LA COMPOSICIÓN ALGORÍTMICA COMO UN PROBLEMA DE SATISFACCIÓN DE RESTRICCIONES

Rubén Hinojosa Chapel
Grupo de Tecnología Musical
Universidad Pompeu Fabra
Ocata 1, 08003 Barcelona, España
(2002) Revisado: Noviembre de 2005
contacto@hinojosachapel.com

Resumen

Históricamente los músicos siempre han utilizado reglas para componer, reglas que, en muchos casos, constituyen prohibiciones (o restricciones). Los elementos de la música con los cuales trabaja el compositor, así como las combinaciones posibles de estos elementos, constituyen un conjunto finito enormemente grande de valores. La función del compositor es seleccionar un subconjunto de valores de este gran conjunto, darles una determinada organización temporal que cumpla las reglas musicales, y crear una obra de arte. Desde un punto de vista matemático, este proceso se relaciona directamente con uno de los campos de mayor actualidad de la Inteligencia Artificial: la Programación de Restricciones. El presente trabajo pretende reflexionar sobre la relación estrecha entre composición musical y programación de restricciones, lo que nos llevará a pensar que esta última podría constituir una prometedora técnica para el desarrollo de sistemas informáticos de composición algorítmica. Este texto es resultado de las conferencias impartidas por el profesor Dr. Héctor Geffner, como parte de su asignatura de doctorado *Resolución de Problemas en Inteligencia Artificial.*

1 Introducción

"Schönberg –ha dicho alguien– es un inventor de teoremas, que siempre parece estarlos demostrando en el encerado, con fórmulas geométricas... Lo cierto es que la música no puede nunca recurrir a las fuerzas de lo inconsciente, como la poesía o la pintura; es un arte –aunque me atemorice el término– terriblemente euclidiano. Y una fuga de Bach, como un motete de Victoria, pueden siempre traducirce en el encerado por medio de fórmulas geométricas." [1]

El elevado grado de abstracción y formalización que posee la música, en sus aspectos teóricos, ha llevado a menudo a compararla con las matemáticas. No sin razón, a una de las más importantes disciplinas musicales, la armonía, suele elevársele a la categoría de ciencia. Se ha afirmado, por ejemplo, que los músicos crearon los sistemas de coordenadas antes que los estudiosos de los números, ya que una partitura musical es realmente un sistema de coordenadas, donde son representados valores de frecuencia en el eje de las ordenadas, y el decursar del tiempo en el eje de las abscisas.

El compositor musical, en su labor creativa, trabaja con los denominados elementos de la música, como son: melodía, armonía, ritmo, timbre, articulación, dinámica, forma, etc.. Podríamos comparar este proceso con el que sigue un cocinero que crea una nueva receta. El cocinero dispone de multitud de ingredientes que puede utilizar, pero sólo

usará una pequeña parte de estos. Los seleccionará del conjunto de ingredientes que puede adquirir, por ejemplo, en uno o varios mercados, y luego los irá reuniendo, preparando, cocinando, etc., según las reglas y proporciones estudiadas y desarrolladas durante siglos de arte culinario. Por ejemplo, en general, la cantidad de sal que puede utilizar tiene una cota superior muy baja, ya que la comida podría quedarle salada si sobrepasase esta cota.

Pensemos ahora en un compositor que desea crear una nueva pieza. Él dispone de un conjunto enormemente grandes de valores (de frecuencias o alturas, duraciones, intensidades, timbres o instrumentos, etc.) que puede combinar, tanto horizontal como verticalmente, de muchas, muchísimas formas diferentes. La explosión combinatoria es enorme. La elevadísima cantidad de piezas musicales conocidas lo confirma. Entonces, ¿qué criterio seguir para seleccionar un subconjunto de valores que pueda ser manipulable y factible de organizar lógicamente? Cuando decimos "organizar lógicamente" pensamos en lo que solemos llamar lógica musical. Pero, ¿qué es lo que define una lógica musical?

Pídale a una persona que no haya estudiado música, que se siente al piano e intente tocar varias teclas durante un par de minutos. Luego repita el experimento con alguien que sí haya estudiado. ¿Por qué al escuchar al primer "ejecutante", pensamos en cualquier cosa menos en música? ¿Por qué el segundo nos sonará inconfundiblemente a música? Respuesta: el primero carece de un sistema de reglas, leyes y restricciones, mientras que el segundo sí las posee. El estudio y aplicación de las reglas, leyes y restricciones musicales durante siglos, ha llevado a la creación de muchos subconjuntos ordenados de elementos y valores musicales, que normalmente reconocemos como piezas de música.

