

A GAP TUTORIAL FOR TRANSFORMATIONAL MUSIC THEORY**

ROBERT W. PECK

TRANSLATORS: MEHDI SHAMS^{ORCID}* AND SEYYED ALI MOHAMMADIYEH^{ORCID}

ABSTRACT. GAP (an acronym for Groups, Algorithms, and Programming) is a computational discrete algebra system that can be applied as a practical tool for transformational music theory. In particular, this software offers theorists tools that are not easily accessible elsewhere. This tutorial demonstrates how to use GAP to generate algebraic group structures utilized by music theorists. Additionally, it explores how these structures can be applied to the study of transformational theory using familiar concepts derived from the Klumpenhouwer network and Neo-Riemannian theories.

1. Introduction

GAP (an acronym for Groups, Algorithms, and Programming) is a mathematical software package that has great potential as a tool for researchers and students of transformational music theory. Whereas many computer resources are available to music theorists in the investigation of pitch-class set theory and serial theory—pitch-class set calculators, matrix generators, and the like—few, if any, permit users to construct, analyze, and manipulate algebraic systems themselves. Such techniques, however, are an increasingly large part of the transformation theorist's methodologies. It is often necessary to consider concepts such as the subgroup structure of a group, or to determine a group's

Keywords: computer-assisted research, transformational music theory, group theory, Klumpenhouwer networks, Neo-Riemannian theory

Article Type: Translation Paper.

Communicated by Alireza Abdollahi.

*Corresponding author.

Received: 12-11-2023, Accepted: 16-04-2025, Published Online: 30-06-2025.

** The above abstract has been extracted by the translator from the original article (R. W. Peck, A GAP tutorial for transformational music theory, *Music Theory Online*, **18** no. 2 (2011) pp. 22.).

<http://dx.doi.org/10.22108/msci.2025.139767.1621> .

automorphisms. Whereas one can do such calculations by hand, the speed at which **GAP** performs them can dramatically increase efficiency of time for research. From a pedagogical point of view, **GAP** provides students of transformation theory a resourceful and effective environment for experimentation, and a means of providing preliminary empirical proofs for their results.

GAP was developed by and for mathematicians and other computational scientists. It began in 1986 as a project by four students at *RWTH Aachen University*, in Aachen, North Rhine-Westphalia, Germany: Johannes Meier, Alice Niemeyer, Werner Nickel, and Martin Schönert. Its first public release was in 1988, with version 2.4. Since that time, it has undergone a number of revisions, and has assumed worldwide recognition, (1) although it has not received much attention in musictheoretical research. (2) **GAP** 4.4.12 is the latest version. It is free and is available for Windows, Macintosh, and UNIX operating systems, along with all its documentation, at <https://www.gap-system.org>. An interactive online version is also available via Sage.

2. Main Results

Throughout this tutorial we use certain orthographic (i.e., notational) conventions, which sometimes differ from those commonly used in music theory. In some instances, these conventions are required by **GAP**; at other times, they are not necessary, but are more consistent with standard mathematical conventions. For instance, although it is not required by **GAP**, we represent all operations with lowercase letters: $g := (1, 2, 3)$, $h := (1, 2)$.

This notation differs from the common practice in music theory of using uppercase letters for certain operators, such as T_n and I . For these particular operators, we instead use t^n and i , respectively. Operations also appear in cyclic notation. In the case of g above, the parenthetical expression $(1, 2, 3)$ tells us that, under g , the integer 1 moves to 2, 2 moves to 3, and 3 moves back to 1. The result is a permutation of order 3 (i.e., a 3-cycle). In contrast, under h , 1 moves to 2, and 2 moves directly back to 1 (i.e., a 2-cycle, or involution). Then, the composition of g and h , $gh = (1, 2, 3)(1, 2)$, sends 1 to 2 under g , but then 2 back to 1 under h . Hence, it fixes (or stabilizes) 1. Further, it sends 2 to 3 under g , where it remains under h ; and it sends 3 to 1 under g , and 1 to 2 under h . Overall, it stabilizes 1, sends 2 to 3, and 3 to 2, so $gh = (2, 3)$.

We use multiplicative notation rather than additive notation, even for abelian (i.e., commutative) groups: $g \times h$, or gh (rather than $g + h$). Furthermore, **GAP** uses right-functional orthography (i.e., read from right to left). That is, in the composition gh , perform g first, then h . This practice differs from the left-functional notation commonly used in music theory, where, for instance, the operation $T_n I$ means invert first (I), then transpose by n (T_n).

We represent sets, groups, and other algebraic structures with uppercase letters: $G := \langle g, h \rangle$. Here, the angle brackets signify that G is the group generated by g and h . Additionally, we give functions

that map such structures to one another (morphisms) with Greek letters; but as **GAP** reads only ASCII characters, we spell out these Greek letters: $pi : G^{pi} \rightarrow H$, where $g^{pi} \times h^{pi} = (g \times h)^{pi}$, for all $g, h \in G$.

In **GAP**, we use exponential notation for functions, as in the example above, rather than prefix notation, which is more common in music-theoretical writing. In other words, we write G^{pi} , rather than $pi(G)$. In this case, G^{pi} indicates a function pi that maps G to H , where g^{pi} is the image of g in the mapping, h^{pi} is the image of h , and so on. Such functions may be homomorphisms, as in this example; isomorphisms, which are one-to-one; surjective homomorphisms; or automorphisms, which are isomorphisms of a group to itself.

Right-functional orthography is also consistent with **GAP**'s use of exponential notation for functions. The product of functions $alpha$ and $beta$ (in that order) applied to a structure G receives the exponential notation:

$$\left(G^{alpha}\right)^{beta} = G^{alpha.beta}$$

whereas in prefix notation we would have:

$$beta(alpha(G)) = beta.alpha(G)$$

reading from the right to the left.

In cases where gh indicates a composition of group elements that act on a set S , or on a member x of that set, we once again use exponential notation, such as Sg^h , or xg^h , rather than prefix notation, such as $h(g(S))$ or $h(g(x))$. Finally, because we use multiplicative notation, we also use exponential notation for exponents in the usual sense (powers):

$$g \times g = g^2, \quad g \times g \times g = g^3, \quad \dots$$

Throughout this tutorial, **GAP** commands are given in **red**, and **GAP**'s corresponding output appear in **blue**. All commands are entered at the **GAP** prompt **gap >** (see Figure 1) and end in a semicolon (followed by **[enter]**). **GAP** commands may be typed with or without spaces. In **GAP**, we may define various objects using the standard notation **:=** for definition. These objects include operations:

```
gap > g := (1,2,3);
(1,2,3)
gap > h := (1,2);
(1,2)
```

GAP places the output on the line(s) below the defining commands. In this case, the output confirms the definition by redisplaying the cyclic permutations that g and h represent.



FIGURE 1. The GAP welcome screen

We may similarly define algebraic structures:

```
gap> G := Group(g,h);
Group([ (1,2,3), (1,2) ])
gap> N := Subgroup(G,[g]);
Group([ (1,2,3) ])
```

Again, the output confirms the structure that is being defined: the group generated by g and h in the first case, and the subgroup of G generated by g in the second. Notice, however, that the output alone of the second command does not indicate that N is the subgroup of G generated by g .

We may also call objects from one of GAP's libraries:

```
gap> D_6 := DihedralGroup(6);
<pc group of size 6 with 2 generators>
```

Here, the dihedral group of order 6 is a purely abstract algebraic structure, not a permutation group such as G above. The output confirms the size of the group, tells how many generators GAP uses to build the group, and references a particular type of abstract structure: a “*pcgroup*.” The term “*pc*” in the output does not abbreviate “*pitch-class*”, as appears commonly in music theory. Rather, it stands for “*polycyclic*”, a type of group structure that includes the dihedral groups. Then, the elements of the D are not permutations, such as $(1, 2, 3)$ and $(1, 2)$, but rather compositions of abstract generators f_1 (of order 2) and f_2 (of order 3), as seen in the output of the following command.

We may define morphisms and other mappings:

```
gap> pi := IsomorphismGroups(D_6,G);
[ f1, f2 ] -> [ (2,3), (1,2,3) ]
```

The output for this command tells us that GAP has located an isomorphism from D_6 to G ; otherwise, the output would indicate *fail*. Figure 2 provides a mapping for all six elements in each of the respective groups.

$$\begin{aligned}
 f_1 &\mapsto (2, 3) \\
 f_1 \cdot f_1 &\mapsto (2, 3)(2, 3) = () \\
 f_2 &\mapsto (1, 2, 3) \\
 f_2 \cdot f_2 &\mapsto (1, 2, 3)(1, 2, 3) = (1, 3, 2) \\
 f_1 \cdot f_2 &\mapsto (2, 3)(1, 2, 3) = (1, 2) \\
 f_2 \cdot f_1 &\mapsto (1, 2, 3)(2, 3) = (3, 1)
 \end{aligned}$$

FIGURE 2. Mapping of D_6 to G under π

We may display objects, including those defined previously:

```

gap> g;
(1,2,3)
gap> List(G);
[ (), (1,3,2), (1,2,3), (2,3), (1,3), (1,2) ]

```

Whereas the output to the “ g ,” command above displays a single cyclic permutation, the output for “ $List(G)$ ” presents all six group elements in G . The empty parentheses, $()$, in the output that follows the latter command denotes the identity element of the group—for example, the trivial permutation, which sends each member of the set on which G acts (i.e., the integers 1, 2, and 3) to itself.

Additionally, we may determine products, factors, quotients, etc.:

```

gap> g * h;
(2,3)
gap> g ^ 2;
(1,3,2)
gap> DirectProduct(G, D_6);
<group of size 36 with 4 generators>
gap> G / N;
Group([ f1 ])

```

In GAP, we may perform various functions, arithmetic and otherwise:

```

gap> 1 * 2 * 3;
6
gap> Size(G);
6

```

and run various tests:

```
gap> Size(G) = Factorial(3);
true
gap> g * h = h * g;
false
gap> IsAbelian(G);
false
```

We may enter two or more commands on a single line (each followed by a semicolon), and GAP displays their respective output on adjacent successive lines:

```
gap> H := Subgroup(G, [h]); Size(H);
Group([ (1,2) ])
2
```

Long commands may be broken by *[enter]* before reaching the semicolon. In these instances, GAP continues the command line on a subsequent line with a simple “>” prompt:

```
gap> Size(InnerAutomorphismsAutomorphismGroup( [enter]
> AutomorphismGroup(G)));
1
```

The syntax of the above command is complex, but follows a certain logic. In a series of parentheticals, we are asking GAP first to establish the automorphism group for G (*AutomorphismGroup(G)*); second, to determine which of the automorphisms in this group are inner, hence derive from conjugation by a group element (*InnerAutomorphismsAutomorphismGroup*); and finally, to display how many of these inner automorphisms exist (*Size*). The output tells us that the inner automorphism group for G contains only one member.

We may suppress output for a command that would normally produce it by following the command with two semicolons:

```
gap> List(H);;
```

GAP stores a history of command lines for any one session. Users may recall a previous command by using the $\uparrow$ (up-arrow) key repeatedly as needed, and/or with the $\downarrow$ (down-arrow) key, to scroll through the history. Finally, to end a session, we use:

```
gap> quit;
```

Help for GAP users is available in several ways. First, the GAP website offers a number of online manuals, including tutorials, *FAQ*, examples, and instructions for joining the GAP Forum e-mail distribution

list. Help is also available within a **GAP** session; one may find help on a particular topic by typing a search term preceded by a question mark (again, the semicolon is not used):

gap > ?orbit

Help: several entries match this topic - type ?2 to get match [2]

- [1] Tutorial: Orbit
- [2] Reference: Orbit
- [3] Reference: Orbit Stabilizer Methods for Polycyclic Groups
- [4] Reference: Orbits
- [5] Reference: Orbits!operation/attribute
- [6] Reference: OrbitsDomain
- [7] Reference: OrbitLength
- [8] Reference: OrbitLengths
- [9] Reference: OrbitLengthsDomain
- [10] Reference: OrbitStabilizer
- [11] Reference: OrbitStabilizerAlgorithm
- [12] Reference: OrbitPerms
- [13] Reference: OrbitsPerms
- [14] Reference: OrbitStabChain
- [15] Reference: OrbitPowerMaps
- [16] Reference: OrbitFusions
- [17] Extending: Orbits
- [18] Extending: OrbitsishOperation
- [19] Extending: OrbitishF0
- [20] New Features: OrbitGenerators
- [21] New Features: OrbitGeneratorsInv
- [22] New Features: OrbitGeneratorsOfGroup

At a subsequent **GAP** prompt (which need not be immediate), we may enter the number of the entry we wish to read, following a question mark:

gap > ?2

<Orbit(<G>[, <Omega>], <pnt>, [<gens>, <acts>,] <act>)

The orbit of the point <pnt> is the list of all images of <pnt> under the action.

(Note that the arrangement of points in this list is not defined by the operation.)

