GenoMus: Representing procedural musical structures with an encoded functional grammar optimized for metaprogramming and machine learning

A Preprint

José López-Montes[*]
University of Granada
lopezmontes@correo.ugr.es

October 19, 2019

Abstract

We present GenoMus, a new model for artificial musical creativity based on a procedural approach, able to represent and learn the compositional techniques behind a musical score. The aim of this model is to build a framework for automatic creativity, easily adaptable to other domains beyond music. The core of GenoMus is a functional grammar designed to cover a wide range of styles, integrating traditional and contemporary composing techniques. Musical *genotypes* are defined as functional trees, able to generate musical scores described as *phenotypes*. To enable the maximal diversity of outputs, each process uses the same generic functional structure, no matter what time scale, polyphonic structure or additional characteristics are being employed. The goal of this highly homogeneous and modular approach is to simplify metaprogramming of genotypes, as well as maximize search space. Genotypes and phenotypes are encoded as normalized numeric vectors. This abstract representation of musical knowledge as pure numeric arrays is convenient for the application of different machine learning paradigms. The user interface developed for GenoMus is oriented to the exploration of augmented creativity, regardless of user expertise. However, a composer can create and alter manually genotypes and algorithms to modify automatic results. The system allows the implementation of user-defined processes, which will expand the procedures library.

Keywords: automatic musical composition · metaprogramming · procedural representation of music · artificial creativity · GenoMus

Contents

[*] https://www.lopezmontes.es

1 Introduction

> Edward Fredkin suggested to me the theory that listening to music might exercise some innate map-making mechanism in the brain. When I mentioned the puzzle of music's repetitiousness, he compared it to the way rodents explore new places: first they go one way a little, then back to home. They do it again a few times, then go a little farther. They try small digressions, but frequently return to base. Both people and mice explore new territories that way, making mental maps lest they get lost. Music might portray this building process, or even exercise those very parts of the mind.[1]

I?ll grant that the carliest and simplest programs were little more than simple lists and loops of commands like ?Do thzs. Do that. Do thas and that and thzs agazn until that happens ? That made it hard to imagine how more could emerge from such programs than their programmers envisioned. But t!here?s a big difference bct,ween ?impossible? and ?hard to imagine.? The first is about at; the second is about *you*![2]

A generation later, we should be experimenting on programs that write better programs to replace themselves. Then at last it will be clear how foolish was our first idea-that never, by their nature, could machines create new things.[2]

I don?t believe there?s any substantial difference between ordinary thought and creative thought. Then why do we think there?s a difference? I?ll argue that this is really not a ma.tter of what?s in the mind of the artist-but of what?s in the mind of the critic; the less one understands an artist?s mind the more creat,ive seems the work the artist does.[2]

[3] -> interesante para situar la discusion y encuadrar las motivaciones y posibles peligros. Motivations classes: 1. computer programs are written by the composer as an idiosyncratic extension to her own compositional processes; 2. computer programs are written as general tools to aid any composer in the composition of music; 3. theories of a musical style are implemented as computer programs; 4. cognitive theories of the processes

Failures: 1. a failure to specify the precise practical or theoretical aims of research; 2. a failure to adopt an appropriate methodology for achieving the stated aims; 3. a failure to adopt a means of evaluation appropriate for judging the degree to which the aims have been satisfied.

> Lerner appears to believe that transformations that could be carried out by a computer program , such as "mechanical" cut-and-paste, could not possibly generate anything sensible--and that no program could tell sense from nonsense anyway. The implication, so far as theoretical psychology is concerned, is that no computational theory could describe the generation of valuable new ideas, and that only an unanalyzable faculty of "intuition" or "insight" could recognize their value. None of these beliefs is justified.[4]

1.1 Composing composers

Metalevel -> [5]

[6] -> interesante la disgresion filosofica sobre la autoria. De Music composition to music recognition. Dice: Is the same? Yo digo: Can computer solutions get us to new styles, to nwe ways or feeling music?s

[7] -> ACTUAL! Algoritmic Music Composition Based on Artificial Intelligence: A Survey

El survey de Vico [8]

[9] -> survey actual de evolutionary music generation (para inicio)

Research in artificial musical intelligence demand for formalized grammars of musical structures. Besides, a model of creative mind is required to operate these abstractions. Aesthetic criteria are extremely subjective, furthermore the details of every model of automatic composition impose, consciously or not, a limited search space. Delimiting these boundaries and setting evaluation principles can be seen as metacomposition, namely composing composers.