Además de todos los factores subjetivos que intervienen en la creación de una composición musical, entre otros, la inspiración, la intuición o la experiencia, el músico selecciona "parámetros" objetivos con los cuales va diseñando poco a poco cómo será su pieza. Normalmente decidirá si utilizará una textura polifónica u homofónica, cómo la estructurará, para cuántos instrumentos será, como será la armonía, etc.. En muchos casos deberá ceñirse a restricciones más o menos estrictas. Por ejemplo, no debe escribir una nota para un instrumento que no pueda ejecutarla. Cada instrumento posee un registro o rango de valores de notas que puede ejecutar. En general, el estudio de la teoría musical conlleva el estudio de numerosas reglas y restricciones. Veamos a continuación un ejemplo concreto, extraído de un libro de armonía [2]:

> ***Enlace del acorde de tónica con el de dominante y con el de subdominante***
>
> *Antes de intentar los más sencillos enlaces de acordes, es preciso tener presente ciertas* ***reglas*** *generales:*
>
> *1- La nota común a los acordes que se trata de enlazar (do-mi-sol sol-si-re) debe permanecer en la misma voz (...)*
> *2- Todas las voces, menos el bajo, deben ser conducidas en lo posible por grados conjuntos, evitando el uso frecuente de marchas por grados disjuntos o saltos melódicos.*
> *3- El soprano y el bajo deben marchar por movimiento contrario con preferencia.*

4- *La distancia entre el bajo y el tenor puede rebasar la octava en ocasiones; mas para obtener una sonoridad bien equilibrada se debe considerar la octava como la máxima distancia admisible entre el soprano y el alto, y sobre todo, entre el alto y el tenor.*

Hemos visto que el compositor musical, en su trabajo de creación, manipula muchas variables, cada una de las cuales posee su propio dominio de valores. Estos valores se ven restringidos por un conjunto de reglas, que deben ser satisfechas simultáneamente. Pero esta situación que acabamos de describir constituye directamente lo que se conoce como *Problema de Satisfacción de Restricciones*, un tipo de problema que estudia una de las disciplinas de la Inteligencia Artificial: la *Programación de Restricciones*.

2 Programación de Restricciones

La *Programación de Restricciones* (*Constraint Programming, CP*) es el estudio de sistemas computacionales basados en restricciones. La idea de la CP consiste en resolver problemas mediante la declaración de restricciones (requerimientos) relativas al area de estos problemas, y por lo tanto, encontrar la o las soluciones que satisfagan todas las restricciones.

"La Programación de Restricciones representa uno de los mayores acercamientos que ha logrado la Ciencia de la Computación al Santo Grial de la programación: el usuario declara el problema, y la computadora lo resuelve." (E. Freuder, citado en [3])

La representación de un problema mediante restricciones suele ser muy flexible, debido a que estas pueden ser añadidas, eliminadas o modificadas. El proceso de programación consta de dos fases: generación de una representación del problema original como *Problema de Satisfacción de Restricciones* (*Constraint Satisfaction Problem, CSP*), y búsqueda de la solución de este último. Un *CSP* se define como:

- un conjunto de *variables* $X = \{x_1,...,x_n\}$,
- para cada variable x_i, un conjunto finito D_i de posibles valores (su *dominio*),
- un conjunto de *restricciones* que limitan los valores que pueden tomar las variables simultáneamente sobre sus respectivos dominios.

Una *solución de un CSP* es la asignacion de un valor a cada variable, de manera que se satisfagan todas las restricciones al mismo tiempo. Se dice que un CSP es *consistente* si posee una o más soluciones. Al resolver un CSP, es posible que deseemos encontrar:

- una solución cualquiera, no importa cuál sea,
- todas las soluciones posibles,
- una solución óptima, o al menos una buena solución, dada alguna función objetivo definida en términos de algunas o todas las variables.

