P_{BACH} and Musical Transformations

Carlos de Lemos Almada *
Federal University of Rio de Janeiro (UFRJ)
carlosalmada@musica.ufrj.br
Orcid: 0000-0001-5888-2868

Abstract: *Departing from a specific passage of a text by Douglas Hofstadter which addresses recursive algorithmic application, this paper reformulates its basic structure, by using formal tools under a transformational-musical perspective. The last section of the study proposes some discussion about possible expansions and generalization from the obtained results.*

Keywords*: Recursion. Ordered Sets. Operations and Functions. Transformational Theory.*

I. Introduction

In the dialogue that opens chapter VI (entitled "Canon by Intervallic Augmentation") of Douglas Hofstadter's well-acclaimed, Pulitzer-prized *Gödel, Escher, and Bach* ([1], pp. 153–157), the character Tortoise tries to convince his friend Achilles that two different songs played in his phonograph can be "coded inside the same record": the first "based on the famous old tune B-A-C-H", and a "totally different melody (...) C-A-G-E". This is the kern of Tortoise's argumentation:

> **Tortoise:** (...) What do you get if you list the successive intervals in the melody B-A-C-H?
> **Achilles:** Let me see. First it goes down one semitone, from B to A (where B is taken the German way); then it rises three semitones to C; and finally it falls one semitone, to H. That yields the pattern: $-1, +3, -1$.
> **Tortoise:** Precisely. What about C-A-G-E, now?
> **Achilles:** Well, in this case, it begins by falling three semitones, then rises ten semitones (nearly an octave), and finally falls three more semitones. That means the pattern is: $-3, +10, -3$.
> (...)
> **Tortoise:** They have exactly the same "skeleton", in a certain sense. You can make C-A-G-E out of B-A-C-H by multiplying all the intervals by 3⅓, and taking the nearest whole number.
> (...)
> **Tortoise:** The melody consisted of enormously wide intervals, and went B-C-A-H. (...) It can be gotten from the CAGE pattern by yet another multiplication by 3⅓, and rounding to whole numbers. (...)

*I am very grateful to Dr. Douglas Hofstadter for reading this paper, for his kind words about it, and for give me his personal permission for using an excerpt of his book.

Hofstadter's algorithm, described by Tortoise, could be succinctly expressed as follows:

1. Translate the sequence "BACH" into pitches (adopting the most compact disposition), using German musical notation. Let us name as w the pitch sequence. Thus,
   w = $<B\flat_4, A_4, C_5, B\natural_4>$[1] ;

2. Extract the melodic intervals from w, adopting as unity the semitone. Use the minus signal to indicate descending intervals, and the plus signal to ascending intervals. Let **INT** be the function used to determine intervals between sequential pitches. Let variable x represent the sequence of intervals. Thus,
   x = **INT**(w) = $< -1, +3, -1 >$;

3. Multiply any element of x by 3⅓, and approximate the result in case of fractioned number. Name y this product. Thus,
   y = x × 3⅓ = $< -1, +3, -1 > \times$ 3⅓ = $< -3.33..., +9.99..., -3.33... > \approx < -3, +10, -3 >$;

4. Apply step 3 to y, and let z represent the resulted sequence. Thus,
   z = y × 3⅓ = $< -3, +10, -3 > \times$ 3⅓ = $< -9.99..., +33.33..., -9.99... > \approx < -10, +33, -10 >$;

Figure 1 presents the pitch structure of referential "melody" BACH and of the two recursive transformations.

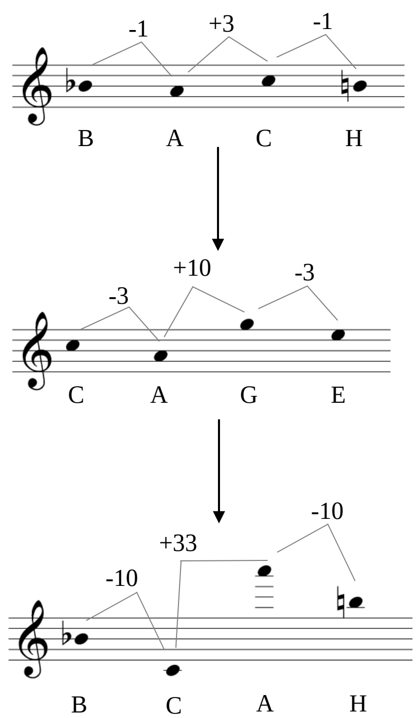

Figure 1: *Transformation of "melodies", according to Hofstadter's algorithm: BACH into CAGE, and CAGE into BCAH.*

[1] Angled brackets indicate that their content is ordered.