The orbit of <pnt> will always contain one element that is **equal** to <pnt>.

however for performance reasons this element is not necessarily **identical** to `<pnt>`, in particular if `<pnt>` is mutable.

```
gap> g := Group((1,3,2),(2,4,3));
gap> Orbit(g,1);
[ 1, 3, 2, 4 ]
gap> Orbit(g,[1,2],OnSets);
[[ 1, 2 ], [ 1, 3 ], [ 1, 4 ], [ 2, 3 ], [ 3, 4 ], [ 2, 4 ] ]
(See Section "Basic Actions" for information about specific actions.)
> Orbits( <G>, <seeds>[, <gens>, <acts>][, <act>] )!
```

```
{operation/attribute}
> Orbits( <xset> ){operation/attribute}
```

--- `<space>` page, `<n>` next line, `<b>` back, `<p>` back line, `<q>` quit --

GAP then prompts for some form of continuation or exit from the manual; for example, *n* or *q*. The `[space]`, *n*, etc., are not followed by a semicolon (nor by `[enter]`). On encountering an error, the output provides a description of the problem, whether it is syntactical:

```
gap> Size(Group(g);
Syntax error: ) expected
Size(Group(g);
^
```

or logical:

```
gap> G/H;
Error, <N> must be a normal subgroup of <G> called from
<function>( <arguments>) called from read-eval-loop
Entering break read-eval-print loop ...
you can 'quit;' to quit to outer loop, or
you can 'return;' to continue
brk>
```

In the latter instance, we are given the opportunity to exit the description by typing

```
brk> quit;
```

at the break prompt, or by continuing the description:

```
brk> return;
Error, <H> must be contained in <G> called from
```

```

oper( super, sub ) called from
Index( G, N ) called from
oper( super, sub ) called from
NaturalHomomorphismByNormalSubgroupNCOrig( G, N ) called from
NaturalHomomorphismByNormalSubgroupNC( G, N ) called from
...
Entering break read-eval-print loop ...
you can 'quit;' to quit to outer loop, or
you can 'return;' to continue
brk>

```

Entering either *brk > return;* return; or *brk > quit;* at this point continues or quits the description accordingly.

Users may also access GAP on the World Wide Web via the mathematics software system Sage: www.sagemath.org. Including GAP, Sage has interfaces for Mathematica, Magma, and several other mathematics applications useful to music theorists. Therefore, in addition to computational discrete algebra, Sage may be used to study other algebras, calculus, number theory, cryptography, numerical computation, combinatorics, graph theory, and the like.

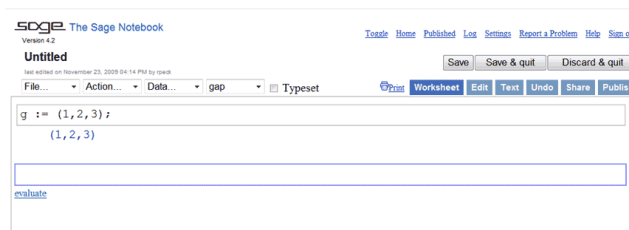


FIGURE 3. A sample Sage Notebook

To work in an online GAP session in Sage, create a notebook (or edit a pre-existing notebook). First, go to www.sagenb.org. Then, either sign in to an existing account, or register for a new one and sign in. For a new notebook, click on *[newworksheet]* and title it. This takes you to a worksheet (see Figure 3). From the drop-down menu near the top, select gap sage is the default). Type a GAP command into the first cell, and click on the corresponding *[evaluate]* link to display its output. A new cell appears, in which you may type the next command, and so on. You may save your worksheet (by clicking the *[save]* button), and publish your worksheet (by clicking the *[publish]* button). The latter gives you a URL, through which others may view your notebook. For example, a notebook *GAP – MTO* which contains all of the commands in this tutorial, may be found at <http://ssagenb.org/pub/1117/>.

Now let us build some familiar music-theoretical groups using **GAP**. In this section, we generate the transposition group **T**, the transposition and inversion group **T/I**, and the affine group **TT0** that includes the transpositions, inversions, and various multiplicative operations. We also examine how **GAP** can model these groups' actions on pitch classes and pitch-class sets, and how they relate to various abstract algebraic structures.

Call **T** the musical transposition group (mathematically, a translation group). **T** has an action on the set **PCS** of twelve pitch classes (residue classes of the infinite set of chromatic pitches under enharmonic and octave equivalence). We may use **GAP** to model groups such as **T**, using a permutation representation on some subset of the positive integers. Such representations in **GAP** do not incorporate the integer 0, to which the pitch class **C** is usually mapped. Instead, we map pitch class **C** to the integer 12. Put

$$\begin{aligned} C\# &= 1 \\ D &= 2 \\ Eb &= 3 \\ &\vdots \\ B &= 11 \\ C &= 12 \end{aligned}$$

Hence, we may define a unit transposition t (translation), a pitch-class transposition “up” one semitone:

```
gap> t := (1,2,3,4,5,6,7,8,9,10,11,12);
(1,2,3,4,5,6,7,8,9,10,11,12)
```

We may compose operators using multiplication, such as t with t (T_1 with T_1):

```
gap> t * t;
(1,3,5,7,9,11)(2,4,6,8,10,12)
```

We may also use exponents for an operator composed with itself, as before:

```
gap> t^2;
(1,3,5,7,9,11)(2,4,6,8,10,12)
```

We may show inverses:

```
gap> t^-1;
(1,12,11,10,9,8,7,6,5,4,3,2)
gap> (t^2)^-1;
(1,11,9,7,5,3)(2,12,10,8,6,4)
```

We can use t as a generator of a group, T :

```
gap> T := Group(t);
Group([ (1,2,3,4,5,6,7,8,9,10,11,12) ])
```

The output for this command confirms that T is the group generated by t , where t is the permutation $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$. T is isomorphic to the cyclic group of order 12, C_{12} :

```
gap> IsomorphismGroups(T, CyclicGroup(12));
[ (1,5,9)(2,6,10)(3,7,11)(4,8,12), (1,4,7,10)(2,5,8,11)(3,6,9,12) ] ->
[ f3, f1*f2 ]
```

GAP can determine the orbit of a pitch class under the members of the T group, shown here using pitch class 1:

```
gap> Orbit(T, 1);
[ 1, 9, 5, 11, 7, 3, 12, 8, 4, 10, 6, 2 ]
```

GAP can also display the orbit of an unordered set of pitch classes (a pcset), such as a C major triad $\{0, 4, 7\}$ (recalling that we use the integer 12 in place of pitch class 0):

```
gap> Orbit(T, [4, 7, 12], OnSets);
[ [ 4, 7, 12 ], [ 4, 8, 11 ], [ 3, 8, 12 ], [ 2, 6, 9 ], [ 1, 6, 10 ],
[ 2, 5, 10 ], [ 1, 5, 8 ], [ 5, 9, 12 ], [ 1, 4, 9 ], [ 3, 7, 10 ],
[ 2, 7, 11 ], [ 3, 6, 11 ] ]
```

One way that we might use GAP as a tool for music theory is in the study of Klumpenhower networks (Lewin [17], Klumpenhower [11]). Klumpenhower networks, or K -nets, are directed graphs with nodes (vertices) labeled in the set of pitch classes and arrows (edges) in the T/I group (see Figure 4):

```
gap> t^2; i*t^3; i^5;
(1,3,5,7,9,11)(2,4,6,8,10,12)
(1,2)(3,12)(4,11)(5,10)(6,9)(7,8)
(1,4)(2,3)(5,12)(6,11)(7,10)(8,9)
```

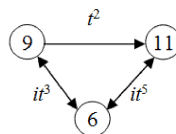


FIGURE 4. A sample K -net

Following Philip Lambert's ([14]) analysis, Figure 5 presents the first six measures of Arnold Schoenberg's Sechs kleine Klavierstücke, op. 19, no. 6.

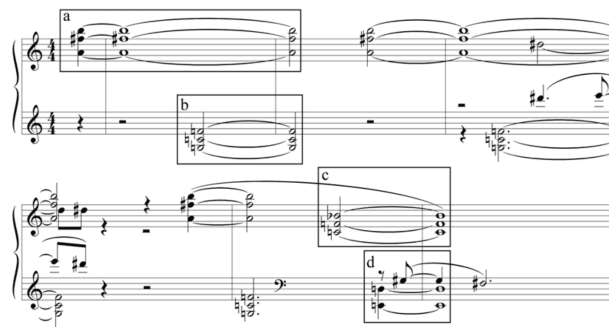


FIGURE 5. Schoenberg, *Sechs kleine Klavierstücke*, op. 19, no. 6, measures 1 – 6 (following Lambert [14])

Another area of music-theoretic research for which **GAP** is useful is neo-Riemannian theory. This theory deals largely with contextual operations on the set of consonant triads, or Klänge. To that end, define an arbitrary consonant (major or minor) triad as an unordered pitch-class set (pcset):

```
gap> CM := [4,7,12];
[ 4, 7, 12 ]
```

Call the orbit of a pitch-class set under T/I a set-class (in this case, the set of pcsets to which a C major triad maps under the elements of the T/I group), and label the set-class of consonant triads K :

```
gap> K := Orbit(TI,CM,OnSets);
[ [ 4, 7, 12 ], [ 1, 5, 8 ], [ 5, 8, 12 ], [ 2, 6, 9 ], [ 4, 7, 11 ],
[ 1, 6, 9 ], [ 3, 7, 10 ], [ 3, 6, 10 ], [ 2, 7, 10 ], [ 3, 6, 11 ],
[ 4, 8, 11 ], [ 2, 5, 9 ], [ 3, 8, 11 ], [ 2, 5, 10 ], [ 5, 9, 12 ],
[ 1, 4, 8 ], [ 4, 9, 12 ], [ 1, 4, 9 ], [ 1, 6, 10 ], [ 3, 7, 12 ],
[ 1, 5, 10 ], [ 3, 8, 12 ], [ 2, 7, 11 ], [ 2, 6, 11 ] ]
```

K , as a set, has twenty-four members (twelve major triads and twelve minor triads, all represented above) on which T/I acts:

```
gap> TI_K := Action(TI,K,OnSets);
Group([ (1,2,4,7,11,15,19,23,22,18,14,10) (3,6,9,13,17,21,24,20,16,12,8,5),
(1,3)(2,5)(4,8)(6,10)(7,12)(9,14)(11,16) (13,18)(15,20)(17,22)(19,24)(21,23) ])
```

We can arrive at an isomorphic structure on $[1 \dots 24]$ that is perhaps more intuitive. Let $[1 \dots 12]$ represent the pitch classes that function as the roots of the major triads, and let $[13 \dots 24]$ represent the roots of the minor triads, where x and $x + 12$ belong to the same pitch class. (That is, 1 and 13 both map to the pitch class $C\#$; 2 and 14 to D ; $\dots$, 12 and 24 to C .) See Table 1.

TABLE 1. Alternate mapping of K to $[1 \cdots 24]$

13	$\mapsto$	C#-	1	$\mapsto$	C#+
14	$\mapsto$	D-	2	$\mapsto$	D+
15	$\mapsto$	Eb-	3	$\mapsto$	Eb+
16	$\mapsto$	E-	4	$\mapsto$	E+
17	$\mapsto$	F-	5	$\mapsto$	F+
18	$\mapsto$	F#-	6	$\mapsto$	F#+
19	$\mapsto$	G-	7	$\mapsto$	G+
20	$\mapsto$	Ab-	8	$\mapsto$	Ab+
21	$\mapsto$	A-	9	$\mapsto$	A+
22	$\mapsto$	Bb-	10	$\mapsto$	Bb+
23	$\mapsto$	B-	11	$\mapsto$	B+
24	$\mapsto$	C-	12	$\mapsto$	C+

3. Conclusions

This tutorial endeavors to introduce some of the utilities in **GAP** to the music-theoretic community in the context of familiar transformation-theoretical concepts: transformation groups and their actions, Klumpenhouwer networks, and neo-Riemannian theory. Therefore, it presupposes a basic—but not advanced—understanding of these concepts, and no prior knowledge of **GAP**. After an initial section that describes the basics of the **GAP** environment (how to input commands, interpret output, get help, etc.), we generate the transposition group **T**, the transposition and inversion group **T/I**, and the affine group **TT0**. We examine how these groups act on the set of pitch classes and on sets of pitch-class sets, and perform various tests that relate them to well-known abstract-algebraic structures. Next, we use **GAP** to study Klumpenhouwer networks, generating the automorphism groups from which K-net isographies derive. Finally, we consider the Schritt and Wechsel group of neo-Riemannian theory, how it relates to the transposition and inversion group, how these are subgroups of Hook’s [9] Uniform (and Quasi-uniform) Triadic Transformations, and how to generate other related groups of contextual operations.

The preceding discussion is intended to demonstrate **GAP**’s usefulness in studying transformational music theory. Whereas most of the examples recreate well known models, it has not been our purpose here to discover novel results. Nevertheless, **GAP** may function quite effectively in the pursuit of new theoretical systems. Moreover, it can be a valuable tool in transformation theory pedagogy, affording teachers and students an environment for experimentation with either existing or original concepts.