Composers' interest in musical language pervaded the 20th century aesthetics. Transformation and overcoming of well-established methods inherited from Romanticism led to post-tonal music. Linguistic structuralism applied to musical syntax stimulated relativization and consciousness of compositional procedures. Reversing the logic of this analytic knowledge, the methods of serial dodecaphonic music was the first step for the foundations of an inverse creative strategy: synthesize new styles from the predefinition of new rules.

Computer assisted composition enabled far more complex procedures, tedious or unfeasible to explore by hand. Eventually, composers began to use computers not only for analysis and calculation of complex structures, but for the automation of the creative processes themselves. That fact opened the door to a new approach to composition: a metamusical level characterized by modeling the processes within the minds of composers.

[Reflexiones sobre metacomposicion, el concepto de autoria y consideraciones pedagogicas y humanas de fondo.]

[Interes de la musica en el modelado de creatividad artificial - multidimensionalidad de la percepcion y analisis]

[Sobre la necesidad de usar el metanivel de los procedimientos antes que la partitura]

Many approaches to artificial intelligence applied to the automatic composition of music are modeled using scores as its data source...

[Complejidad del diseno de lenguajes de representacion musical en la composicion asistida por ordenador. Cita de algunas aproximaciones analogas.]

1.2 The next step of computer assisted composition

[Repaso rapido de los principales paradigmas de herramientas de CAC para hacer notar como es necesario un metanivel de trabajo]

Citar [10] -> -Pag. 4-5: Buen sumario de los paradigmas -> -5: justifica problema de neural nets con fragmentos largos coherentes

This proposal, beyond the technical details, is a model of augmented creativity from the point of view of the composer. The new paradigm of computation applied to creative tasks is tilting from ordering to the computer "what to do", to saying "what to get", as a starting point to detonate supervised or non-supervised creative processes. So, the goal of GenoMus is to combine a knowledge base of compositional procedures with the maximal freedom of recombination.

1.3 Scope of the framework

Hacer resumen del proyecto completo, referencias a los antecedentes y dejar claro que trata este texto

1.4 Complementary materials to this text

[Specifications. Code, Max Patches, Examples of specimens, Music excerpts]

2 A functional grammar to represent musical procedures

Similaridad con [11] como el referente mas cercano. Mostrar diferencias. Tambien cercano a [12] como lenguaje creado por un compositor para autoexpansion.

[13] -> solo por citar una aproximacion hibirda muy actual, pero poco interesante

[14] > aproximacion similar en su planteamiento, pero muy limitada

[15] -> un ejemplo analogo a la generacion automatica de arboles de funciones

Referencia a compositor de referencia -> [16] t

2.1 Foundations and requirements

[17] -> Conditions for creativity: Knowledge representation is organised in such a way that the number of possible associations is maximised. A flexible knowledge representation scheme. Similarly Boden (1996) says that representation should allow to explore and transform the conceptual space. Tolerate ambiguity in representations. Allow multiple representations in order to avoid the problem of ?functional fixity?. The usefulness of new combinations should be assessable. New combinations need to be elaboratable to find out their consequences. El articulo de Boden es: What is Creativity. In M. Boden, editor, Dimensions of Creativity, pages 75?118. MIT Press, 1996. (citado antes).

[18] -> Quiza recoger su apunte de las conclusiones: falta el long-term approach y el higher-level concept

Just con [19] -> para la generacion automatica de arboles y gramatica (f mas a fondo)

too [20] -> para justificar el uso de arboles de funciones, ya que este articulo habla de parsear lenguaje (musical entre otros) para llegar al arbol original.

In any approach to artificial creativity, the representation system used is a condition that restricts the search space and determines aesthetic biases a priori, either consciously or unconsciously. On the other hand, the design of algorithms to generate music is ultimately an act of composition. With this in mind, our proposal seeks to be as open as possible, so that it can represent multiple styles. However, since the purpose of the project is not to imitate styles, but to create results of great originality and also serve as a tool for the *augmented creativity* of human composers, this grammar integrates contemporary compositional techniques such as recursive processes, controlled randomness, self-similarity, etc.