I was deeply impressed by this BACH/CAGE dialogue since the first time I read it (as well as the whole book, of course), but only recently, involved with a research based on Transformational and Group theories,[2] I started to conjecture if I could reformulate Hofstadter's clever idea using pitch classes instead of intervals, and if it would be possible to formalize more strictly the transformations.
This brief article was born as an attempt in this direction. The next section defines sets, functions, and operations needed to the transformations, which are implemented in section III, reaching the central aim of the article. An additional section explores the results obtained, attempting to propose some expansion and generalization.

II. Definitions

(1) Let X be a set formed by German symbolic representation for musical notes (upper-case letters or group of upper-case/lower-case letters).[3]

X = {C, Cis, D, Es, E, F, Fis, G, As, A, B, H}

(2) Let Y be a set formed by the twelve pitch classes, or else, Y is isomorphic to $\mathbb{Z}_{12}$.

Y = {0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}

(3) Let f be a bijective function that maps same-order members of X onto Y | f: X $\rightarrow$ Y.

(4) Let f^{-1} be the function inverse of f, that maps same-order members of Y onto X | f^{-1}: Y $\rightarrow$ X.

Figure 2 presents sets X and Y, exemplifying the actions of functions f and f^{-1}.

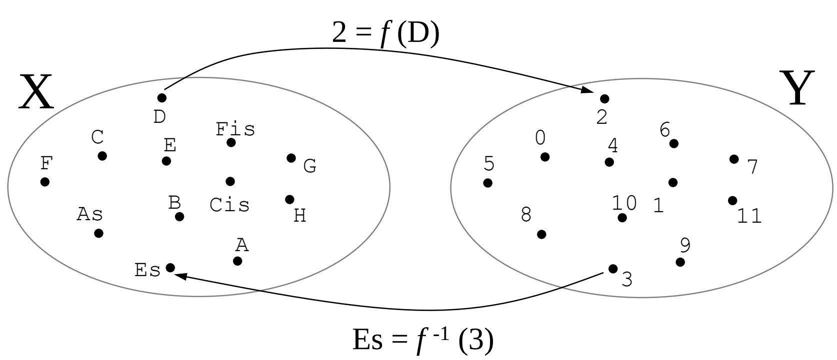

Figure 2: *Sets X and Y and two examples of application of functions f and f^{-1}.*

(5) Let g be the polynomial quadratic function g(x) = $x^2 + 3$ | g: $\mathbb{Z} \rightarrow \mathbb{Z}$.

(6) The operation *retrogradation*, labelled as **R**, flips leftwards the content of a given ordered set S. Ex.: Let set S = <i, j, k, ..., m, n>, then **R**(S) = <n, m, ..., k, j, i>.

[2] For some of my references, see, for example, [2], [3], [4], [5], [6], and [7].
[3] Suffixes "is" and "s" stand for, respectively, the accidents "sharp" and "flat".

(7) The operation *rotation*, labelled as $\mathbf{ROT}_t$, permutes t times (t is an integer greater than zero) the content of a given ordered set S, i.e., it sends at each application the first member of S to the last position of S, keeping unaltered the order of the remaining members.
Ex.: Let set S = <i, j, k, ..., m, n>, then $\mathbf{ROT}_2$(S) = **ROT**(**ROT**(S)) = **ROT**(<j, k, ..., m, n, i>) =
= <k,..., m, n, i, j>.

(8) The operation *extraction*, labelled as $\mathbf{EXT}_{t:u}$, extracts from a given ordered set S a subset formed by contiguous members, delimited by the $t^{th}$and the u^{th} members of S.
Ex.: Let set S = <i, j, k, ..., m, n>, then $\mathbf{EXT}_{2:3}$(S) = T = <j, k>.

(9) The operation *merging*, labelled as **MRG**(S, T), concatenates two ordered sets S and T, keeping unaltered their internal order, forming an ordered superset ST = <S, T>.
Ex.: Let sets S = <i, j, k, ..., m, n> and T = <o, p, q>, then **MRG**(S, T) = ST = <i, j, k, ..., m, n, o, p, q>.

