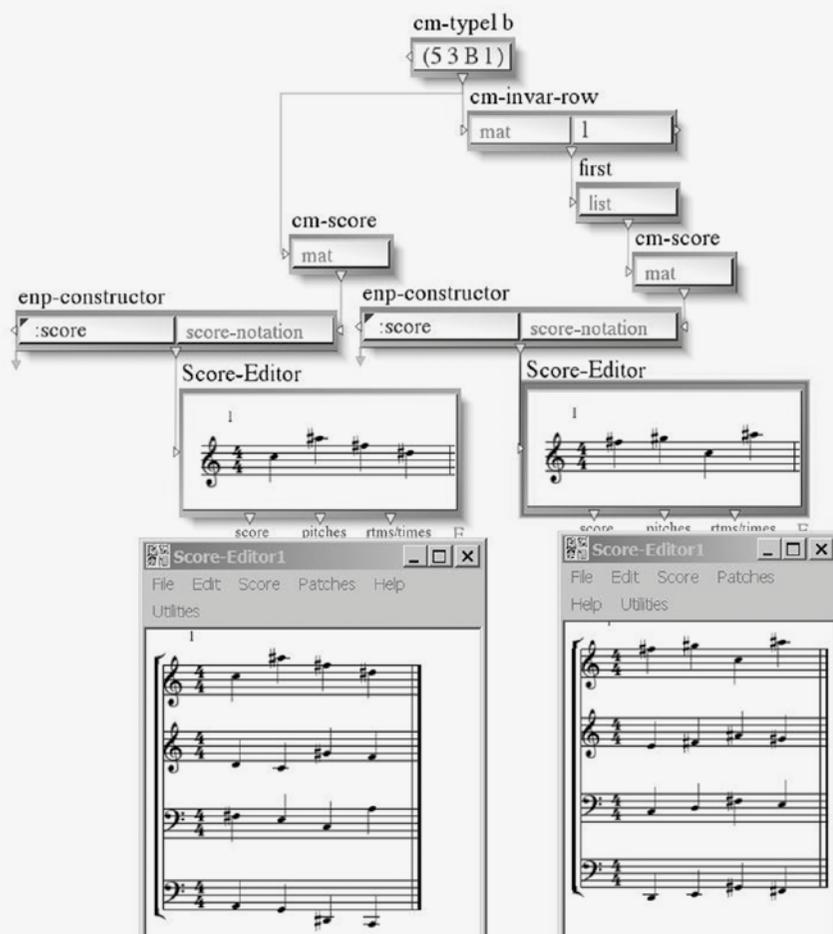


Figura 16: Uso de *cm-invar-row* para mantener invariante la primera fila de una matriz tipo *1b*.

cm-add: encadena matrices combinatorias y las devuelve en una lista que las contiene. El objeto puede expandirse presionando la tecla "+", permitiendo así la suma de varias de ellas. En el ejemplo siguiente, generamos una matriz del tipo cuadrado romano y obtenemos luego dos transposiciones de la misma. Por medio del objeto *cm-add*, logramos la representación en notación musical del conjunto de matrices, utilizando *cm-score*.

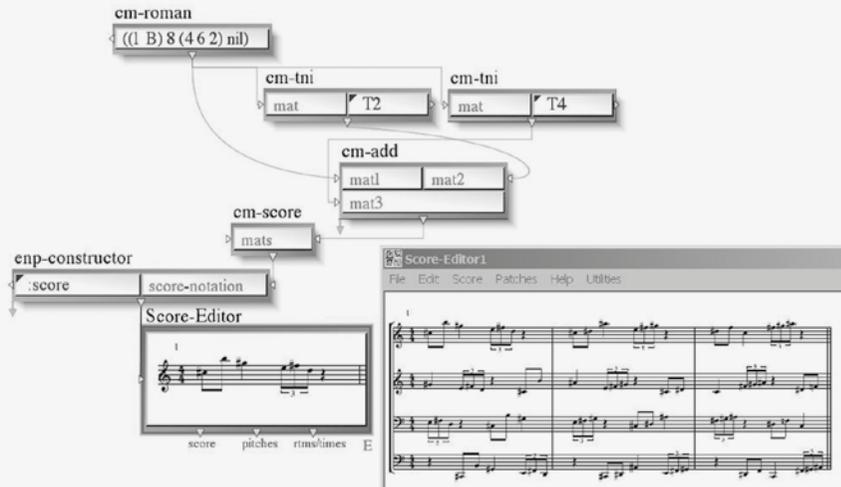


Figura 17: Suma de matrices con el objeto *cm-add*.

3.2.3 Impresión de matrices

Los siguientes objetos sirven a la representación gráfica de las matrices combinatorias:

cm-print: imprime una matriz o una lista de matrices en la ventana de salida de PWGL, en forma de texto.

cm-score: ya utilizado en los ejemplos anteriores, genera el código necesario para que los objetos de PWGL *enp-constructor* y *Score-Editor* puedan representar en notación musical una matriz o una lista de matrices combinatorias. Para que el objeto pueda imprimir una lista de matrices, todas ellas deben tener las mismas dimensiones. Si esta condición no se cumple el objeto lo informa en la ventana de salida de PWGL.

cm-norm-print: imprime los PCS que conforman la norma horizontal y vertical de una matriz combinatoria en la ventana de salida de PWGL.

4. CONCLUSIONES

La técnica de los conjuntos de grados cromáticos, de interés tanto para el análisis musical como para la composición, resulta impracticable sin programas de computación que resuelvan su implementación. Lo mismo ocurre con las matrices combinatorias, si deseamos utilizarlas compositivamente a través de la experimentación, pues su generación y transformación no es tarea fácil, y no todas las matrices posibles de obtener garantizan un resultado interesante desde el punto de vista musical. Por estas razones, el desarrollo de las herramientas adecuadas para el manejo de estas técnicas se torna imprescindible. Según pudimos ver a partir de ejemplos simples, PWGL es un entorno de programación sumamente interesante para el desarrollo de aplicaciones destinadas a la asistencia en la composición. Las posibilidades de integración de los objetos de la librería *Matrix* a esta plataforma son prácticamente ilimitadas, gracias a la versatilidad del lenguaje.

Bibliografía

- [1] Schönberg, Arnold (1963): *El estilo y la idea*. Madrid: Editorial Taurus.
- [2] Morris, Robert (1983): *Combinatoriality without the aggregate, Perspectives of New Music*, Vol. 21 No. 1/2, pp. 432-486.
- [3] Cetta, Pablo y Oscar Di Liscia (2009): *Elementos de Contrapunto atonal*. Buenos Aires: Educa.