Las soluciones de un CSP pueden encontrarse buscando sistemáticamente a través de posibles asignaciones de valores para cada variable. Los métodos de búsqueda se dividen en dos amplios grupos: aquellos que atraviesan el espacio de soluciones parciales (o asignación de valores parcial), y aquellos que exploran completamente el espacio de asignación de valores de forma estocástica.

La tarea de la programación de restricciones consiste en reformular el problema inicial como un CSP, y resolverlo mediante métodos generales o relativos a un dominio particular determinado. Usualmente los métodos generales se relacionan con técnicas para reducir el espacio de búsqueda, y con métodos de búsqueda específicos. La idea fundamental consiste en reducir un CSP dado, a otro que sea equivalente; es decir, que posea el mismo conjunto de soluciones, pero que sea más facil de resolver. Este proceso es conocido como *Propagación de las Restricciones*. Los algoritmos de propagación de las restricciones reducen el espacio de búsqueda y, por lo tanto, limitan la explosión combinatoria.

Por otra parte, generalmente los métodos relativos a un dominio particular, suelen proveerse como algoritmos de propósito específico o paquetes especializados, llamados usualmente solucionadores de restricciones (*Constraint Solver*s). Algunos ejemplos son:

- un programa que resuelve sistemas de ecuaciones lineales
- un paquete de programación lineal
- una implementación del algoritmo de unificación, piedra angular de las demostraciones automáticas de teoremas.

Hemos visto que siempre los músicos han estado resolviendo problemas de satisfacción de restricciones para crear sus obras musicales, aunque sin darles esa denominación. En realidad, la Programación de Restricciones, como disciplina científica, ha formalizado y desarrollado el fundamento teórico-matemático-algorítmico de uno de los tantos procesos intelectuales que se desarrollan en la mente humana, posibilitando así su simulación (programación) mediante sistemas computacionales de Inteligencia Artificial, y la resolución de problemas de la vida real de forma automática.

La evidencia de que las teorías musicales se pueden expresar como CSPs, ha llevado a algunos investigadores a desarrollar sistemas de composición o armonización automáticos, basados en estos formalismos. También se ha utilizado esta teoría para fundamentar sistemas automáticos de análisis musical. En [6], los autores describen una herramienta interactiva que utiliza la propagación de restricciones en un sistema experto, para la composición musical. En este trabajo se modela el contrapunto como un CSP.

Otro de los proyectos desarrollados ha sido implementado en un lenguaje de programación de restricciones llamado Oz, y se denomina COMPOzE. Con este sistema queremos ejemplificar cómo se puede, en la práctica, aplicar la teoría de los CSPs a la composición algorítmica.

3 COMPOzE: Composición Musical mediante Programación de Restricciones

COMPOzE, desarrollado por Martin Henz *et al.* ([7], [8]), es un sistema interactivo basado en Programación de Restricciones para la creación de sencillas piezas musicales a cuatro voces, a partir de un plan de desarrollo musical. Este plan describe el flujo armónico y algunas características de la composición deseada. El sistema posee una interfaz de usuario gráfica que permite seleccionar reglas musicales mediante manipulación directa. El proceso de composición, así como las soluciones, se representa

gráficamente. Además, el usuario puede escuchar y comparar las soluciones. Como resultado final, COMPOzE genera un archivo MIDI.

El plan musical está integrado por una progresión armónica, por ejemplo: **T**ónica, **S**ubdominante, **D**ominante, **T**ónica. Además, también lo integran otros parámetros, como la tonalidad, tempo, ritmo, la forma de los acordes, si tienen notas alteradas, etc.. La secuencia de acordes que se genera no solamente debe seguir la progresión armónica, también debe cumplir las reglas (restricciones) de composición programadas. Estas reglas incluyen restricciones para las notas (registro), para los acordes (prohibición del cruce de voces, distancia del bajo), para las secuencias (compensación de saltos, figurado del bajo), etc.. Ejemplos de estas reglas son las siguientes:

- **Prohibición del cruce de voces:** las voces dentro de un acorde no deben cruzarse, es decir, una voz inferior no debe poseer una nota de altura mayor que la de una voz superior. Por ejemplo: el bajo no debe tocar una nota de altura superior a la del tenor, dentro de un acorde.
- **Ley de salto:** El salto de una voz desde un acorde hasta su acorde vecino, que exceda una distancia dada, debe ser contrapuesto en el próximo acorde por un salto de uno o dos pasos en la dirección opuesta.