(10) The operation *modulo12*, labelled as $\mathbf{MOD}_{12}$, maps members of set $\mathbb{Z}$ onto members set Y | $\mathbf{MOD}_{12}$: $\mathbb{Z} \rightarrow \mathbb{Z}_{12}$.
Ex.: Let set S = {2, 25, 13, 72, 0, 375}, then $\mathbf{MOD}_{12}$(S) = {2, 1, 1, 0, 0, 3}.

III. Transforming BACH into CAGE

Given the sets, functions, and operations previously defined, this section describes a sequence of nine transformations to be recursively applied. The initial input is a referential set (a_0), representing "BACH", whose output becomes the input of another transformation, with the process being then replicated until the target-set (a_9), namely the "CAGE" string, is reached.

- Let a_0 be an subset of set X, in the following specific order:
  a_0 = <B, A, C, H>
- First transformation: $a_1 = f(a_0)$ = {10, 9, 0, 11};
  Figure 3 provides a graphical representation of this transformation. It is possible to consider not only the individual mappings of the four members of a_0, but also the higher-level action of f on the whole set, in a kind of "holistic" [4] transformation (indicated by the blue arrow).

[4] In reference to a frequent adjective in Hofstadter's book.

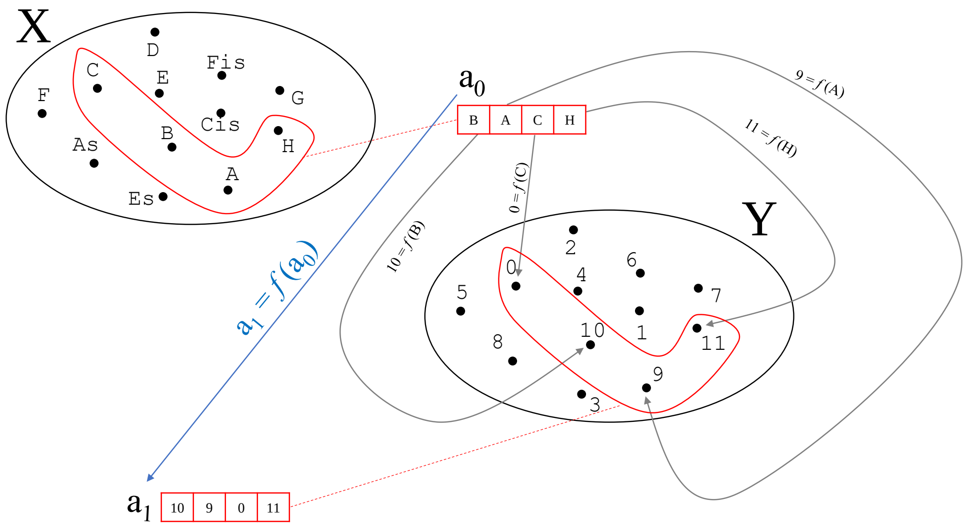

Figure 3: *Representation of the transformation of a_0 into a_1.*

- Second transformation (Figure 4): a_2 = $\mathbf{R}(a_1)$ = <11, 0, 9, 10>;

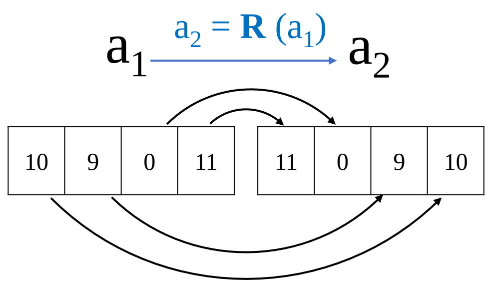

Figure 4: *Representation of the transformation of a_1 into a_2.*

- Third transformation (Figure 5): a_3 = $\mathbf{ROT}_1(a_2)$ = <0, 9, 10, 11>;

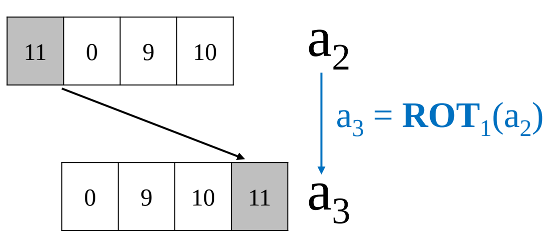

Figure 5: *Representation of the transformation of a_2 into a_3.*

- Fourth transformation (Figure 6): a_4 = $\mathbf{EXT}_{3:4}(a_3)$ = <10, 11>;