A system expressive!! Outputs not a score but a musical excerpt with expressive nuances and values. Both are not separated.

[21] -> Propone trabajar con ‘building blocks‘ mas grandes, cercano a frases

[]

[Conveniencia del paradigma de programacion funcional para la metaprogramacion. Antecedentes procedimientos compositivos como funciones (referencias a Haskell y LISP en la tradicion)]

[Necesidad de encontrar un medio de representacion adecuado a la automatizacion de analisis, a la flexibilidad de estilos, y a conjugar la programacion manual con la modularidad necesaria para las tareas automatizadas]

[Importancia de la codificacion como vector numerico como representacion abstracta de estructuras y resultados musicales para la aplicacion de tecnicas de IA.]

2.2 Musical genotypes and phenotypes

para aclarar que no hay consenso entre el uso de la nomenclatura de genotipos y fenotipos [22]

[23] -> usa el paradigma genotipo fenotipo, aunque de un modo diferente Muy importante: Indirect encoding: IAMUS -> importancia de la evo-devo incremental. El proceso importa, no solo la codificacion del ADN ?Using an effective indirect encoding, a small genotype can potentially specify a large and complex phenotype, accounting for the scalability problem previously mentioned. Additionally, a small change in the genotype can potentially provoke a variety of coordinated changes in the phenotype?

?They are currently being used to a certain extent for automating tasks that demand creativity, proposing different variations to existing solutions, which evolve toward desired design targets, resembling an automated form of brainstorming.?

[24] -> In order to meet this challenge, many researchers are proposing indirect encodings, that is, evolutionary mechanisms where the same genes are used multiple times in the process of building a phenotype.

[Marco conceptual basico del paradigma genotipo-fenotipo (referencias de otros proyectos)]

[Definiciones estrictas de genotipo y fenotipo]

[Similitud con la programacion funcional: la pieza musical como funcion de funciones.]

2.3 Species and specimens

2.4 Anatomy of a genotype function

2.5 Function types

[Tabla con los tipos de funciones. Figura ilustrando la estructura score/voice/chord]

2.6 Leaves, parameter mapping and readability

2.7 Function libraries

2.8 Specimen data structure

[Tabla con la estructura de datos de cada par genofeno generado]

3 Encoded genotypes and phenotypes

- Proposito de la codificacion en el marco del machine learning
- Codificacion como vectores unidimensionales normalizados
- Universal valid format
- Modularidad y posibilidad de manipulacion manual
- Encoding as safety filter

3.1 Encoding genotypes with Gödelian resonances

[25] -> para las resonancias godelianas

Any random sequence of numbers generates a valid program.