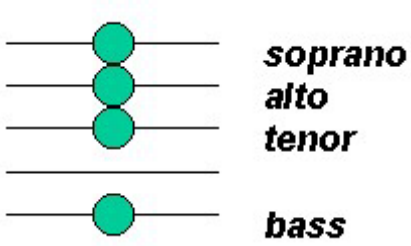


Figura 1: Acorde

Según la hipótesis inicial, el marco más apropiado para formalizar la composición musical lo aporta la teoría de satisfacción de restricciones. Al representar este problema como un CSP, quedaría expresado de la siguiente manera:

- En general tenemos n x v variables, donde n es el número de acordes en la secuencia, y v el número de notas en cada acorde. En este caso tenemos cuatro voces en cada acorde, llamadas: **B**ajo, **T**enor, **A**lto y **S**oprano. Nombraremos a las variables como: B_i, T_i, A_i, S_i, donde $i \in \{1,\ldots, n\}$.
- El dominio de las variables viene dado por un rango de alturas (notas) ejecutables.
- Las funciones armónicas y las reglas de composición se pueden formular como restricciones, que pueden abarcar una o más variables. Por ejemplo, la prohibición del cruce de voces se puede expresar mediante la siguiente restricción:
  $\forall i \in \{1,\ldots, n\}, B_i \leq T_i \leq A_i \leq S_i$

Visto de una manera más formal, el *Problema de Satisfacción de Restricciones* quedaría formulado en los siguientes términos:

Variables y dominios

- 4 x n variables B_i, T_i, A_i, S_i, $i \in \{1,\ldots, n\}$,
- con dominio de valores enteros $\{0,\ldots,60\}$.

Restricciones:

- Funciones armónicas: $B_i \bmod 12 \in \{0, 4, 7\}$
- Registro: $\forall i \;\; 0 \leq B_i \leq 20$
- Distancia del bajo: $\forall i \;\; B_i + DistBajo \leq T_i, A_i, S_i$
- Compensación de salto:

$$\forall i \;\; S_i - S_{i+1} \geq salto \rightarrow S_{i+2} - S_{i+1} \in \{1, 2\}$$
$$S_{i+1} - S_i \geq salto \rightarrow S_{i+1} - S_{i+2} \in \{1, 2\}$$

El objetivo de la Propagación de Restricciones es restringir el conjunto de posibles valores que pueden tomar las variables, mediante la aplicación de las restricciones, hasta que finalmente se encuentre un solo valor para cada variable. El conjunto de posibles valores se almacena en una estructura denominada *depósito de restricciones* (*constraint store*). Por ejemplo, el hecho de que la primera nota del primer acorde se debe tomar entre las primeras 25 alturas de la escala, se expresa mediante la siguiente restricción: $B_1 \in \{0,\ldots,24\}$ en el depósito de restricciones. Restricciones más complejas son expresadas mediante los propagadores, los cuales inspeccionan el depósito de restricciones y lo amplían.

Un propagador inspecciona el depósito respecto a un conjunto fijo de variables. Cuando se excluyen valores del dominio de una de ellas, esta acción puede añadir información sobre las otras variables en el depósito, o sea, se puede ampliar el depósito insertándole más restricciones. Como ejemplo consideremos la prohibición del cruce de voces. Para el primer acorde se puede expresar instalando los siguientes tres propagadores:

```
B1 =<: T1 T1 =<: A1 A1 =<: S1
```

Para explicar cómo ellos pueden ampliar el depósito de restricciones, asumamos que