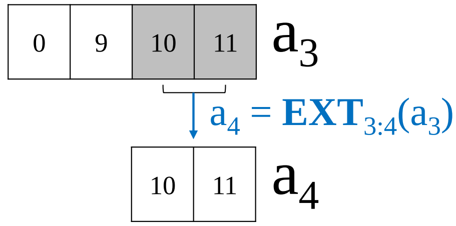

Figure 6: *Representation of the transformation of a_3 into a_4.*

- Fifth transformation (Figure 7): $a_5 = g(a_4) = <(10^2 + 3), (11^2 + 3)> = <103, 124>$;

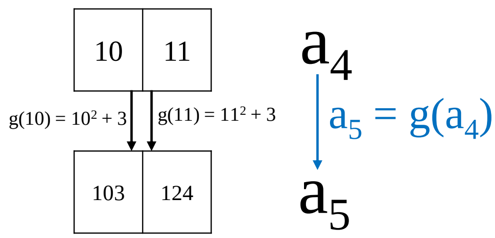

Figure 7: *Representation of the transformation of a_4 into a_5.*

- Sixth transformation (Figure 8): $a_6 = \mathbf{MOD}_{12}(a_5) = <\mathbf{MOD}_{12}(103), \mathbf{MOD}_{12}(124)> = <7, 4>$;

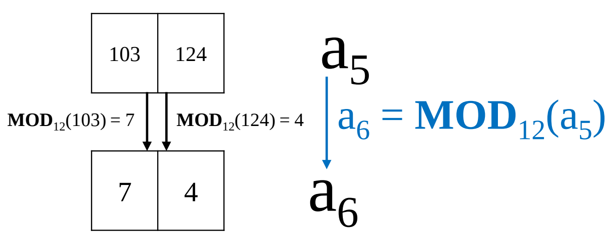

Figure 8: *Representation of the transformation of a_5 into a_6.*

- Seventh transformation (Figure 9): $a_7 = \mathbf{EXT}_{1:2}(a_3) = <0, 9>$;

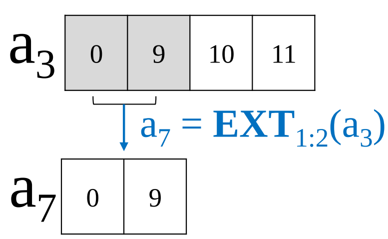

Figure 9: *Representation of the transformation of a_6 into a_7.*

- Eighth transformation (Figure 10): a_8 = **MRG**(a7, a6) = <0, 9, 7, 4>;

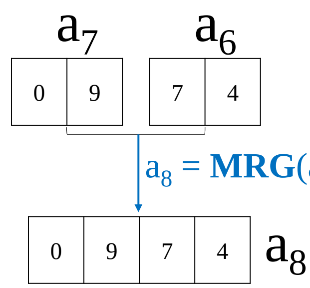

Figure 10: *Representation of the transformation of a_7 into a_8.*

- Ninth transformation (Figure 11): $a_9 = f^{-1}(a_8) = <f^{-1}(0), f^{-1}(9), f^{-1}(7), f^{-1}(4)> = <C, A, G, E>$;