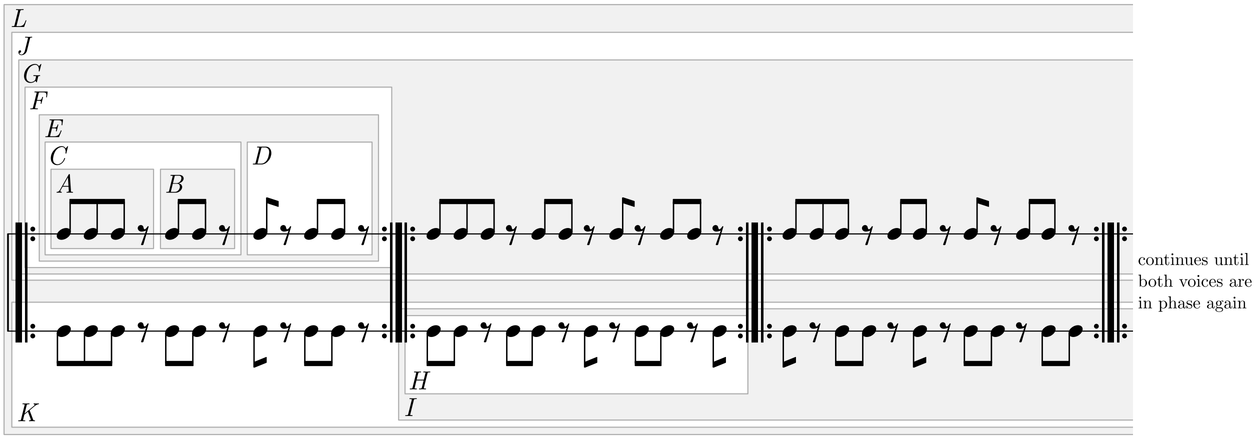

Figure 1: Structures of S. Reich's *Clapping Music.*

$$P = \{x_n \in \mathbb{R} \mid 0 \leq x_n \leq 1\}$$

Multidimensional search space of all possible vectors with normalized parameters:

$$P^n$$

Each vector in P is mapped to a valid encoded genotype (that is a valid functional expression which generates an excerpt of music. Let G be the set of all possible valid encoded genotypes. To create genotypes, the map f works as a decision tree such as every vector $\mathbf{p} = (p_1, p_2, ..., p_n) \in P^n$ will be mapped to G:

$$f: \; P^n \rightarrow G$$

Most of times n doesn't match the number of required items to complete a valid encoded genotype. If $\mathbf{p}$ has more items than needed, they are ignored. If f needs more items than those supplied by $\mathbf{p}$, f reads $\mathbf{p}$ repeatedly from the beginning as a loop, until closing the valid encoded genotype. That implies that even vectors with a single value can be mapped to large functional expressions.

3.2 Scores represented as normalized vectors

3.3 A unique format for procedures and results

3.4 Safety through encoding

4 A minimal example: *Clapping music*

Presentacion de la pieza

Figure 1 shows a boat.

4.1 Code and interface for the experiments

[manual coding to extract procedural and extendable knowledge][11]

4.2 Decoded genotype: a procedural function tree of the piece

```
s2V(                          // score L: joins the 2 voices vertically
  vSlice(                     // voice J: slices last cycle because of phase lag
    vRepeatV(                 // phase G: F 13 times
      vRepeatV(               // cycle F: E 8 times
        vConcatV(             // pattern E: C + D
          vConcatV(           // motif C: A + B
            vMotifLoop(       // core motif A: 3 8th-notes and a silence
```