Mehdi Shams

Department of Statistics, Faculty of Mathematical Sciences, University of Kashan, Kashan 87317-53153 Kashan, I. R. Iran

Email: `mehdishams@kashanu.ac.ir`

Seyyed Ali Mohammadiyeh

Department of Pure Mathematics, Faculty of Mathematical Sciences, University of Kashan, Kashan 87317-53153 Kashan, I. R. Iran

Email: `alim@kashanu.ac.ir`

خودآموز GAP برای نظریه‌ی موسیقی تبدیلی

روبرت ویلیام پک

مترجمان: مهدی شمس*^{ID} و سیدعلی محمدیه^{ID}

چکیده. GAP (یک واژه‌ی اختصاری برای گروه‌ها، الگوریتم‌ها و برنامه‌نویسی) دستگاهی از جبر گسسته‌ی محاسباتی بوده که ممکن است به‌عنوان یک برنامه‌ی کاربردی برای نظریه‌ی موسیقی تبدیلی به‌کار رود. به‌ویژه، این نرم‌افزار ابزارهایی را برای نظریه‌پردازان موسیقی پیشنهاد می‌دهد که در جاهای دیگر به‌آسانی در دسترس نیستند. این خودآموز، نحوه‌ی استفاده از GAP برای تولید ساختارهای گروه‌های جبری قابل استفاده برای نظریه‌پردازان موسیقی را نشان می‌دهد. به‌علاوه، این خودآموز نحوه‌ی اعمال این ساختارها را در مطالعه‌ی نظریه‌ی تبدیل با استفاده از مفاهیم آشنای برگرفته از شبکه‌ی کلمپن‌هاور و نظریه‌های نئورمانی بررسی خواهد کرد.

۱. درباره‌ی GAP

۱.۱. GAP (یک واژه‌ی اختصاری برای گروه‌ها، الگوریتم‌ها و برنامه‌نویسی) یک بسته‌ی نرم‌افزاری ریاضی است که توانایی بالقوه‌ی بالایی به‌عنوان یک ابزار برای تحقیقات و مطالعات نظریه‌ی موسیقی تبدیلی دارد. درحالی‌که بسیاری از منابع رایانه‌ای برای نظریه‌پردازان موسیقی در بررسی نظریه‌ی مجموعه‌ی نغمه‌های هم‌نام و نظریه‌ی ترتیبی (ماشین حساب‌های مجموعه‌ی نغمه‌های هم‌نام، مولدهای ماتریسی و غیره) در دسترس هستند، اما تعداد بسیار کمی از آن‌ها به کاربران اجازه می‌دهند تا دستگاه‌های جبری را تولید و تحلیل و پردازش کنند. با این حال، چنین شیوه‌هایی به‌طور فزاینده، بخش بزرگی از روش‌های بررسی نظریه‌پرداز تبدیل را شامل می‌شود. به عنوان مثال، معمولاً نیاز است مفاهیمی همچون ساختار زیرگروه یک گروه یا تعیین خودریختی‌های یک گروه مورد بررسی قرار گیرند. با وجود این‌که یک فرد می‌تواند چنین محاسباتی را با دست انجام دهد، اما سرعت اجرا در GAP، موجب افزایش قابل توجه بهره‌وری از زمان برای تحقیقات می‌شود. از دیدگاه آموزشی، GAP محیط پرمحتوا و کارآمدی را برای آزمایش‌ها و همچنین ابزارهایی را برای ارائه اثبات‌های تجربی مقدماتی دانشجویان نظریه‌ی تبدیل فراهم می‌کند.

عبارات و کلمات کلیدی: پژوهش به کمک رایانه، نظریه‌ی موسیقی تبدیلی، نظریه‌ی گروه، شبکه‌های کلمپن‌هاور، نظریه‌ی نئورمانی.

نوع مقاله: ترجمه‌ای

دبیرتخصصی رابط: علیرضا عبدالحی

نویسنده مسئول

تاریخ دریافت: ۱۴۰۲/۰۸/۲۱ تاریخ پذیرش: ۱۴۰۴/۰۱/۲۷ تاریخ انتشار آنلاین: ۱۴۰۴/۰۴/۰۹

ارجاع به مقاله: ر. و. پک، مترجمان: م. شمس، سید ع. محمدیه، خودآموز GAP برای نظریه‌ی موسیقی تبدیلی، نشریه ریاضی و جامعه، ۱۰ شماره ۲ (۱۴۰۴) ۱۱۵-۱۶۲.

<http://dx.doi.org/10.22108/msci.2025.139767.1621>

** نوشته حاضر ترجمه مقاله زیر است:

R. W. Peck, A GAP tutorial for transformational music theory, music theory online, 17 no. 1 (2011) pp. 22.

¹Groups, Algorithms, and Programming

۲.۱. GAP به وسیله ریاضیدانان و سایر دانشمندان محاسباتی توسعه داده شد. این کار در سال ۱۹۸۶ به عنوان یک تحقیق به وسیله چهار دانشجو در دانشگاه آخن RWTH^۲، آغاز شد (جونز مایر^۳، آلیس نیمایر^۴، ورنر نیکل^۵ و مارتین شونرت^۶). این تحقیق ابتدا در سال ۱۹۸۸ به صورت عمومی در نسخه ۲.۴ منتشر شد. از آن زمان تاکنون، چندین اصلاح روی این نسخه انجام شد و در سرتاسر جهان به رسمیت شناخته شد^۷، ولی در تحقیقات موسیقی نظری مورد توجه قرار نگرفت^۸. آخرین نسخه ۴.۴.۱۲ GAP است^۹، این نسخه رایگان است و برای دستگاه‌های عملیاتی ویندوز، مکینتاش، و یونیکس به همراه تمام مستندات آن در تارنمای <https://www.gap-system.org> در دسترس است. یک نسخه آنلاین فعال نیز از طریق Sage در دسترس است (۱۸.۲ را ببینید).

۳.۱. این خودآموز برای معرفی برخی از امکانات GAP و مفاهیم تبدیل نظری، به منظور کاربرد برای جامعه علمی موسیقی استفاده می‌شود که شامل گروه‌های تبدیل و عمل‌های آنها، شبکه‌های کلمپین‌هاور و نظریه نوریمانی است. بنابراین، این خودآموز، یک پیش‌فرض فهم پایه‌ای و اولیه (نه پیشرفته) ارائه می‌کند و به دانش قبلی درباره‌ی GAP نیازی ندارد. پس از بخش اولیه‌ای که پایه‌های محیط GAP را توصیف می‌کند (نحوه ورود فرمان‌ها، تفسیر خروجی‌ها، راهنما و غیره)، گروه ترانهش T ، گروه ترانهش و انعکاس T/I و گروه آفین TTO را تولید می‌کنیم. در این مقاله، عمل این گروه‌ها روی مجموعه نغمه‌های هم‌نام بررسی شده و آزمون‌های مختلفی اجرا می‌شوند که آن‌ها را به ساختارهای جبر مجرد شناخته شده برای ریاضی‌دانان ارتباط می‌دهند. سپس، از GAP برای مطالعه شبکه‌های کلمپین‌هاور استفاده می‌کنیم که گروه‌های خودریختی حاصل از برابرنگاری‌های K -شبکه را تولید می‌کند. در نهایت، گروه شریث^{۱۰} و وچسل^{۱۱} نظریه نوریمانی، نحوه ارتباط آن به گروه ترانهش و انعکاس، دلیل تشکیل زیرگروه‌های تبدیلات سه‌تایی یکنواخت (شبه یکنواخت) هوک [۹] و نحوه تولید سایر گروه‌های مربوط به عمل‌های وابسته را در نظر می‌گیریم.

۲. قراردادهای نوشتاری و فرمان‌های GAP

۱.۲. در سرتاسر این خودآموز از قراردادهای نوشتاری معینی (نمادین) استفاده می‌کنیم که گاهی اوقات با قراردادهای مرسوم در نظریه موسیقی متفاوت هستند. در برخی جاها، این قراردادها مورد نیاز GAP هستند و در برخی جاها دیگر، الزامی و ضروری نبوده، بلکه با قراردادهای ریاضی استاندارد سازگارتر هستند. برای نمونه، با وجود این‌که GAP نیازی به این قرارداد ندارد، اما تمام عمل‌ها را با حروف کوچک نمایش می‌دهیم، مثلاً

$$g := (1, 2, 3)$$

یا

$$h := (1, 2)$$

این نمادگذاری با رویه رایج در نظریه موسیقی فرق دارد؛ در نظریه موسیقی از حروف بزرگ برای عملگرهای معین مانند T_n و I استفاده می‌شود. در GAP به جای این عملگرهای خاص، به ترتیب از t^n و i استفاده می‌کنیم (بخش ۳ را ببینید). عمل‌ها در نمادگذاری دوری نیز استفاده می‌شوند. عبارت داخل پرانتز $(1, 2, 3)$ در حالت g که در بالا اشاره شد، بیان می‌کند

^۷ یک کتاب‌شناسی ناکامل شامل هزاران اثر منتشر شده برای استاندارد به GAP، در تارنمای (وبسایت) <https://www.gap-system.org/Doc/Bib/gap-published.html> موجود است.

^۸ یک مثال از کاربرد GAP در موضوع موسیقی نظری در پک [۲۹] ارائه شده است.

^۹ آخرین نسخه اشاره شده در مقاله مربوط به سال ۲۰۰۸ است و در ژوئن ۲۰۲۴ جدیدترین نسخه‌ی GAP، یعنی نسخه ۴.۱۳.۱ منتشر شده است، که محققین علاقه‌مند می‌توانند برای کسب اطلاعات بیشتر به تارنمای <https://www.gap-system.org/Releases> مراجعه کنند: مترجم

^۲RWTH Aachen University ^۳Johannes Meier ^۴Alice Niemeyer ^۵Werner Nickel ^۶Martin Schönert ^{۱۰}Schritt ^{۱۱}Wechsel

که تحت g ، عدد صحیح ۱ به ۲، عدد صحیح ۲ به ۳ منتقل می‌شود و عدد ۳ به ۱ رجعت می‌کند. این نتیجه، یک جایگشت مرتبه‌ی ۳ (یعنی، یک ۳-دور) است. در مقابل، تحت h ، ۱ به ۲ می‌رود و ۲ به ۱ برمی‌گردد (یعنی، یک ۲-دور یا پیچ). سپس، ترکیب g و h یعنی $gh = (1, 2, 3)(1, 2)$ ابتدا تحت g ، ۱ را به ۲ می‌نگارد و سپس تحت h ، ۲ به سمت ۱ باز می‌گردد. از این‌رو، این ترکیب، عدد صحیح ۱ را ثابت نگه می‌دارد. به‌علاوه، این ترکیب تحت g ، ۲ را به ۳ منتقل می‌کند که تحت h ثابت می‌ماند و تحت g ، ۳ را به ۱ و تحت h ، ۱ را به ۲ می‌نگارد. به‌طور کلی، این ترکیب عدد ۱ را ثابت نگه می‌دارد، ۲ را به ۳ و ۳ را به ۲ می‌نگارد، بنابراین، $gh = (2, 3)$.^{۱۲}

۲.۲. از نماد ضربی به‌جای نماد جمعی استفاده می‌کنیم؛ حتی برای گروه‌های آبلی (تعویض‌پذیر یا جابه‌جایی):

$$g \times h \text{ یا } gh$$

$$(g + h \text{ به‌جای } g)$$

علاوه‌بر این، GAP از نوشتار راست-تابعی (یعنی، از راست به چپ خوانده شود) استفاده می‌کند. یعنی، در ترکیب gh ، ابتدا g و سپس h را اجرا می‌کند. این عمل با نماد چپ-تابعی مرسوم در نظریه‌ی موسیقی فرق دارد که در آن برای نمونه، عمل $T_n I$ بدین معنی است که ابتدا عمل (I) یعنی معکوس لحاظ می‌شود، سپس به‌وسیله‌ی عمل $n(T_n)$ ترانهاده می‌شود.

۳.۲. مجموعه‌ها، گروه‌ها و سایر ساختارها را با حروف بزرگ نمایش می‌دهیم،

$$G := \langle g, h \rangle$$

که در اینجا، قلاب زاویه‌ای بدین معنی است که G گروهی است که به‌وسیله‌ی g و h تولید شده است. به‌علاوه، توابعی را ارائه می‌دهیم که چنین ساختارهایی را با استفاده از حروف یونانی، به ساختارهای دیگر (ریختی‌ها) می‌نگارد؛ اما هنگامی که GAP تنها کاراکترهای ASCII را می‌خواند، این حروف یونانی را توضیح می‌دهیم، یعنی

$$pi : G^{pi} \rightarrow H \text{ داریم } g^{pi} \times h^{pi} = (g \times h)^{pi},$$

$$(g, h \in G \text{ همه برای همه})$$

همانند مثال فوق، در GAP به‌جای نمادگذاری پیشوندی که در نوشتن موسیقی نظری متداول‌تر است، از نمادگذاری نمایی برای نشان دادن توابع استفاده می‌کنیم. به عبارت دیگر، به‌جای $pi(G)$ ، می‌نویسیم G^{pi} . در این حالت، G^{pi} تابع pi را نشان می‌دهد که از G به H نگاشته می‌شود و در آن g^{pi} تصویر g در نگاشت است، h^{pi} تصویر h است و به همین ترتیب. چنین توابعی ممکن است همریختی (مانند این مثال)، یک‌ریختی (همریختی یک‌به‌یک)، بروریختی (همریختی پوشا) یا خودریختی (یک‌ریختی یک گروه به خودش) باشند.