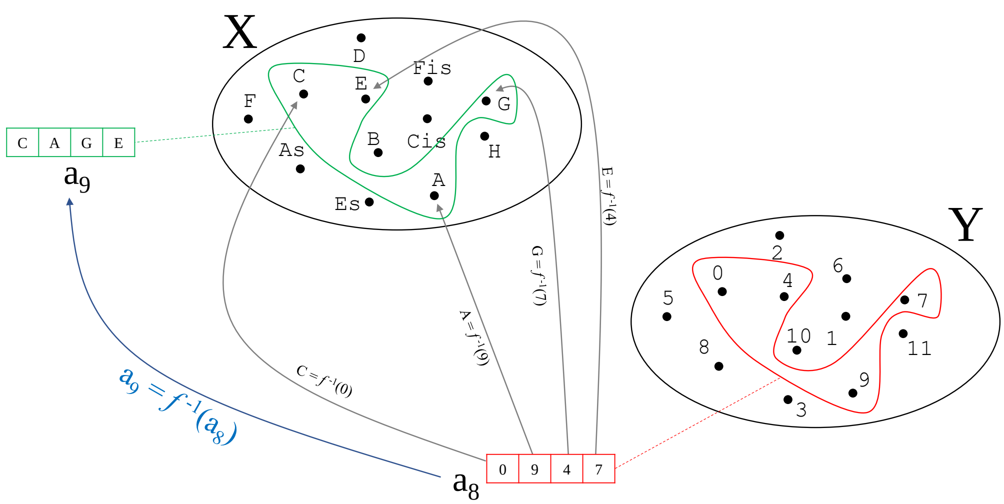

Figure 11: *Representation of the transformation of a_8 into a_9.*

Finally, Figure 12 summarizes the whole process with the aid of a *oriented transformational network*.[5] Under the same holistic perspective applied to the previous cases, I propose the creation of a high-level operation (called **B2C**)[6] that manages to map directly sequence BACH onto CAGE, bypassing the intermediary functions and operations.

[5]Concept coined by David Lewin ([2]). For a comprehensive typology of these networks, see [3], pp. 101-121)
[6]The label stands for *Bach-to-Cage*.

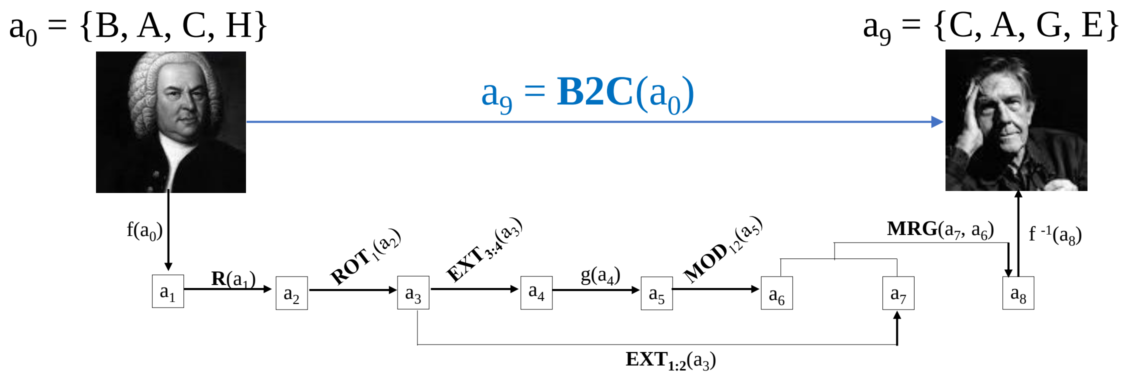

Figure 12: *Network of the nine transformations, including high-level operation **B2C**, that maps a_0 directly onto a_9.*

IV. Going a little further

After reaching the goal aimed by the article, namely, the formalization of the transformational process of the motive/sequence BACH into CAGE, some speculation and questions can arise. For example, what about the *individual* outputs of operation **B2C**? That is, in reverse to what has been done so far, could we implement a low-level function in such a manner that the elements of BACH could be sent to the corresponding members of CAGE? Or yet, it would be possible with this method, after reaching CAGE, turn back to BACH through recursive application of the same transformation (just as Hofstadter managed in his ”prove”)? If affirmative, could we generalize this function and use it to transform ”melodies” of any combination of notes in any possible extension?

Aiming to investigate these possibilities, I propose initially, for simplicity, to work with a subset of X (which ultimately represents the chromatic scale), and to adopt Guido d'Arezzo's hexachord (*Ut-Re-Mi-Fa-Sol-La*, or C-D-E-F-G-A), extended by the ”molle” and ”dur” versions of *Si* (B and H, in German notation).[7] Let us label this new set X' (Figure 13).

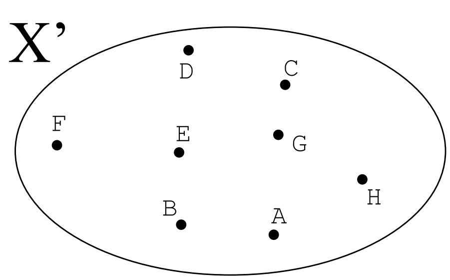

Figure 13: *Subset X'.*

Now, let *b2c* be the low-level function that maps members of X' onto members of itself | b2c: X' → X'

[7]Since this task deals with permutation, if maintained the original cardinality of 12 elements in the set, the number of possible arrangements of the letters would equal 479,001,600.