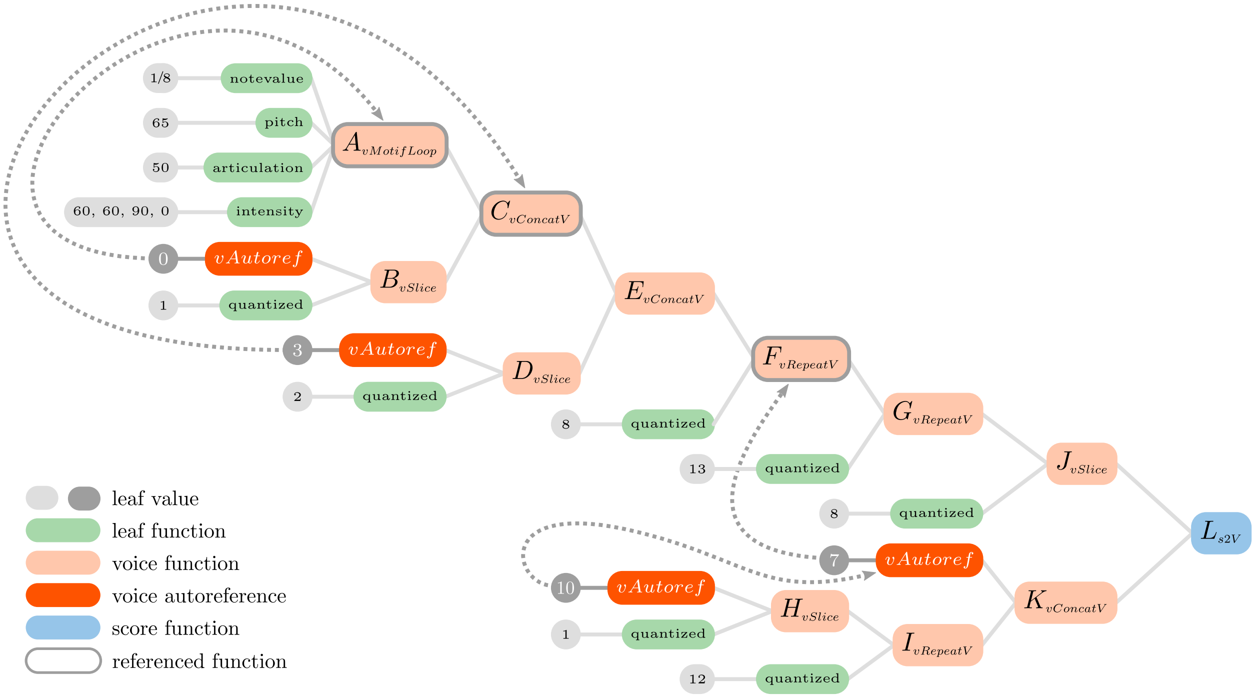

Figure 2: Structures of S. Reich's *Clapping Music*.

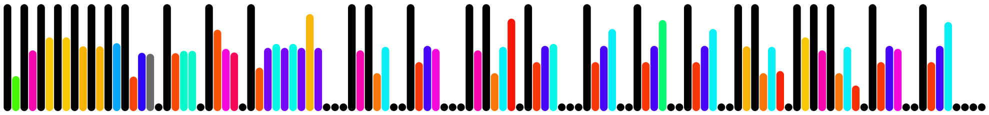

Figure 3: Structures of S. Reich's *Clapping Music*.

```
            ln(1/8),         // note values
            lm(65),          // pitch (irrelevant for this piece)
            la(50),          // articulation
            li(60,60,90,0)), // intensities (last note louder for clarity)
          vSlice(            // motif B: A with 1st note sliced
            vAutoref(0),
            q(1))),
        vSlice(              // motif D: C with 1st note sliced
          vAutoref(3),
          q(2))),
      q(8)),
    q(13)),
  q(8)),
vConcatV(                    // voice K: F + H
  vAutoref(7),
  vRepeatV(                  // phase I: H 12 times
    vSlice(                  // cycle H: cycle F with 1st note sliced
      vAutoref(10),
      q(1)),
    q(12))))
```

Listing 1: A possible decoded genotype for S. Reich's *Clapping Music*

Figure 2 shows a boat.

Figure 3 Visualization of the encoded genotype of *Clapping Music*.

Extrapolacion de estructuras y variantes:

```
ClappingMusic ([.04,.04,.08],[65],[.5],[50,50,120],1,3,2,8,13,8,1,12) =
```

4.3 Encoded genotype: an abstract representation of compositional processes

```
[1, 0.275535, 1, 0.534808, 1, 0.665631, 1, 0.665631, 1, 0.575462, 1, 0.575462, 1,
    0.606798, 1, 0.27051, 0.51, 0.5, 0, 1, 0.506578, 0.53, 0.53, 0, 1, 0.742646,
    0.55, 0.51346, 0, 1, 0.36068, 0.56, 0.6, 0.56, 0.6, 0.56, 0.9, 0.56, 0, 0, 0,
    1, 0.534808, 1, 0.304952, 0.57, 0, 0, 1, 0.416408, 0.58, 0.55, 0, 0, 0, 1,
    0.534808, 1, 0.304952, 0.57, 0.854102, 0, 1, 0.416408, 0.58, 0.6, 0, 0, 0, 1,
    0.416408, 0.58, 0.75, 0, 0, 1, 0.416408, 0.58, 0.84, 0, 0, 1, 0.416408, 0.58,
    0.75, 0, 0, 1, 0.575462, 1, 0.304952, 0.57, 0.326238, 0, 1, 0.665631, 1,
    0.534808, 1, 0.304952, 0.57, 0.18034, 0, 1, 0.416408, 0.58, 0.55, 0, 0, 1,
    0.416408, 0.58, 0.82, 0, 0, 0, 0]
```

Listing 2: Encoded genotype for S. Reich's *Clapping Music*

4.4 Encoded phenotype and possible outputs

- Un ejemplo clasico con varias voces y conteniendo armonia, dinamica y articulacion
- Modularidad y posibilidad de manipulacion manual
- Ejemplos basicos de tecnicas habituales en CAC (movimiento browniano,
- Handling of recursive techniques (fibonacci, y extension del modelo a expresiones matematicas complejas)
- Puentes entra la notacion tradicional, la sintesis de sonido y la espacializacion
- Multimedia