۴.۲. نوشتار راست-تابعی با کاربرد نمادگذاری نمایی GAP برای توابع نیز سازگار است. ضرب توابع $alpha$ و $beta$ (در آن مرتبه) که بر ساختار G اعمال شده به نمادگذاری نمایی

$$(G^{alpha})^{beta} = G^{alpha.beta}$$

منجر می‌شود. درحالی‌که در نمادگذاری پیشوندی، رابطه به صورت

$$beta(alpha(G)) = beta.alpha(G)$$

^{۱۲} خواننده‌ی علاقه‌مند برای اطلاعات بیشتر، می‌تواند به دیکسون و مورتیمر [۶] مراجعه کند.

خواهد بود (از راست به چپ بخوانید).

۵.۲. در حالت‌هایی که gh ، ترکیبی از عناصر گروه را نشان می‌دهد و بر مجموعه‌ی S عمل می‌کند، اولین بار از نمادگذاری نمایی (مانند، S^{gh} یا x^{gh}) به جای نمادگذاری پیشوندی (مانند، $h(g(S))$ یا $h(g(x))$) استفاده می‌کنیم. در نهایت، از آنجایی که از نمادگذاری ضربی استفاده می‌کنیم، از نمادگذاری نمایی نیز برای توان‌ها در مفهوم عادی (توان‌ها) استفاده می‌کنیم:

$$g \times g = g^2, g \times g \times g = g^3, \dots$$

۶.۲. فرمان‌های GAP در سرتاسر این خودآموز با رنگ قرمز و خروجی متناظر GAP با رنگ آبی^{۱۳} نشان داده شده است. توجه کنید GAP، به حروف بزرگ و کوچک حساس است. تمام فرمان‌ها در GAP، با دستور " $gap >$ " وارد می‌شوند (شکل ۱ را ببینید) و با نقطه ویرگول به پایان می‌رسند (بعد از آن کلید ورود $[enter]$ را فشار دهید). فرمان‌های GAP ممکن است با فضا یا بدون فضای خالی نوشته شوند. در GAP توسط نمادگذاری استاندارد " $:=$ "، اشیاء مختلف تعریف می‌شوند. این اشیاء شامل عمل‌ها نیز هستند:

```
gap > g := (1,2,3);
(1,2,3)
gap > h := (1,2);
(1,2)
```

GAP، خروجی را در خط (خطوط) در زیر تعاریف فرمان‌ها قرار دارد. در این حالت، خروجی نمایش جایگشت‌های دوری g و h در بخش ۱.۲ را به‌طور مجدد در بخش ۷.۲ مورد بررسی قرار می‌دهیم.



شکل ۱. صفحه خوش آمدگویی GAP

Figure 1: The GAP welcome screen

^{۱۳}GAP از این رنگ برای کدنویسی استفاده نمی‌کند؛ در اینجا برای وضوح ارائه استفاده می‌شود.

۷.۲. ممکن است ساختارهای جبری را به صورت مشابه تعریف کنیم:

```
gap > G := Group(g, h);  
Group([ (1,2,3), (1,2) ])  
gap > N := Subgroup(G, [g]);  
Group([ (1,2,3) ])
```

دوباره خروجی، ساختار تعریف شده را می‌پذیرد: در سطر اول و دوم، گروه تولیدشده به وسیله g و h و در سطر سوم و چهارم، زیرگروه G تولیدشده به وسیله g معرفی می‌شوند. با این حال، توجه داشته باشید که تنها خروجی فرمان دوم (سطر چهارم)، نشان می‌دهد که N زیرگروه G هست که توسط g تولید شده است.

۸.۲. همچنین، ممکن است اشیاء حاصل از یکی از کتابخانه‌های GAP را فراخوانی کنیم:

```
gap > D_6 := DihedralGroup(6);  
<pc group of size 6 with 2 generators>
```

در اینجا، گروه دو وجهی مرتبه‌ی ۶ یک ساختار در جبر مجرد است، اما یک گروه جایگشتی مانند گروه G که در بخش ۷.۲ مورد بررسی قرار گرفت، دارای چنین ساختاری نیست. خروجی، اندازه‌ی گروه را می‌پذیرد که نحوه‌ی استفاده از بسیاری از مولدهای GAP را برای ساخت گروه بیان می‌کند و به نوع خاصی از ساختار مجرد “گروه pc” اشاره دارد. عبارت “pc” در خروجی، برخلاف نظریه‌ی موسیقی، واژه‌ی نغمه‌های هم‌نام^{۱۴} را خلاصه نمی‌کند. در عوض، این برای گروه‌های “چنددوری”^{۱۵} که نوعی از ساختارهای گروه که شامل گروه‌های دو وجهی هستند، برقرار است. بنابراین، عناصر D_6 مثل، $(1, 2, 3)$ و $(1, 2)$ جایگشت نیستند، اما همان‌طور که در خروجی فرمان بعدی (در بخش ۹.۲) می‌توان دید، ترکیب‌های مولدهای مجرد f_1 (از مرتبه‌ی ۲) و f_2 (از مرتبه‌ی ۳) جایگشتی هستند.

۹.۲. ممکن است ریختی‌ها و سایر نگاشت‌ها را تعریف کنیم:

```
gap > pi := IsomorphismGroups(D_6, G);  
[ f1, f2 ] -> [ (2,3), (1,2,3) ]
```

خروجی این فرمان بیان می‌کند که GAP، یک یک‌ریختی از D_6 به G را مشخص می‌کند؛ در غیر این صورت، خروجی “fail” را نشان خواهد داد. (ذکر این نکته ضروری است، در حالت‌هایی مانند این مورد که بیش از یک، یک‌ریختی ممکن وجود دارد، GAP فقط یکی از آن‌ها را برای نمایش انتخاب می‌کند). یک‌ریختی pi ، یک نگاشت از هر عنصر D_6 به یک عنصر G است، اما خروجی نرم‌افزار، این یک‌ریختی را تنها بر حسب مولدهای (مجرد) D_6 ، f_1 و f_2 و تصاویر آن‌ها در G نشان می‌دهد که نگاشت‌های عناصر گروه اضافی ممکن است از طریق آن به دست آیند (به عنوان مثال، $h = (1, 2) = (2, 3)(1, 2, 3)$ در G ، تصویری از $f_1.f_2$ در D_6 است). جدول ۱، نگاشتی را برای تمام شش عنصر موجود در هر گروه مربوطه ارائه می‌دهد.

^{۱۴} مخفف Pitch-Class: مترجم.

^{۱۵} در این مقاله pc، مخفف Polycyclic است: مترجم.

جدول ۱. نگاشت D_6 به G تحت π

Table 1: Mapping of D_6 to G under π

f_1	$\mapsto$	$(2, 3)$
$f_1 \cdot f_1$	$\mapsto$	$(2, 3)(2, 3) = ()$
f_2	$\mapsto$	$(1, 2, 3)$
$f_2 \cdot f_2$	$\mapsto$	$(1, 2, 3)(1, 2, 3) = (1, 3, 2)$
$f_1 \cdot f_2$	$\mapsto$	$(2, 3)(1, 2, 3) = (1, 2)$
$f_2 \cdot f_1$	$\mapsto$	$(1, 2, 3)(2, 3) = (3, 1)$

۱۰.۲. ممکن است اشیاء را نمایش دهیم که شامل موارد قبلاً تعریف شده باشند:

```
gap > g;
(1,2,3)
gap > List(G);
[ (), (1,3,2), (1,2,3), (2,3), (1,3), (1,2) ]
```

با وجود این که خروجی فرمان “ g ” در دستور بالا، جایگشت دوری منحصر به فردی را نشان می‌دهد، اما خروجی “ $List(G)$ ” تمام شش عنصر گروه موجود در G را نشان می‌دهد. پرانتزهای خالی () در خروجی حاصل از فرمان اخیر، عنصر همانی گروه را نشان می‌دهد؛ برای مثال، جایگشت بدیهی که هر عضو مجموعه‌های که G روی آن عمل می‌کند (یعنی، اعداد صحیح ۱، ۲ و ۳) را به خود آن عضو می‌نگارد.

۱۱.۲. به علاوه، می‌توانیم ضرب‌ها، ضرایب، خارج قسمت‌ها و غیره را تعیین کنیم:

```
gap > g * h;
(2,3)
gap > g ^ 2;
(1,3,2)
gap > DirectProduct(G, D_6);
<group of size 36 with 4 generators>
gap > G / N;
Group([ f1 ])
```

خروجی‌های دو فرمان اول با توجه به مبحث قبلی سراسر و ساده هستند، اما خروجی‌های فرمان‌های سوم و چهارم به توضیحات بیشتری نیاز دارند. خروجی فرمان سوم بیان می‌کند که ضرب مستقیم، گروهی از مرتبه‌ی ۳۶ (یا ۶۲، هنگامی که دو گروه در ضرب مستقیم دارای اندازه‌ی ۶ هستند) است و چهار مولد (یعنی، مولدهای مربوط به گروه‌های G و D_6 : g و h و f_1 و f_2) دارد. فرمان چهارم، گروه خارج قسمتی را فراخوانی می‌کند که شامل هم‌دسته‌های مختلفی از N در G است. f_1 در خروجی این خط، یکی از آن هم‌دسته‌ها را نشان می‌دهد و به f_1 دیگری (یک مولد از D_6) اشاره ندارد.

۱۲.۲. ممکن است توابع، علوم محاسباتی و روش‌های متنوع دیگری را در GAP پیاده‌سازی کنیم:

```
gap > 1 * 2 * 3;
```

```
6
```

```
gap > Size(G);
```

```
6
```

و آزمون‌های مختلفی را اجرا کنیم:

```
gap > Size(G) = Factorial(3);
```

```
true
```

```
gap > g * h = h * g;
```

```
false
```

```
gap > IsAbelian(G);
```

```
false
```

۱۳.۲. ممکن است دو فرمان یا بیشتر را روی یک خط منحصر به فرد (هر کدام با نقطه ویرگول به پایان می‌رسند) وارد کنیم و GAP خروجی مربوط به آن‌ها را روی خطوط متوالی نمایش دهد:

```
gap > H := Subgroup(G, [h]); Size(H);
```

```
Group([ (1,2) ])
```

```
2
```

فرمان‌های طولانی می‌توانند به وسیله کلید *[enter]* پیش از رسیدن به نقطه ویرگول شکسته شوند. در این نمونه‌ها، GAP شامل خط فرمان روی یک خط متعاقب با علامت ">" ساده است:

```
gap > Size(InnerAutomorphismsAutomorphismGroup( [enter]
```

```
> AutomorphismGroup(G)));
```

```
1
```

ترکیب فرمان بالا پیچیده است، اما منطق مشخصی را نتیجه می‌دهد. در مرحله‌ی اول، از GAP در دنباله‌های دارای پرانتز می‌خواهیم که گروه خودریختی را برای G به دست آورد ($AutomorphismGroup(G)$)؛ سپس می‌خواهیم خوردیختی‌هایی را در این گروه تعیین کند که درونی هستند؛ این خوردیختی‌ها از مزدوج با یک عنصر گروه به دست می‌آیند ($InnerAutomorphismsAutomorphismGroup$)؛ و در نهایت، فرمان ($Size$) برای نمایش تعداد خوردیختی‌های درونی موجود اجرا می‌شود. خروجی بیان می‌کند که گروه خودریختی درونی G تنها شامل یک عضو است.