$A_1 \in \{30,\ldots,45\}$ y $S_1 \in \{25,\ldots,60\}$

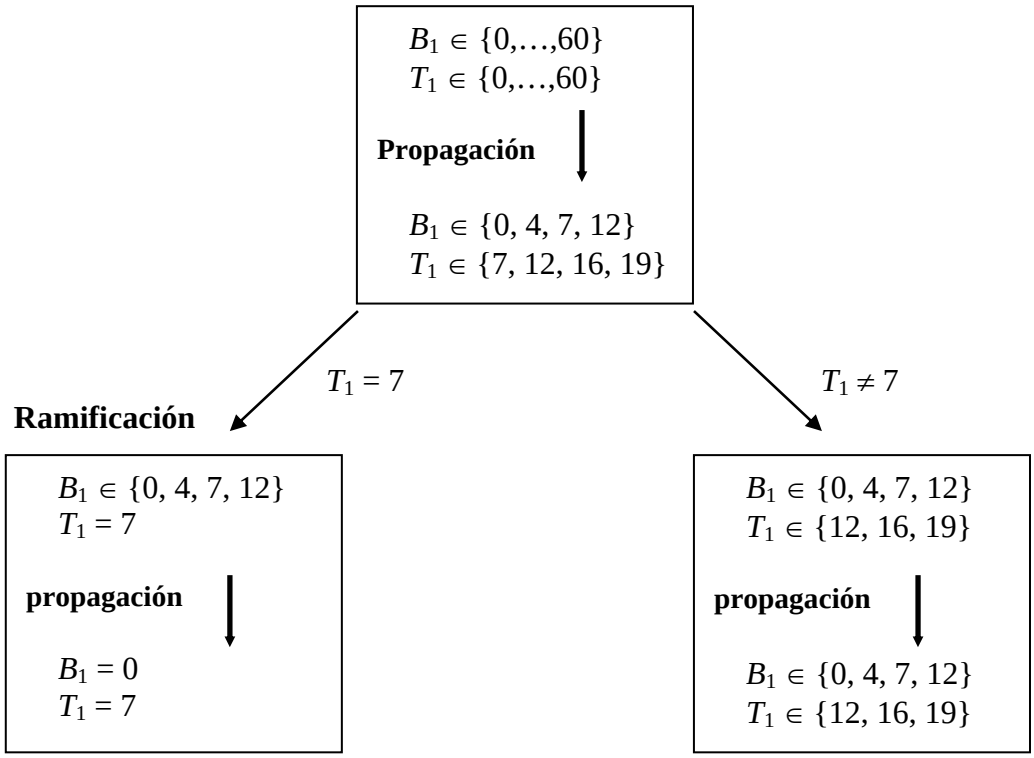
Entonces el tercer propagador excluirá los valores 25,...,29 del dominio de S_1, reduciendo su dominio a $\{30,\ldots,60\}$. Inversamente, si posteriormente se sabe que $S_1 \in \{30,\ldots,40\}$, entonces A_1 será restringida al nuevo dominio $A_1 \in \{30,\ldots,40\}$. Nótese que este propagador permanece activo, esperando por información acerca de A_1 ó S_1. Sólo se detendrá cuando se tenga la certeza de que no volverá a ampliar el depósito. Véanse los siguientes gráficos:

Restricciones iniciales:
$B_1 \in \{0,\ldots,60\}$
$T_1 \in \{0,\ldots,60\}$
$DistBajo = 6$

↓ **Propagación:**
$B_1, T_1 \bmod 12 \in \{0, 4, 7\}$
$0 \leq B_i, T_i \leq 20$
$B_1 + DistBajo \leq T_1$

Resultado de la propagación:
$B_1 \in \{0, 4, 7, 12\}$
$T_1 \in \{7, 12, 16, 19\}$
$DistBajo = 6$

BÚSQUEDA



COMPOzE toma como entrada un plan musical, y devuelve como resultado una o varias composiciones que siguen dicho plan, a la vez que cumple los criterios definidos por el usuario. El sistema le permite al usuario decidir, para cada regla musical, si ésta debe ser ignorada (`off`), obedecida estrictamente (`hard`), u obedecida preferiblemente (`soft`), a través de un valor (`weight`) comprendido entre 0 y 100.

La implementación utiliza la técnica de ramificar y delimitar (`branch-and-bound`) para minimizar la violación de las reglas poco estrictas (`soft`). Si existen varias de estas reglas, se utilizan sus correspondientes valores de peso (`weight`) para determinar su importancia. El explorador de Oz visualiza el árbol de búsqueda mientras que el usuario puede, de forma interactiva, escuchar las soluciones generadas haciendo *click* con el ratón sobre los nodos de soluciones representados gráficamente.