5 Evaluation and evolution

5.1 Flowchart

[26] para el problema del ?fitness bottleneck? con la eval. humana

[27] -> IMPORTANTE distincion entre Genetic Programming and Gen. Algorithms: 'An overview of earlier studies in EC for musical composition is offered in [12], determining that Ge- netic Programming (GP) methods perform better than those that use Genetic Algorithms (GA). This may be unsurprising as GP methods use a tree-based structure whereas GAs are limited to a linear string in their representation. Hence, GP can represent more complex representations and operations | some- thing that would be very useful in representing music.'

[28] -> The evolution of a population offers so much scope and possibility that it is reminiscent of the music creation process a solution is not linearly determined but instead emerges from a uid, incremental process.

[2] - > interesantes perspectivas de la evaluacion, relacionadas con los prejuicios apuntados por Minsky

Citar [29] para -> Ejemplo de sistema basado en programacion textual, para introduccion manual de expresiones complejas -> A menudo son sistemas creados por los propios compositores como medio de extender su estilo

Buscar donde meter el asunto de la hibridacion Otro ejemplo de hibrido -> [13]

El texto de [30] segun Mantaras, usan fitness basado en contorno, intervalica, y otras cuestiones cuantificables

When modeling artistic creativity with algorithms, probably the most evasive issue to address is programming fitness functions. By definition, the assessment of a piece of art can only be made from a subjective point of view, since the goal of art is to provoke inner and personal reactions. These individual responses are very dependent on cultural and social context. However, provided enough data some predictions can be made about the expected rating for a new piece.

In GenoMus, we divide evaluation of each specimen in two categories:

- Autoanalytic profile: objective analysis of a set of musical features, such as variability, rhythmic complelxity, tonal stability, global dissonance index, level of inner autoreference, etc.

- Human evaluations: subjective ratings made by human users, attending to aesthetic value, originality, mood and emotional intensity. This informations are stored individually and together as global statistics.

The self-analysis contained in every specimen allows to measure distance and similaritiy to other specimens, as well as to classify results and to drive evolution processes.

Defining how to evaluate and select results is now the most creative effort, and can be identified with the act of composition itself, since composing music is ultimately making choices.

- Esta propuesta de gramatica posibilita la implementacion y competencia de diferentes sistemas de evaluacion
- Design of evaluation methods as the crucial act of composition
- Objective vs. subjective evaluation
- What to learn?
- Evolutionary paradigm as the most promising

5.2 Evolutionary paradigm

Determining how to evolve and mutate an specimen towards a best version is crucial question too. Starting from a simple motif, endless evolution paths can lead to satisfying results based on heuristic approaches, using accumulated knowledge from examples, human ratings and automatic self-analysis.

The GenoMus grammar is designed to favor the broadest diversity of combinations and transformations. Genetic algorithms are suitable for the automation of an incremental exploration and selection of multiple ways. A GenoMus decoded genotype tree expression can be trasnformed using these methods:

- createGen
- mutateLeaves
- growTrunk
- growBranch
- insertBranch
- flattenBranch
- pruneBranch
- splitGen

Approaches based on neural networks need a very controlled format of data and big training datasets. The encoded genotype format of GenoMus can represent any piece of music as a simple unidimensional sequence of normalized floats, which can be profitable for techniques as recurrent neural networks (ref. to LMSTD), able to learn patterns from sequential streams of data.

5.3 Scalability

- Como conjugar universalidad de las expresiones con optimizacion para tener los vectores codificados con mayores diferencias entre si.
- Estrategias de caracterizacion de perfiles estilisticos
- El problema del mapeo de funciones y su extensibilidad
- Como establecer una base de datos de conocimiento
- Metricas automatizadas de ciertos resultados

5.4 Integrating traditional and contemporary techniques

[31] -> para afirmar que GenoMus reune diferentes paradigmas en sus arboles multicapa. Estos paradigmas segun el articulo son: Analytic, Transformational and Generative

[A genotype como un arbol multiagente, que puede incorporar nodos con funciones de todo tipo: analiticos, recursivos, de constraints, etc. Una vez se tiene un marco de funcion, todo puede caber en el arbol de procesos.]