۱۴.۲. ممکن است خروجی را برای یک فرمان متوقف کنیم که معمولاً در این فرمان با دو نقطه ویرگول ایجاد خواهد شد:

```
gap > List(H);;
```

GAP، تاریخچه‌ی خطوط فرمان را برای هر یک از نشست‌ها ذخیره می‌کند. کاربران می‌توانند فرمان قبلی را در صورت نیاز با استفاده مکرر از کلید ↑ (فلش رو به بالا) و یا با کلید ↓ (فلش رو به پایین) برای پیمایش در سرتاسر تاریخچه فراخوانی کنند. در نهایت، از دستور زیر برای خاتمه این نشست استفاده می‌کنیم:

```
gap > quit;
```

۱۵.۲. به جای این که فرمان های GAP را در هر نشست دوباره بنویسیم، می توانیم برنامه هایی را برای ذخیره سازی این فرمان ها به صورت فایل های متن معمولی بسازیم. بنابراین، برنامه ها می توانند در یک نشست GAP بارگذاری شوند. در این روش، می توانیم یک محیط را به آسانی برای آزمایش پیوسته تنظیم کنیم. برای مثال، تمام فرمان های مورد استفاده در بخش های ۳-۵ این خودآموز در فایل متنی "SampleGAPprogram.txt" ظاهر می شوند. از فرمان زیر برای بارگذاری این برنامه در یک نشست GAP استفاده می کنیم^{۱۶}:

```
gap > Read("SampleGAPprogram.txt");
```

توجه داشته باشید که این فرمان هیچ خروجی ندارد، هرچند فرمان هایی که شامل این برنامه هستند به طور معمولی ایجاد خواهند شد. در نتیجه، برنامه به جز تعاریف، به فرمان های خاصی نیاز ندارد. اولین بار که چنین برنامه ای خوانده می شود، کاربر می تواند فرمان های نمایشی را به سادگی فراخوانی کند، تعاریف اضافی را شرح دهد، آزمون ها را اجرا کند و غیره. همچنین، برنامه ها می توانند شامل تذکری باشند که GAP به عنوان فرمان در نظر نمی گیرد. تذکرها، از قبل با علامت # تعیین می شوند و با نقطه ویرگول به پایان نمی رسند:

```
gap > # This is a remark. It is not a command.
```

لازم به ذکر است که در اینجا، دستور بالا هیچ خروجی ندارد.

۱۶.۲. کلید Help با روش های مختلفی برای کاربران GAP در دسترس است. ابتدا، تارنمای GAP تعدادی از کتابچه های راهنمای آنلاین را پیشنهاد می کند که شامل دوره های آموزشی خودآموز، FAQ، مثال ها و دستورالعمل هایی برای پیوستن به فهرست توزیع رایانامه (ایمیل) انجمن GAP است. کلید Help در داخل یک نشست GAP نیز در دسترس است؛ ممکن است Help را پیرامون موضوع خاصی با نوشتن عبارت جست و جو همراه با علامت سؤال پیدا کنید (دوباره، نقطه ویرگول استفاده نمی شود):

```
gap > ?orbit
```

Help: several entries match this topic - type ?2 to get match [2]

- [1] Tutorial: Orbit
- [2] Reference: Orbit
- [3] Reference: Orbit Stabilizer Methods for Polycyclic Groups
- [4] Reference: Orbits
- [5] Reference: Orbits!operation/attribute
- [6] Reference: OrbitsDomain
- [7] Reference: OrbitLength
- [8] Reference: OrbitLengths
- [9] Reference: OrbitLengthsDomain
- [10] Reference: OrbitStabilizer
- [11] Reference: OrbitStabilizerAlgorithm
- [12] Reference: OrbitPerms
- [13] Reference: OrbitsPerms

^{۱۶} برای این که GAP این فرمان را همان طور که در اینجا ارائه شده بخواند، برنامه باید در فهرست راهنمای مشابهی ذخیره شود که GAP برنامه را از آن شروع می کند. برای مثال، این فهرست راهنما در نصب پیش فرض ویندوز در مسیر C:\gap4r4\bin قرار دارد.

- [14] Reference: OrbitStabChain
- [15] Reference: OrbitPowerMaps
- [16] Reference: OrbitFusions
- [17] Extending: Orbits
- [18] Extending: OrbitsishOperation
- [19] Extending: OrbitishFO
- [20] New Features: OrbitGenerators
- [21] New Features: OrbitGeneratorsInv
- [22] New Features: OrbitGeneratorsOfGroup

ممکن است چندین ورودی را برای خواندن در علامت GAP بعدی (که بلافاصله مورد نیاز نیست) پس از علامت سؤال وارد کنیم:

gap > ?2

`<Orbit( <G>[, <Omega>], <pnt>, [<gens>, <acts>, ] <act>)`

The orbit of the point `<pnt>` is the list of all images of `<pnt>` under the action.

(Note that the arrangement of points in this list is not defined by the operation.)

The orbit of `<pnt>` will always contain one element that is **equal** to `<pnt>`, however for performance reasons this element is not necessarily **identical** to `<pnt>`, in particular if `<pnt>` is mutable.

gap > `g := Group((1,3,2), (2,4,3));;`

gap > `Orbit(g,1);`

`[ 1, 3, 2, 4 ]`

gap > `Orbit(g,[1,2],OnSets);`

`[ [ 1, 2 ], [ 1, 3 ], [ 1, 4 ], [ 2, 3 ], [ 3, 4 ], [ 2, 4 ] ]`

(See Section “Basic Actions” for information about specific actions.)

> `Orbits( <G>, <seeds>[, <gens>, <acts>][, <act>] )!`

{operation/attribute}

> `Orbits( <xset> )!{operation/attribute}`

--- <space> page, <n>next line, back, <p> back line, <q> quit --

بنابراین، GAP برای تداوم چندین شکل یا خروج از کتابچه‌ی راهنما وارد عمل می‌شود؛ برای مثال، *n* یا *q*. کلید `[space]`، *n* و غیره با نقطه ویرگول (و نه با کلید `[enter]`) به پایان نمی‌رسند.

۱۷.۲. خروجی، توصیفی از مسئله را در زمان مواجهه با یک خطا ارائه می‌دهد، خواه نحوی باشد:

```
gap > Size(Group(g);
Syntax error: ) expected
Size(Group(g); ^
```

یا منطقی:

```
gap > G/H;
Error, <N> must be a normal subgroup of <G> called from
<function>( <arguments>) called from read-eval-loop
Entering break read-eval-print loop ...
you can 'quit;' to quit to outer loop, or
you can 'return;' to continue
brk>
```

در نمونه‌ی اخیر، فرصتی را برای خروج از متن توصیفی با نوشتن عبارت

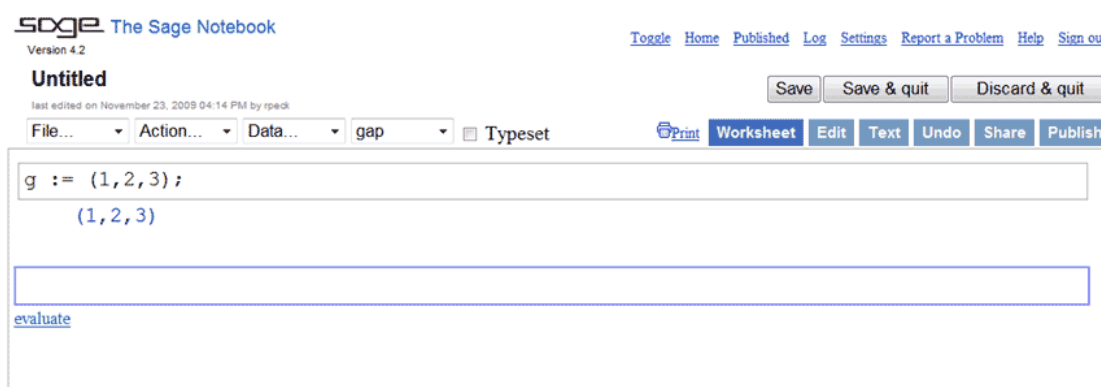
```
brk > quit;
```

فراهم کرده‌ایم یا متن را به صورت زیر ادامه می‌دهیم:

```
brk > return;
Error, <H> must be contained in <G> called from
oper( super, sub ) called from
Index( G, N ) called from
oper( super, sub ) called from
NaturalHomomorphismByNormalSubgroupNCOOrig( G, N ) called from
NaturalHomomorphismByNormalSubgroupNC( G, N ) called from
...
Entering break read-eval-print loop ...
you can 'quit;' to quit to outer loop, or
you can 'return;' to continue
brk>
```

بر این اساس، متن توصیفی با وارد کردن دو دستور `brk > return;` یا `brk > quit;` در این نقطه یا ادامه پیدا می‌کند یا پایان می‌یابد.

۱۸.۲. کاربران ممکن است به GAP اجازه دهند تا به شبکه‌ی جهانی وب از طریق سامانه‌ی نرم‌افزار ریاضی، یعنی Sage با آدرس www.sagemath.org دسترسی داشته باشد. Sage به انضمام GAP دارای گذرگاه‌هایی برای برنامه‌های Magma، Mathematica و چندین برنامه ریاضی دیگر است که برای نظریه‌پردازان موسیقی مفید هستند. بنابراین، Sage ممکن است علاوه بر جبر گسسته‌ی محاسباتی، برای مطالعه‌ی سایر جبرها، حساب دیفرانسیل و انتگرال، نظریه‌ی اعداد، رمزنگاری، محاسبات عددی، ترکیبیات، نظریه‌ی گراف و سایر شاخه‌های نظیر آن استفاده شود.



شکل ۲. نمونه صفحه Sage

Figure 2: A sample Sage Notebook

۱۹.۲. یک دفترچه یادداشت برای کار در یک نشست GAP آنلاین در Sage ایجاد کنید (یا یک دفترچه یادداشت قدیمی را اصلاح کنید). ابتدا، به تارنمای www.sagenb.org بروید. سپس، وارد یک حساب موجود شوید یا حساب جدیدی را ثبت کنید و وارد شوید. روی "new worksheet"، یعنی "کاربرگ جدید" و عنوان آن برای یک دفترچه یادداشت جدید کلیک کنید. این یک کاربرگ جدید را ارائه می‌دهد (شکل ۲ را ببینید). "gap" را از فهرست کشویی نزدیک به بالا انتخاب کنید ("sage" پیش‌فرض است). فرمان GAP را در سلول اول بنویسید و بر پیوند "evaluate"، یعنی "ارزیابی" متناظر برای نمایش خروجی آن کلیک کنید. یک سلول جدید ظاهر می‌شود که می‌توانید فرمان بعدی را در آن بنویسید و به همین ترتیب. می‌توانید کاربرگ خود را ذخیره (با کلیک بر بخش "save"، یعنی "ذخیره") و چاپ کنید (با کلیک بر بخش "publish"، یعنی "چاپ"). حالت اخیر به شما یک URL می‌دهد که از طریق آن می‌توانید کاربرگ خود را در میان سایر کاربرگ‌ها ببینید. برای مثال، کاربرگ "GAP-MTO" ممکن است در <http://ssagenb.org/pub/1117/> یافت شود که شامل تمام فرمان‌های موجود در این خودآموز است.

۳. ساخت گروه‌های T/I ، T و TTO

۱.۳. حال، برخی از گروه‌های موسیقی نظری آشنا را با استفاده از GAP می‌سازیم. در این بخش، گروه ترانهش T ، گروه ترانهش و انعکاس T/I و گروه آفین TTO را به دست می‌آوریم که شامل عمل‌های ترانهش‌ها، معکوس و عمل‌های ضربی مختلفی است. همچنین، نحوه‌ی مدل‌سازی عمل‌های این گروه‌ها روی رده‌ی نغمه‌های هم‌نام و مجموعه‌های نغمه‌های هم‌نام به وسیله‌ی GAP و نحوه‌ی ارتباط آن‌ها با سایر ساختارهای جبری مجرد مختلف را بررسی می‌کنیم.

۲.۳. T را گروه ترانهش موسیقایی (از نظر ریاضی، یک گروه انتقال) می‌نامیم^{۱۷}. T بر مجموعه‌ی PCS نغمه‌های هم‌نام دوازده‌گانه عمل می‌کند (رده‌های باقیمانده مجموعه‌ی نامتناهی از نغمه‌های کروماتیک تحت هم‌ارزی هم‌دانگ و نت‌های هنگام). ممکن است از GAP برای مدل‌سازی گروه‌های نظری T با استفاده از یک نمایش جایگشتی بر مبنای چندین

^{۱۷} ریاضیدانان به عملی که نظریه پردازان موسیقی معمولاً از آن به عنوان "ترانهش" یاد می‌کنند را "انتقال" می‌نامند. از نظر ریاضی، یک انتقال هر نقطه را با فاصله یکسان در همان جهت حرکت می‌دهد، در حالی که ترانهش تنها دو عنصر را در یک مجموعه جابه‌جا می‌کند و تمام عناصر دیگر ناورد را باقی می‌مانند.

زیرمجموعه از اعداد صحیح مثبت استفاده کنیم. چنین نمایش‌هایی در GAP، عدد صحیح صفر که معمولاً نغمه‌ی هم‌نام C برای آن نگاشته می‌شود را در نظر نمی‌گیرند^{۱۸}. در واقع، نغمه‌ی هم‌نام C را به عدد صحیح ۱۲ می‌نگارد.