From the general action of operation **B2C** it is possible to deduce the individual ″behavior″ of BACH's notes (B→C, A→A, C→G, and H→E), but what about the remaining four letters that form subset X' (D, E, F, and G)? Strictly speaking, there are 4^8 (65,536) possible solutions for this problem, but with the aid of logic, and by keeping in mind the idea of recursion (i.e., CAGE returning to BACH),[8] this number can be dramatically reduced to 2 alternatives.

Table 1 proposes an initial approach for the question. As it can be observed, is not possible to go back directly from CAGE to BACH, since ″C″ is sent to ″G″, which will demand at least a second application of operation b2c.

Table 1: *Outputs obtained from two recursive applications of function b2c to members of subset X′.*

w	x = b2c(w)	y = b2c(x)
B	C	G
A	A	A
C	G	?
H	E	?
G	?	?
E	?	?
D	?	?
F	?	?

Assuming a minimal possible number of iterations for going from BACH to CAGE and back to BACH, it is easy to complete the output list: ″G″ must sent to ″B″ (which in turn goes to ″C″), and ″E″ to ″D″ (or ″F″), that goes to ″H″ (and this returns to ″E″). Selecting ″D″ as output of ″E″, ″F″ must necessarily map to itself, like ″A″.[9] Table 2 depicts the definitive configuration of the cyclic transformations of the X′ members.

Table 2: *Outputs obtained from three recursive applications of function b2c to members of subset X′.*

w	x = b2c(w)	y = b2c(x)	z = b2c(y)
B	**C**	G	**B**
A	**A**	A	**A**
C	**G**	B	**C**
H	**E**	D	**H**
G	B	C	G
E	D	H	E
D	H	E	D
F	F	F	F

I will name P_{BACH} this special permutation in X′. It can be written in cyclic notation, as follows:[10]

$$(BCG)A(HED)F$$

[8]What must not necessarily be done in an immediate transformation.

[9]Or, alternatively, E→F and D→D. I chose arbitrarily the first option.

[10]It would also be possible to consider here the formation of a subgroup of the symmetrical group S_8, defined by set X′ and binary composition b2c, but do not intend to pursue this issue in the present article.

V. Concluding remarks

This paper aimed primarily, and unpretentiously to use Hofstadter's "prove" (certainly provocative, but humorous and extremely imaginative in his discussion about recursive algorithms) as a pretext to examine more deeply some of its effects in musical contexts. The system P_{BACH}, derived from the formalization of the transformational process, opens a promising connection between the notion of recursive transformation and musical variation, one of my current research interests.

Figure 14 presents a musical example, a pseudo-Haydnian theme, which is gradually "distorted" by recursive b2c-transformations of its notes (observe, however, that a third application of b2c leads the melody to its original state).[11] In spite of being a very simple (almost rudimentary, I would say) case of ("cyclic") variation,[12] the very essence of the process, namely, the application of recursive transformations, formalized as algebraic operations or functions, is potentially a powerful theoretical construct to be used both for analysis and composition, as it is being currently pursued in the course of my research.

Figure 14: *Cyclic variations of a melody by recursive application of function b2c.*

References

[1] Hofstadter, D. (1979). *Gödel, Escher, and Bach*: An Eternal Golden Braid. New York: Basic Books.

[2] Lewin, D. (1987). *Generalized Musical Intervals and Transformations*. New Haven: Yale University Press.

[3] Rings, S. (2011). *Transformation and Tonality*. Oxford: Oxford University Press.

[4] Gollin, E. (2000). *Representations of Space and Conceptions of Distance in Transformational Music Theories.* Thesis (PhD in Music). Harvard University.

[5] Hoskinson, D. (2006). *The Grundgestalt and Network Transformations in the Late Choral Works of Anton Webern.* Thesis (PhD in Music). University of Oregon.

[6] Schiano, M. (1992). *Arnold Schoenberg's Grundgestalt and its Influence.* Thesis (PhD in Music). Brandeis University.

[11] Put another way, identity is achieved by applying $b2c^2$ to any member of X' (even "A" and "F"), matching one of the group properties.

[12] The complexity of the system could, of course, be increased if we relaxed the constraint of "minimal number of iterations" in the system, forming new, more extended permutations.

[7] Papadopoulos, H. (2014). Mathematics and Group Theory. *Mathematics*, [math.HO], arXiv: 149757v1, pp. 1-33.