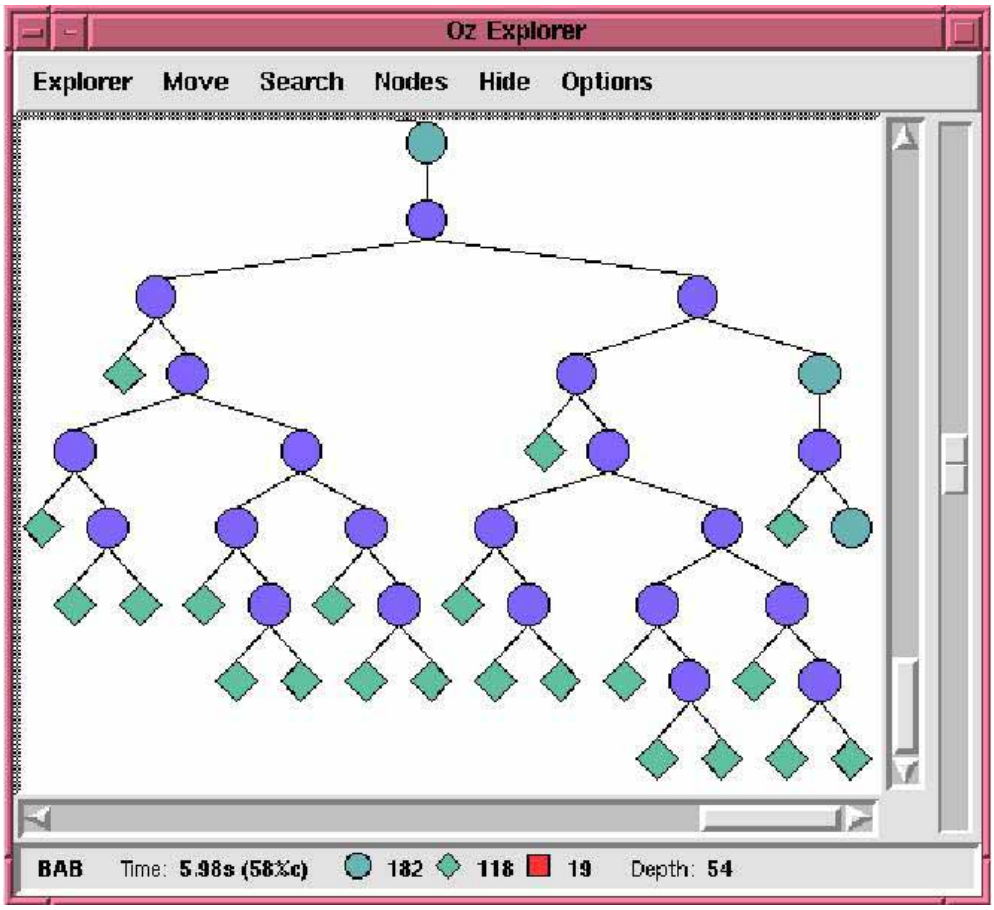


Figura 2: El explorador de Oz visualiza el árbol de búsqueda

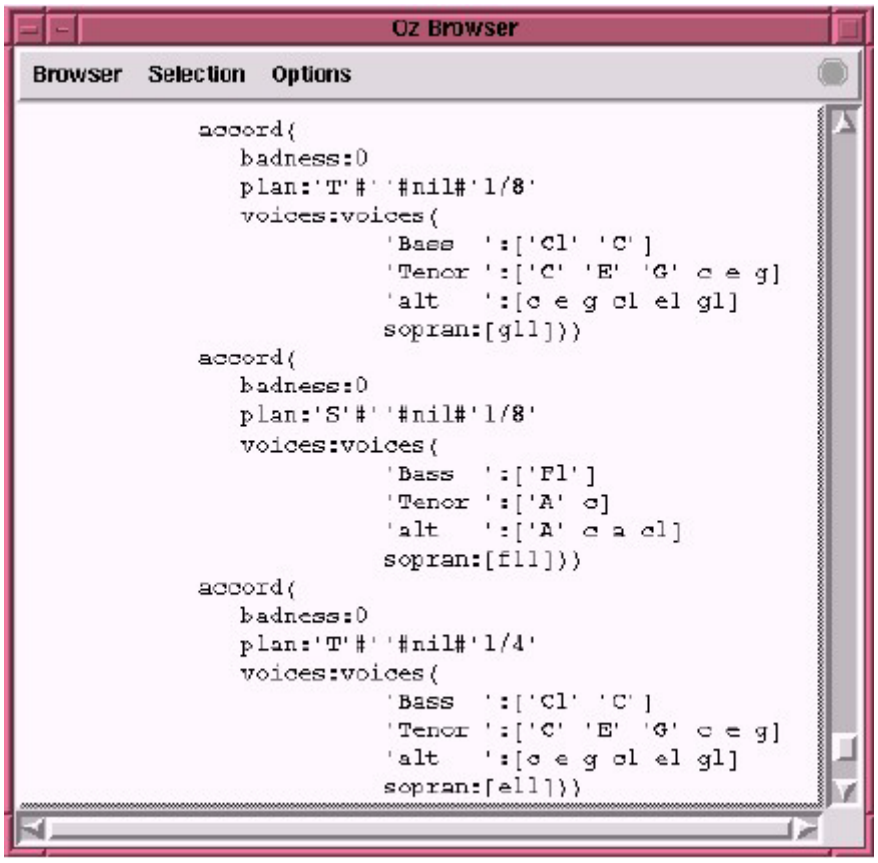


Figura 3: Un fragmento del código de COMPOzE, en el *browser* de Oz

4 CONCLUSIONES

Durante siglos los músicos han utilizado reglas para componer, que en muchos casos constituyen restricciones que permiten navegar y reducir un enorme espacio de búsqueda, conformado por un conjunto finito enormemente grande de valores musicales. Este tipo de problema es similar a los que cotidianamente resolvemos en muchas áreas de la actividad humana, y que han sido estudiados y formalizados por una de las disciplinas de la Inteligencia Artificial: la Programación de Restricciones.

La modelación de la composición musical como un Problema de Satisfacción de Restricciones, surge de manera natural cuando intentamos desarrollar sistemas de composición algorítmica con las herramientas que nos ofrece actualmente la Inteligencia Artificial, lo que se ejemplifica con los variados proyectos de investigación llevados a cabo en los últimos años, no sólo para la composición, sino también para la armonización o el análisis.

En este trabajo hemos comentado la identificación existente entre una actividad artística eminentemente intelectual, la composición musical, y su posible formalización científica con objetivos teórico-prácticos. Hemos ejemplificado con uno de los proyectos desarrollados, el sistema COMOzE, cómo realizar esta formalización, y la resolución automática del problema propuesto, es decir, la creación automática de una sencilla pieza de música.