۳.۳. قرار دهید

$$C\# = ۱$$

$$D = ۲$$

$$Eb = ۳$$

$$\vdots$$

$$B = ۱۱$$

$$C = ۱۲$$

از این‌رو، ممکن است ترانهش واحد t (انتقال) را به‌صورت زیر تعریف کنیم، که یک ترانهش نغمه‌ی هم‌نام تا یک نیم‌پرده است:

$gap > t := (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12);$

$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$

به‌عبارت دیگر، t صرفاً نماد دیگری برای عمل‌هایی است که معمولاً به‌صورت T_1 با استفاده از حروف کوچک برای نام متغیر نوشته می‌شود. خروجی نشان می‌دهد، t یک نماد دوری است: تحت t ، نغمه‌ی هم‌نام ۱ به ۲، ۲ به ۳ و به‌همین ترتیب، ۱۱ به ۱۲ نگاشته می‌شود و ۱۲ به ۱ برمی‌گردد.

۴.۳. ممکن است عملگرها را با استفاده از عمل ضرب ترکیب کنیم، مانند t با t (T_1 با T_1):

$gap > t * t;$

$(1, 3, 5, 7, 9, 11)(2, 4, 6, 8, 10, 12)$

توجه داشته باشید که خروجی این فرمان شامل ۲-دور است: یکی که از نغمه‌های هم‌نام زوج و دیگری از نغمه‌های هم‌نام فرد می‌گذرد. چنین دوره‌هایی بر اساس تعریف مجزا هستند.

۵.۳. همچنین، ممکن است توان‌های یک عملگر را مشابه با قبل در ترکیب با خودش استفاده کنیم^{۱۹}:

$gap > t^2;$

$(1, 3, 5, 7, 9, 11)(2, 4, 6, 8, 10, 12)$

در اینجا، t^2 بر عمل‌های T_2 منطبق است.

^{۱۸} در حقیقت، GAP مجاز به حساب پیمانه‌ای است، اما در آن در اینجا منجر به سطحی از پیچیدگی می‌شود که برای اهداف ما ضروری نیست. کتابچه‌ی راهنمای $gap > ?modulo$ را ببینید.
^{۱۹} علامت هشتم “ $^$ ”، چندین معنای ممکن در GAP دارد که همه آن‌ها با نمادگذاری نمایی ارتباط دارند. در اینجا، به‌طور تحت‌اللفظی به معنی توان (یعنی، t^i) است. ممکن است یک تابع (بخش ۳.۴ را ببینید) یا یک مزدوج (بخش ۱۵.۴ را ببینید) را نشان دهد.

۶.۳. ممکن است وارون‌ها را به صورت زیر نمایش دهیم^{۲۰}:

```
gap > t^-1;
(1,12,11,10,9,8,7,6,5,4,3,2)
gap > (t^2)^-1;
(1,11,9,7,5,3)(2,12,10,8,6,4)
```

به عبارت دیگر، t^{-1} با $T_{11}(=T_{-1})$ و $t^{-2} = (t^2)^{-1}$ با $T_{10}(=T_{-2})$ مطابقت دارد. ممکن است برای چنین هم‌ارزی‌هایی نیز آزمایش کنیم:

```
gap > (t^2)^-1 = t^10;
true
```

۷.۳. می‌توانیم از t به عنوان مولد گروه، یعنی T به کار ببریم:

```
gap > T := Group(t);
Group([ (1,2,3,4,5,6,7,8,9,10,11,12) ])
```

خروجی این فرمان نشان می‌دهد که T گروه تولیدشده به وسیله t است که در آن t جایگشت $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$ است. T با گروه دوری مرتبه‌ی ۱۲، (C_{12}) یکریخت است:

```
gap > IsomorphismGroups(T, CyclicGroup(12));
-> [ (1,5,9)(2,6,10)(3,7,11)(4,8,12), (1,4,7,10)(2,5,8,11)(3,6,9,12) ]
[ f3, f1*f2 ]
```

خروجی این فرمان نیازمند توضیحاتی است. اول، GAP بیان می‌کند که یک یکریختی از T به C_{12} وجود دارد؛ در غیر این صورت، پیام “fail” را ارسال خواهد کرد. با این حال، پیش از این به جای توصیف یکریختی بر حسب مولد t مربوط به T که قبلاً تعریف کردیم، GAP از t^3 و t^4 (T_4 و T_3) استفاده می‌کند که هر دو T را نیز تولید می‌کنند. (دلیل اصلی انتخاب این مولدها باید با استدلال برنامه‌نویسی داخلی GAP انجام شود و گاهی اوقات، خارج از حوزه این خودآموز است). بنابراین، GAP این مولدها را به $f1.f2$ و $f3$ می‌نگارد که مولدهای مجرد گروه دوری مرتبه‌ی ۱۲ هستند و در کتابخانه‌ی داخلی GAP ذخیره و نگهداری شده‌اند. مجدداً تأکید می‌شود که نکته مهم در اینجا، وجود یک یکریختی است.

۸.۳. ممکن است مرتبه‌ی T را بررسی کنیم:

```
gap > Size(T);
12
```

و عناصر آن را نشان دهیم:

```
gap > List(T);
[ (), (1,9,5)(2,10,6)(3,11,7)(4,12,8), (1,5,9)(2,6,10)(3,7,11)(4,8,12),
(1,11,9,7,5,3)(2,12,10,8,6,4), (1,7)(2,8)(3,9)(4,10)(5,11)(6,12),
(1,3,5,7,9,11)(2,4,6,8,10,12), (1,12,11,10,9,8,7,6,5,4,3,2),
(1,8,3,10,5,12,7,2,9,4,11,6), (1,4,7,10)(2,5,8,11)(3,6,9,12),
```

^{۲۰} به کاربرد پرانتز در فرمان GAP دوم “ $gap > (t^2)^{-1}$ ” توجه کنید. هنگامی که نما به طور کلی شرکت‌پذیر نیست، پرانتز ضروری است.

$$(1,10,7,4)(2,11,8,5)(3,12,9,6), (1,6,11,4,9,2,7,12,5,10,3,8), \\ (1,2,3,4,5,6,7,8,9,10,11,12) \]$$

در این خروجی، GAP این عناصر را به شکل نمادگذاری دوری در مرتبه‌ی

$$t^{\circ}, t^{\wedge}, t^{\mathfrak{f}}, t^{!^{\circ}}, t^{\mathfrak{e}}, t^{!^!}, t^{\vee}, t^{\mathfrak{z}}, t^{\mathfrak{q}}, t^{\omega}, t(\mathrm{T}_{\circ}, \mathrm{T}_{\wedge}, \mathrm{T}_{\mathfrak{f}}, \mathrm{T}_{!^{\circ}}, \dots)$$

می‌نگارد. (همانند قبل، استدلال GAP برای فهرست‌نویسی عناصر بر اساس این طرح، خارج از حوزه این خودآموز است، اما این مرتب‌سازی در بخش‌های بعدی ذکر می‌شود).

۹.۳. GAP می‌تواند مدار یک نغمه‌ی هم‌نام را تحت اعضای گروه T تعیین کند که در اینجا با استفاده از نغمه‌ی هم‌نام ۱ نشان داده شده است:

$$gap > Orbit(T, 1);$$

[1, 9, 5, 11, 7, 3, 12, 8, 4, 10, 6, 2]

این مدار شامل اعضای مجموعه‌ی نغمه‌های هم‌نام است که نغمه‌ی هم‌نام ۱ تحت عناصر گروه T نگاشته می‌شود. یعنی، تحت $(T_0)^t$ ، ۱ به ۱، تحت $(T_8)^t$ ، ۱ به ۹ و تحت $(T_4)^t$ ، ۱ به ۵ نگاشته می‌شود و به همین ترتیب، با استفاده از تمام دوازده ترانهش‌های هم‌نام انتقال پیدا می‌کند. در اینجا توجه داشته باشید که مرتب‌سازی نغمه‌ی هم‌نام خروجی به وسیله‌ی GAP از مرتب‌سازی عناصر گروه در خروجی فرمان “ $List(T)$ ” در کدهای بالا نتیجه می‌شود. به علاوه، از آنجایی که عمل T بر PCS از نوع انتقالی است، پس

```
gap > IsTransitive(T);
```

true

نغمه‌ی هم‌نام ۱ به تمام دوازده عضو مجموعه‌ی نغمه‌های هم‌نام نگاشته می‌شود.

۱۰.۳. GAP می‌تواند مدار یک مجموعه‌ی نامرتب از نغمه‌های هم‌نام را نیز نمایش دهد، مانند یک آکورد سه‌صدایی C ماژور {۰, ۴, ۷} (یادآوری می‌شود که از عدد صحیح ۱۲ به‌جای نغمه‌ی هم‌نام صفر استفاده کردیم):

```
gap > Orbit(T, [4, 7, 12], OnSets);
```

$$[[4, 7, 12], [4, 8, 11], [3, 8, 12], [2, 6, 9], [1, 6, 10],$$
$$[2, 5, 10], [1, 5, 8], [5, 9, 12], [1, 4, 9], [3, 7, 10],$$
$$[2, 7, 11], [3, 6, 11]$$

در اینجا، خروجی (در مجموعه‌هایی متوالی که داخل پرانتز به توان ۲ رسیده‌اند) مجموعه‌ی نغمه‌های هم‌نامی را نشان می‌دهد که آکورد سه‌صدایی C ماژور تحت اعضای گروه ترانه‌ش نگاشته می‌شود: C ماژور، E ماژور، $A\flat$ ماژور، D ماژور، $F\sharp$ ماژور و به‌همین ترتیب از طریق تمام دوازده آکورد سه‌صدایی ماژور. مجدداً، مرتب‌سازی عملگرهایی که آکورد سه‌صدایی C ماژور را برای این آکوردهای سه‌صدایی متنوع دیگر اجرا می‌کنند، مربوط به ترتیب ارائه شده در بخش ۹.۳ است؛ توجه داشته باشید که عملگرهایی که در این توالی در دسته وارون‌های مربوط به عملگرهای موجود در مرتب‌سازی بخش قبلی قرار می‌گیرند، در این مرتب‌سازی در نظر گرفته نمی‌شوند. باید توجه داشت که، با وجود این‌که GAP ، $\{0, 4, 7\}$ را به‌عنوان یک مجموعه‌ی نامرتب از نغمه‌های هم‌نام در این موضوع در نظر می‌گیرد، اما باید این‌را در افزایش مرتبه در فرمان “[4, 7, 12]” وارد کنیم؛ در غیر این‌صورت، یک پیام خطا دریافت می‌کنیم.

۱۱.۳. GAP می‌تواند مدارهای مجموعه‌ی نغمه‌های هم‌نام را مدل‌سازی کند که عمل T روی آن باوفا نیست، مانند آکورد سه‌صدایی افزوده $\{0, 4, 8\}$:

```
gap > Orbit(T, [4, 8, 12], OnSets);
[[ 4, 8, 12 ], [ 2, 6, 10 ], [ 1, 5, 9 ], [ 3, 7, 11 ] ]
```

در اینجا، خروجی چهار آکورد سه‌صدایی افزوده را نشان می‌دهد که برای آن $\{0, 4, 8\}$ می‌تواند تحت اعضای گروه ترانش نشا شده شود. عمل T بر این مجموعه از مجموعه‌ها باوفا نیست، زیرا عناصر غیرهمانی معین این گروه (به‌ویژه $t^4[T_4]$ و $t^8[T_8]$) از عنصر $t^0[T_0]$ غیرقابل تشخیص هستند. یعنی، هر چهار آکورد سه‌صدایی افزوده ناوردا هستند.

۱۲.۳. سپس، عمل انعکاس i را تعریف کنید (این برچسب در GAP برای عمل‌های I یا I_0 استفاده می‌شود):

```
gap > i := (1, 11)(2, 10)(3, 9)(4, 8)(5, 7);
(1, 11)(2, 10)(3, 9)(4, 8)(5, 7)
```

خروجی این فرمان فقط دوره‌های نغمه‌های هم‌نامی را منعکس می‌کند که برای تعریف این عمل‌ها استفاده کرده‌ایم. (قابل ذکر است، ورودی و خروجی، دوره‌های تک نغمه‌های هم‌نام 0 [که در GAP برابر ۱۲ است] و ۶ را حذف می‌کنند که هر دوی آن‌ها تحت انعکاس ناوردا هستند). علاوه بر این، از آنجایی‌که GAP از نگارش راست-تابعی به جای نگارش چپ-تابعی استفاده می‌کند، پس نمادگذاری ما برای ترکیب انعکاس با ترانش همواره در مرتبه مخالف نمادگذاری موجود در نظریه موسیقی ظاهر می‌شود. برای مثال، عمل‌های T_1I (یا I_1) به صورت it (یا I_2) به صورت it^2 و ... ظاهر می‌شود.