Justificar con [32]: -> 2: En los 50 era logico excliur la parte expresiva. Los modelos Markovianos son de resultados muy pobres. Ahora debe estar incluida? IMPORTANTE: -> referencia a Minsky, que plantea la posibilidad de los multiagentes, que puedan actuar sobre bloques mayores de musica, y que sean a veces solo analiticos -> ?the rules dont make the music, it is the music that makes the rules? -> Truly creative! para el final

6 Conclusions and ongoing work

[33] -> concluye que los avances prometedores vendran de la integracion de sistemas diferentes

De la referencia citada al principio [7]: ?Symbolic AI methods still have not enough rules therefore, they hardcoded in limited database knowledge. This could be extended with machine learning techniques but the lack of an automatic evaluation method makes it difficult to solve.?

The artistic results of every algorithm designed for automated composition are strongly constrained by their own representation system of musical data. This paper presents GenoMus, a framework for the exploration of artificial musical creativity based on a generative grammar focused on the abstraction of creative processes as a metalevel of compositional tasks. We define musical genotypes as functional nested expressions, and phenotypes as the pieces created by evaluating these computable expressions. GenoMus' grammar is designed to ease the combination of fundamental procedures behind very different styles, ranging from basic to complex contemporary techniques, particularly those able to produce rich output from very simple recursive algorithms. At the same time, maximal modularity is provided to simplify metaprogramming routines to generate, assess, transform and categorize the selected musical excerpts. The system is conceived to maintain a long term interrelation with different users achieving individual musical styles. This proposed grammar can also be an analytic tool, from the point of view of composition as computation, considering that the best analysis of a piece is the shortest precise description.

Cuestiones interesantes:

- ?Cuantas funciones primitivas son necesarias para generar musica en un determinado estilo? Hay innumerables expresiones funcionales diferentes que pueden generar la misma musica. Se puede deducir que la expresion funcional mas breve es el mejor analisis. Se pueden ver diferentes paradigmas de ensenanza/aprendizaje de la musica con estos modelos.
- ?Como puede hacerse ingenieria inversa automatizada para extraer estructuras desde la musica?

[34] -> PARA FINAL!

References

[1] Marvin Minsky. Machine models of music. chapter Music, Mind, and Meaning, pages 327–354. MIT Press, Cambridge, MA, USA, 1992.

[2] Marvin Minsky. Why people think computers can't. *AI Magazine*, 3(4):3–15, 1982.

[3] Marcus Pearce, David Meredith, and Geraint Wiggins. Motivations and methodologies for automation of the compositional process. *Musicae Scientiae*, 6(2):119–147, September 2002.

[4] Margaret A. Boden. *Dimensions of Creativity*, chapter What Is Creativity?, pages 75–118. The MIT Press, 1996.

[5] Bruce G. Buchanan. Creativity at the Metalevel (AAAI-2000 Presidential Address). *AI Magazine*, 22(3):13–28, 2001.

[6] Bruce L. Jacob. Algorithmic composition as a model of creativity. *Organised Sound*, 1(3):157–165, December 1996.

[7] Omar Lopez-Rincon, Oleg Starostenko, and Gerardo Ayala-San Martin. Algoritmic music composition based on artificial intelligence: A survey. In *2018 International Conference on Electronics, Communications and Computers*. IEEE, 2018.

[8] Jose David Fernández Rodriguez and Francisco J. Vico. AI methods in algorithmic composition: A comprehensive survey. *CoRR*, abs/1402.0585, 2014.

[9] Martin Dostál. Evolutionary music composition. In *Handbook of Optimization*, pages 935–964. Springer Berlin Heidelberg, 2013.

[10] Gerhard Nierhaus. *Algorithmic Composition: Paradigms of Automated Music Generation.* Springer Publishing Company, Incorporated, 1st edition, 2008.

[11] David M. Hofmann. A genetic programming approach to generating musical compositions. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pages 89–100. Springer International Publishing, 2015.

[12] Christopher Ariza. *An Open Design for Computer-Aided Algorithmic Music Composition: athenaCL.* Dissertation.Com, 2005.