Ha quedado fuera del alcance del presente texto, la discusión de temas filosóficos relativos a la necesidad o no de crear sistemas automáticos de composición musical. Sin embargo, consideramos conveniente afirmar que el estudio de este tipo de problemas puede ser tan importante, para la Inteligencia Artificial, como la solución automática de juegos como el rompecabezas del 15, el cubo de Rubik o el problema de las *n* reinas.

5 REFERENCIAS BIBLIOGRÁFICAS

1- Carpentier, Alejo. Crónicas, Tomo I. Editorial Arte y Literatura. La Habana, Cuba, 1975.
2- Scholz, Hans. Compendio de Armonía. Editorial Labor S.A., Barcelona, 1951.
3- Barták, Roman. Constraint Programming: In Pursuit of the Holy Grail. http://kti.ms.mff.cuni.cz/~bartak/downloads/WDS99.pdf
4- R. Apt, Krzysztof. Constraint Programming. ERCIM News No.39 - October 1999. http://www.ercim.org/publication/Ercim_News/enw39/apt.html
5- Pachet, F. and Roy, P. 2001. "Musical harmonization with constraints: A survey". *Constraints*, 6(1):7-19. 2001.
6- Ovans, Russell and Rod Davison. An interactive Constraint-Based Expert Assistant for Music Composition. Proceedings of the Ninth Canadian Conference on Artificial Inteligence, pp. 76-81. http://citeseer.nj.nec.com/ovans92interactive.html
7- Henz, Martin, Stefan Lauer, and Detlev Zimmermann. COMPOzE --- Intention-based Music Composition through Constraint Programming. Proceedings of the 8th IEEE International Conference on Tools with Artificial Intelligence, Nov16--19 1996, IEEE Computer Society Press. ftp://ftp.ps.uni-sb.de/pub/papers/ProgrammingSysLab/COMPOzE96.ps.gz
8- Henz, Martin. COMPOzE Intention-based Music Composition through Constraint Programming. http://www.comp.nus.edu.sg/~henz/talks/tai96/