۱۳.۳. t و i به همراه هم، گروه ترانش و انعکاس T/I را تولید می‌کنند:

```
gap > TI := Group(t, i);
Group([ (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12), (1, 11)(2, 10)(3, 9)(4, 8)(5, 7) ])
```

دوباره، خروجی دوره‌های عمل‌ها را نشان می‌دهد که برای ساخت این گروه استفاده کرده‌ایم. T/I با گروه دو وجهی مرتبه‌ی ۲۴ (D_{24}) یک‌ریخت است:

```
gap > IsomorphismGroups(TI, DihedralGroup(24));
[ (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12), (1, 11)(2, 10)(3, 9)(4, 8)(5, 7) ] ->
[ f2, f1*f3*f4^2 ]
```

همانند خروجی فرمان در بخش ۷.۳ که یک یک‌ریختی از T به C_{12} را نمایش داد، این خروجی بیان می‌کند که GAP یک یک‌ریختی از T/I به D_{24} دارد؛ و همانند قبل، یک‌ریختی را بر حسب مولدهای مختلف گروه‌های مربوطه ارائه می‌دهد. برخلاف مثال قبلی (در بخش ۷.۳، GAP مولدهای $t^3[T_3]$ و $t^4[T_4]$ برای T را با مولد $t[T_1]$ جایگزین کرد که از آن‌ها برای تعریف این گروه استفاده کرده‌ایم)، GAP چنین جایگزینی را در این مثال انجام نمی‌دهد. بلکه، از مولدهای $t[T_1]$ و $i[I_0]$ برای T/I استفاده می‌کند. بنابراین یک‌ریختی، این مولدها را به مولدهای گروه مجرد D_{24} ، f_2 و $f_1.f_3.f_4^2$ می‌نگارد که در کتابخانه‌ی GAP ذخیره و نگهداری می‌شوند.

۱۴.۳. عمل‌های ضربی m اثر موريس [۱۹] را (معمولاً، به صورت M یا M_5 در نوشته‌های موسیقی نظری ارائه می‌شود) با نغمه‌های هم‌نام ترکیب می‌کنیم. یعنی، برای هر $x \in PCS$ ، داریم $12x \bmod 5x$ $m: x \rightarrow$ ^{۲۱}. لذا

^{۲۱} برای نگارش‌های قبلی عمل m و به‌ویژه در موضوع نظریه دوره‌های، استار و موريس [۲۵]، [۲۴]، [۲۶] و استار [۲۴] را ببینید. برای نگارش‌های بعدی، موريس [۲۰] را ببینید.

$gap > m := (1,5)(2,10)(4,8)(7,11);$

$(1,5)(2,10)(4,8)(7,11)$

و GAP مجدداً، خروجی دورهایی که وارد کرده‌ایم را تأیید می‌کند و ورودی و خروجی دورهای تک‌عضوی را حذف می‌کند.

۱۵.۳. اتصال m به T/I منجر به یک گروه TTO از مرتبه ۴۸ می‌شود که مجموعه‌ای از تبدیلات آفین روی $\mathbb{Z}_{12}$ است:

$gap > TTO := Group(t, i, m);$

$Group([(1,2,3,4,5,6,7,8,9,10,11,12),$

$(1,11)(2,10)(3,9)(4,8)(5,7), (1,5)(2,10)(4,8)(7,11)])$

خروجی این فرمان نشان می‌دهد که TTO ، گروه تولیدشده به وسیله‌ی سه دور است. TTO با ضرب مستقیم گروه‌های دو وجهی مرتبه‌ی ۶ و ۸، $(D_6 \times D_8)$ یکریخت است:

$gap > IsomorphismGroups(TTO, DirectProduct([enter]$

$> DihedralGroup(6), DihedralGroup(8)));$

$[(1,2,3,4,5,6,7,8,9,10,11,12), (1,11)(2,10)(3,9)(4,8)(5,7),$

$(1,5)(2,10)(4,8)(7,11)] \rightarrow [f2*f4, f1*f3*f4*f5, f1*f5]$

در خروجی این فرمان، مولدهای $[T_1]$ ، $i[I_0]$ و $m[M_5]$ مربوط به TTO به ضرب‌های f_4, f_3, f_2, f_1 و f_5 نگاشته می‌شوند؛ اینها مولدهایی هستند که کتابخانه‌ی داخلی GAP از آن‌ها برای ساخت گروه مجرد مرتبه‌ی ۴۸ مربوطه استفاده می‌کند. نکته مهم این است که GAP در واقع یک یکریختی را تعیین می‌کند.

۴. یکریختی‌های شبکه‌ی کلمپن‌هاور

۱۰.۴. مطالعه‌ی شبکه‌های کلمپن‌هاور (لویین [۱۷] و کلمپن‌هاور [۱۱])، یک روش برای استفاده از GAP به عنوان

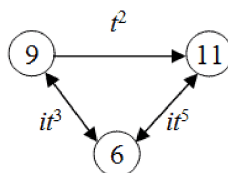
ابزاری برای نظریه‌ی موسیقی است. شبکه‌های کلمپن‌هاور یا $K-nets$ ، گراف‌هایی گسسته با گره‌ها (رأس‌ها) و پیکان‌هایی (یال‌ها) هستند که به ترتیب در مجموعه‌های از نغمه‌های هم‌نام و در گروه T/I برچسب زده شده‌اند (شکل ۳ را ببینید):

$gap > t^2; i * t^3; i^5;$

$(1,3,5,7,9,11)(2,4,6,8,10,12)$

$(1,2)(3,12)(4,11)(5,10)(6,9)(7,8)$

$(1,4)(2,3)(5,12)(6,11)(7,10)(8,9)$



شکل ۳. یک نمونه K-net

Figure 3: A sample K-net

اگر همانند بالا، برجسب‌های مربوط به پیکان‌های موجود در GAP را وارد کنیم، آنگاه خروجی تولید می‌شود که دور (دورها) را نشان می‌دهد و در آن فهرست گره‌ها برقرار هستند. برای مثال، پیکانی (تک‌واحدی) که گره حاصل از pc۹ را به گره حاصل از pc۱۱ متصل می‌کند با $t^2(T_2)$ برجسب زده شده است؛ دورهای آن به صورت زیر هستند:

$$(1, 3, 5, 7, \underline{9}, 11)(2, 4, 6, 8, 10, 12)$$

که نغمه‌ی هم‌نام ۹ را به نغمه‌ی هم‌نام ۱۱ (زیر آن خط کشیده شده است) و سپس ۱، ۳، ۵ و ۷ می‌نگارد و بالاخره به ۹ برمی‌گردد. از این رو، یک دور به طول ۶ (یک ۶-دور) است. از آنجایی که نغمه‌ی هم‌نام ۱۱ تا ۱ ادامه پیدا می‌کند و مستقیماً به ۹ برمی‌گردد، پس در این شبکه از پیکان تک‌واحدی استفاده می‌کنیم.

۲.۴. از سوی دیگر، دوره‌های عمل‌های موجود در شکل ۳ که نغمه‌های هم‌نام ۹ و ۶ و ۱۱ و ۶ را به یکدیگر متصل می‌کنند (it^3 و it^5 و $(I_5$ و $I_3)$)، بازگشتی هستند. یعنی، وارون‌های خود هستند. اگر برجسب‌های پیکان‌های مربوط به آن‌ها را وارد کنیم، آنگاه دوره‌های زیر را داریم که به طول ۲ هستند (۲-دور):

$$(1, 2)(3, 12)(4, 11)(5, 10)(\underline{6, 9})(7, 8)$$

$$(1, 4)(2, 3)(5, 12)(\underline{6, 11})(7, 10)(8, 9)$$

نغمه‌ی هم‌نام ۶ در اولین ۲-دور به نغمه‌ی هم‌نام ۹ و ۹ به ۶ نگاشته می‌شود؛ نغمه‌ی هم‌نام ۶ در دومین ۲-دور به ۱۱ نگاشته می‌شود و برعکس. از این رو، این شبکه، پیکان‌های دوجهته را برای اتصال گره‌های مربوط به آن‌ها ترکیب می‌کند.

۳.۴. شبکه‌های تام ممکن است از طریق یک‌ریختی‌ها با شبکه‌ی دیگری ارتباط داشته باشند، که این یک‌ریختی‌ها مربوط به اعضای گروه خودریختی T/I ، یعنی $Aut(T/I)$ هستند:

gap > AutTI := AutomorphismGroup(TI);

<group of size 48 with 4 generators>

خروجی نشان می‌دهد که $Aut(TT/I)$ گروهی از مرتبه‌ی ۴۸ است و عدد مولدهایی را ارائه می‌دهد که GAP برای ساخت آن استفاده می‌کند. همان‌طور که لویین [۱۷] اشاره کرد، این گروه (که بر ۲۴ عضو از T/I عمل می‌کند) با TTO یک‌ریخت است و آن را $alpha$ یک‌ریخت^{۲۲} می‌نامیم:

gap > alpha := IsomorphismGroups(TTO, AutTI);

[(1,2,3,4,5,6,7,8,9,10,11,12), (1,11)(2,10)(3,9)(4,8)(5,7), (1,5)(2,10)

(4,8)(7,11)] ->

[[(1,6,11,4,9,2,7,12,5,10,3,8), (1,3)(4,12)(5,11)(6,10)(7,9)] ->

[(1,6,11,4,9,2,7,12,5,10,3,8), (1,2)(3,12)(4,11)(5,10)(6,9)(7,8)],

[(1,6,11,4,9,2,7,12,5,10,3,8), (1,3)(4,12)(5,11)(6,10)(7,9)] ->

[(1,8,3,10,5,12,7,2,9,4,11,6), (1,2)(3,12)(4,11)(5,10)(6,9)(7,8)],

^{۲۲} ذکر این نکته ضروری است، برخلاف آنچه که در خروجی برنامه همراه ("SampleGAPprogram.txt") نشان داده شد، GAP ممکن است یک‌ریختی متفاوتی از TTO به $Aut(T/I)$ را در هنگام مطالعه‌ی این برنامه به $alpha$ اختصاص دهد: برای مثال، ممکن است مولدهای t ، i و m گروه آفین TTO را به ابر-عملگرهای $F\langle 5, 8 \rangle$ و $F\langle 11, 5 \rangle$ ، $F\langle 1, 7 \rangle$ ، $F\langle 1, 9 \rangle$ و $F\langle 1, 11 \rangle$ بنگارد. در نتیجه، تعاریف دیگری را در برنامه‌ی نمونه برای ابر- t و ابر- i به جای تعریف موجود در ۵.۴ و ۶.۴ ارائه می‌دهیم؛ یعنی قرار می‌دهیم، $gap > hyp_t := (t^7)^{alpha}$ و $gap > hyp_i := (i^{alpha} * t^{alpha})$. (تحت این نگاشت خاص، عمل ابر- m در بخش ۱۲.۴ نیازی به تعریف مجدد ندارد.) تعاریفی که در کتابچه‌ی یادداشت Sage آنلاین "GAP-MTO" ظاهر می‌شوند، هنوز هم اندکی متفاوت هستند.

$$[(1,6,11,4,9,2,7,12,5,10,3,8), (1,3)(4,12)(5,11)(6,10)(7,9)] \rightarrow [(1,2,3,4,5,6,7,8,9,10,11,12), (1,11)(2,10)(3,9)(4,8)(5,7)]$$

GAP، مولدهای TTO که قبلاً تعریف شده‌اند (یعنی i, t, m و (T_1, I_0, M_0)) را تحت $alpha$ به سه خودریختی از گروه T/I می‌نگارد که با یکدیگر گروه $Aut(T/I)$ را تولید می‌کنند. این خودریختی‌ها در خروجی (اولین پیکان زیر " $>$ ") برحسب عمل‌های مربوط به آن‌ها بر مولدهای (دلخواه) گروه T/I (t^5 و it^4 به T_5 و I_4) ظاهر می‌شوند.^{۲۳} ممکن است از نمادگذاری لوبین [۱۷] برای برچسب‌گذاری این خودریختی‌ها استفاده کنیم: $F\langle u, j \rangle$ که t^n را به t^{un} (T_n را به T_{un}) و it^n را به $it^{(un+j)}$ (I_n را به $I_{(un+j)}$) می‌نگارد، که در آن $u \in \{1, 5, 7, 11\}$ و $j \in \mathbb{Z}_{12}$. مشاهده می‌کنیم که $t(T_1)$ تحت $alpha$ به خودریختی T/I نگاشته می‌شود که t^5 را به t^5 (T_5 را به T_5) و it^4 را به it^3 (I_4 را به I_3) می‌نگارد؛ به‌همین دلیل لوبین [۱۷] آن را $F\langle 1, 11 \rangle$ نامید. سپس، i (I_0) تحت $alpha$ به خودریختی نگاشته می‌شود که t^5 را به t^5 (T_5 را به T_5) و it^4 را به it^3 (I_4 را به I_3) می‌نگارد، که لوبین [۱۷]، نماد $F\langle 11, 7 \rangle$ را پیشنهاد می‌کند. در نهایت، m (M_0) تحت $alpha$ به خودریختی نگاشته می‌شود که t^5 را به t (T_5 را به T_1) و it^4 را به i (I_4 را به I_0) می‌نگارد که لوبین [۱۷]، نماد $F\langle 5, 4 \rangle$ را پیشنهاد می‌کند. بنابراین، $F\langle 1, 11 \rangle$ ، $F\langle 11, 7 \rangle$ و $F\langle 5, 4 \rangle$ به‌همراه هم، گروه $Aut(T/I)$ از مرتبه‌ی ۴۸ را تولید می‌کنند.