[13] Robert Crawford. *Algorithmic Music Composition: A Hybrid Approach.* Northern Kentucky University, 2015.

[14] Alfonso Ortega de la Puente, Rafael Sánchez Alfonso, and Manuel Alfonseca Moreno. Automatic composition of music by means of grammatical evolution. In *Proceedings of the 2002 conference on APL array processing languages: lore, problems, and applications - APL '02.* ACM Press, 2002.

[15] P. Laine and M. Kuuskankare. Genetic algorithms in musical style oriented generation. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence.* IEEE, 1994.

[16] I. Xenakis. *Formalized music:.* Indiana University Press, 1971.

[17] Jon Rowe and Derek Partridge. Creativity: a survey of AI approaches. *Artif. Intell. Rev.*, 7(1):43–70, 1993.

[18] Dorien Herremans, Ching-Hua Chuan, and Elaine Chew. A functional taxonomy of music generation systems. *ACM Comput. Surv.*, 50(5):69:1–69:30, September 2017.

[19] Frank Drewes and Johanna Högberg. An algebra for tree-based music generation. In *Proc. 2nd Intl. Conf. on Algebraic Informatics, Lecture Notes in Computer Science. This issue*, 2007.

[20] Rens Bod. The data-oriented parsing approach: Theory and application. In John Fulcher John Fulcher and Lakhmi C. Jain, editors, *Computational Intelligence: A Compendium*, pages 330–342. Springer-Verlag Berlin Heidelberg, 2008.

[21] Bruce L. Jacob. Composing with genetic algorithms. In *Proceedings of the 1995 International Computer Music Conference, ICMC 1995, Banff, AB, Canada, September 3-7, 1995*, 1995.

[22] Csaba Sulyok, Christopher Harte, and Zalán Bodó. On the impact of domain-specific knowledge in evolutionary music composition. In *Proceedings of the Genetic and Evolutionary Computation Conference on GECCO'19.* ACM Press, 2019.

[23] Carlos Sánchez Quintana, Francisco Moreno Arcas, David Albarracín Molina, José David Fernández Rodriguez, and Francisco J. Vico. Melomics: A case-study of AI in spain. *AI Magazine*, 34(3):99, September 2013.

[24] Kenneth O. Stanley and Risto Miikkulainen. A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130, 2003.

[25] Carlos de Lemos Almada. Gödel-vector and gödel-address as tools for genealogical determination of genetically-produced musical variants. In *Computational Music Science*, pages 9–16. Springer International Publishing, 2017.

[26] John Biles. GenJam: A genetic algorithm for generating jazz solos. In *Proceedings of the 1994 International Computer Music Conference, IGMA, San Francisco*, pages 131–137, 1994.

[27] Anthony R. Burton and Tanya Vladimirova. Generation of musical sequences with genetic techniques. *Computer Music Journal*, 23(4):59–73, December 1999.

[28] Róisín Loughran and Michael O'Neill. Generative music evaluation: Why do we limit to 'human'? 2016.

[29] Anthony Richard Burton. *A Hybrid Neuro-Genetic Pattern Evolution System Applied to Musical Composition.* PhD thesis, University of Surrey, 1998.

[30] George Papadopoulos and Geraint Wiggins. A genetic algorithm for the generation of jazz melodies. In *Proceedings of STeP 98*, pages 7–9, 1998.

[31] Rene Wooller, Andrew R Brown, Eduardo Miranda, Joachim Diederich, and Rodney Berry. A framework for comparison of process in algorithmic music systems. In Burraston David and Edmonds Ernest, editors, *Generative Arts Practice*, pages 109–124, Sydney, Australia, 2005. Creativity and Cognition Studios.

[32] Ramón Lopez de Mantaras. Making music with ai: Some examples. In *Proceedings of the 2006 Conference on Rob Milne: A Tribute to a Pioneering AI Scientist, Entrepreneur and Mountaineer*, pages 90–100, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.

[33] George Papadopoulos and Geraint Wiggins. Ai methods for algorithmic composition: A survey, a critical view and future prospects. In *AISB Symposium on Musical Creativity*, pages 110–117, 1999.

[34] Stephen Jay Gould. Creating the creators. *Discover Magazine*, October 1996.