۴.۴. از آنجایی که $Aut(T/I)$ با TTO یکریخت است و TTO شامل زیرگروه T/I است، پس $Aut(T/I)$ شامل زیرگروهی است که با T/I یکریخت است؛ این زیرگروه را $Hyp(T/I)$ (یا I -بر) می‌نامیم. این زیرگروه شامل خودریختی‌های T/I با $u \in \{1, 11\}$ است. یکریختی‌هایی که از طریق خودریختی‌ها با $u = 1$ به‌دست می‌آیند را مثبت و یکریختی‌هایی که از طریق $u = 11$ به‌دست می‌آیند را منفی بنامید. ممکن است خودریختی نظیر T/I انتخاب شود که تصویر $tt(T_1)$ تحت $alpha$ است، یعنی $F\langle 1, 11 \rangle$ به‌عنوان یک مولد این زیرگروه است؛ و همچنین، خودریختی نظیر T/I انتخاب شود که تصویر i (I.) تحت $alpha$ است، یعنی $F\langle 11, 7 \rangle$:

$$gap > HypTI := Group(t^{\alpha}, i^{\alpha});$$

<group with 2 generators>

از ۴.۲ یادآوری می‌شود که نمادگذاری نمایی به‌کار رفته در این فرمان (برای مثال، “ t^{α} ”), یک تابع را نشان می‌دهد که در این حالت یک یک‌ریختی است. بنابراین، خروجی صرفاً شرح مختصری از ساختار تولیدشده را ارائه می‌دهد (یعنی، گروهی با دو مولد)؛ اما اندازه یا هیچ اطلاعات ظاهری دیگری را نشان نمی‌دهد.

۵.۴. با اقتباس از نمادگذاری اخیر لویین [۱۸] برای عملگرهای ابر- T/I برای نگارش کنونی خود $\langle tj \rangle$ برای $F\langle 1, j \rangle$ و $\langle itj \rangle$ برای $F\langle 11, j \rangle$ توجه داریم که t , T_1 را تحت $alpha$ به خودریختی T/I می‌نگارد که $\langle t^{11} \rangle$ نامیده می‌شود و t^n را به t^n (T_n را به T_n) و it^n را به it^{n+1} (I_{n+1} را به I_n) می‌نگارد. بنابراین، برای تعریف یک ابر- t (ابر- T_1)، قرار دهد:

$$gap > hyp_t := (t^{11})^\alpha;$$
$$[(1,6,11,4,9,2,7,12,5,10,3,8), (1,2)(3,12)(4,11)(5,10)(6,9)(7,8) \rightarrow$$

$$[(1,6,11,4,9,2,7,12,5,10,3,8), (1,3)(4,12)(5,11)(6,10)(7,9)]$$

۲۳ نمایش GAP گروه T/I ممکن است توسط هر عملگر t از مرتبه‌ی ۱۲ (t^5, t^4, t^3) یا (t^{11}, t^8, t^5) تولید شود. دلیل این‌که در اینجا انتخاب‌های GAP برای مدل‌سازی این خودریختی‌ها روی مولدهای t^5 و t^8 است، دلخواه به‌نظر می‌رسد، اما در حقیقت پایهای آن است که فراتر از حیطه‌ی این خودآموز است. در ضمن، GAP ممکن است مولدهای دیگری به جز t^5 و t^8 را روی رایانه‌های مختلف با وجود اجرای دقیق همان نرم‌افزار را در هنگام مطالعه‌ی یک برنامه در مقابل دنباله‌ای از فرمان‌های نوشته‌شده انتخاب کند. در این حالت، باید عمل‌های ابر- t ، ابر- i و ابر- m موجود در ۵.۴، ۶.۴ و ۹.۴ را دوباره تعریف کنیم.

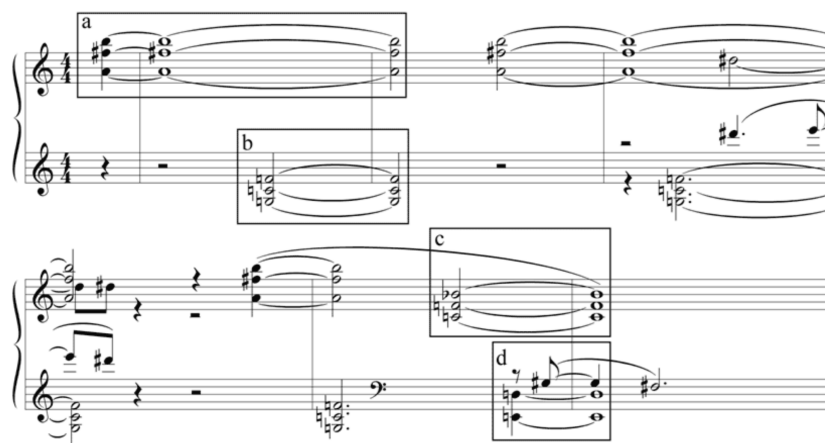
یعنی، $\langle t \rangle$ که t^n را به t^n (یا به T_n) و it^n را به it^{n+1} (یا به I_{n+1}) می‌نگارد و همانند $F\langle 1, j \rangle$ عمل می‌کند. توجه داشته باشید که مولد t^5 (یا T_5) از T/I در خروجی این فرمان به خودش نگاشته می‌شود و مولد it^3 (یا I_3) به it^4 (یا I_4) نگاشته می‌شود.

۶.۴. به‌طور مشابه، i (یا I_0) تحت $alpha$ به خودریختی به نام $\langle it^7 \rangle$ می‌نگارد، یعنی، با خودریختی سازگار است که t^n را به t^{-n} (یا به T_{-n}) و it^n را به it^{-n+7} (یا به I_{-n+7}) می‌نگارد. از این‌رو، برای تعریف عمل‌های ابر- i (یا I_-)، $\langle i \rangle$ ، قرار دهید:

```
gap > hyp_i := (i^alpha) * ((t^7)^alpha);
[ (1,8,3,10,5,12,7,2,9,4,11,6), (1,2)(3,12)(4,11)(5,10)(6,9)(7,8) ] ->
[ (1,6,11,4,9,2,7,12,5,10,3,8), (1,8)(2,7)(3,6)(4,5)(9,12)(10,11) ]
```

که t^n را به t^{-n} (یا به T_{-n}) و it^n را به it^{-n+0} (یا به I_{-n+0}) می‌نگارد. دوباره، خروجی این فرمان، یک عمل را بر مولدهای گروه T/I نشان می‌دهد که ناشی از نگاشت گروه به خودش است و نگاشت t^7 به t^5 (یا T_5) و it^3 به it^9 (یا I_9) را نشان می‌دهد.

۷.۴. در شکل ۴، بر اساس تحلیل فیلیپ لمبرت [۱۴]، شش میزان از اپوس ۲۴، شماره ۶ از آثار پیانوی آرنولد شونبرگ با نام شش قطعه کوچک پیانو انتخاب شده است. ۲۶



شکل ۴. میزان ۱ تا ۶ اپوس ۱۹، شماره ۶ اثر آرنولد شونبرگ در سال ۲۰۰۲ با نام شش قطعه کوچک پیانو

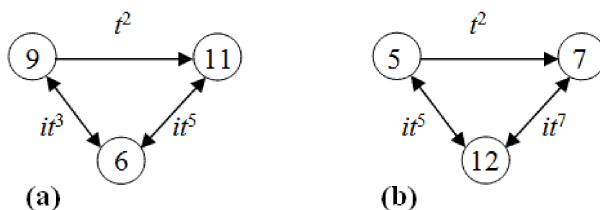
Figure 4: Schoenberg, Sechs kleine Klavierstücke, op. 19, no. 6, measures 1-6 (following Lambert 2002)

^{۲۴} در موسیقی، اپوس یا شماره اثر (opus)، یک شماره‌ی کلی هست که هر آهنگساز برای انتشار آثار خود اختصاص می‌دهد: مترجم.
^{۲۶} به نوبه خود، لمبرت [۱۴]، تحلیل لوبین [۱۷] از سه قطعه پرسیا و پرتنین در اپوس ۱۹، شماره ۶ اثر آرنولد شونبرگ را نقطه شروع تحقیقات خود می‌داند.

۸.۴. توجه داریم که شبکه‌ی برچسب‌گذاری شده (a) در شکل ۵ (مربوط به مجموعه‌ی سه نغمه‌ای که در این مثال با "a" برچسب زده شده است) مربوط به شبکه‌ای است که به وسیله‌ی $\langle t^2 \rangle$ با (b) در شکل ۵ برچسب زده شده است. به‌ویژه، پیکانی که با عملگر ترانهش t^2 (T_2) در شکل (a) برچسب زده شده است، در شکل (b) به پیکان متناظر با همان برچسب نگاشته می‌شود؛ و پیکان‌هایی که در شکل (a) با عملگرهای انعکاس it^3 و it^5 (I_5 و I_3) برچسب زده شده‌اند، در شکل (b) به $it^{3+2} = it^5$ و $it^{5+2} = it^7$ (I_7 و I_5) نگاشته می‌شوند. خروجی، دوره‌های شبکه‌ی (b) را نشان می‌دهد که تصویری از دوره‌های شبکه‌ی (a) تحت $\langle t^2 \rangle$ هستند:

```
gap > (t^2)^(hyp_t^2);
(i*t^3)^(hyp_t^2); (i*t^5)^(hyp_t^2);
(1,3,5,7,9,11)(2,4,6,8,10,12)
(1,4)(2,3)(5,12)(6,11)(7,10)(8,9)
(1,6)(2,5)(3,4)(7,12)(8,11)(9,10)
```

همانند یک ابر-عملگر، $\langle t^2 \rangle$ را تابعی روی T/I می‌نامیم (یعنی، یک خودریختی گروه)، از این رو، این را با استفاده از نمادگذاری نمایی در GAP اعمال می‌کنیم. یعنی، فرمان " $(t^2)^{hyp_t^2}$ " به GAP دستور می‌دهد تا تصویر $t^2(T_2)$ را تحت $\langle t^2 \rangle$ (ابر- T_2) نمایش دهد؛ همین روند را برای برچسب‌های پیکان باقیمانده انجام می‌دهیم.



شکل ۵. دو K-شبکه یک‌نگار مثبت از لمبرت [۱۴]

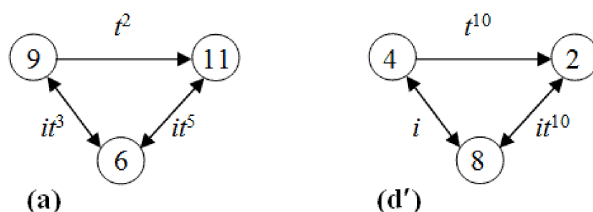
Figure 5: Two positively isographic K -nets from Lambert 2002

۹.۴. به‌طور مشابه، شبکه‌ای که در شکل ۶ با (a) برچسب زده شده است به وسیله‌ی $\langle it^3 \rangle$ به شبکه‌ی برچسب‌شده با (d') مرتبط می‌شود (مجدداً از برچسب‌گذاری مجموعه‌های سه نغمه‌ای شکل ۴ نتیجه شده است t^3)؛ عکس این رابطه زمانی برقرار است که $\langle it^3 \rangle$ بازگشتی باشد. به‌ویژه، پیکان t^2 (T_2) در شکل (a) به پیکان $t^{-2} = t^{10}$ (T_{10}) در شکل (d') نگاشته می‌شود؛ و پیکان‌های it^3 و it^5 (I_5 و I_3) موجود در شکل (a) به ترتیب به پیکان‌های $it^{-5+3} = it^{10}$ و $it^{-3+3} = it^0$ (I_{10} و I_0) در شکل (d') نگاشته می‌شوند:

```
gap > (t^2)^(hyp_i * hyp_t^3);
(1,11,9,7,5,3)(2,12,10,8,6,4)
gap > (i * t^3)^(hyp_i * hyp_t^3);
(1,11)(2,10)(3,9)(4,8)(5,7)
gap > (i * t^5)^(hyp_i * hyp_t^3);
(1,9)(2,8)(3,7)(4,6)(10,12)
```

^{۲۷} لمبرت [۱۴] در ابتدا، تفسیرهای مختلفی از مجموعه‌ی سه‌تایی را پیشنهاد کرد که در شکل ۴ با d برچسب زده شده است؛ وی آن‌ها را "(d') نامید. در نتیجه، تفسیر مجددی را پیشنهاد کرد که آن را با "(d') برچسب زد. همان‌طور که تفسیر اخیر را در اینجا ترکیب کردیم، برچسب‌گذاری آن را حفظ می‌کنیم.

مجدداً، خروجی تصویر (دو نمادگذاری دوری) شبکه‌ی پیکان‌ها را تحت ابر-عملگر مربوطه $\langle it^3 \rangle$ نشان می‌دهد که به صورت فرمان “ $(hyp_i * hyp_t^3)$ ” ارائه شده است. دورهای حاصل با عملگرهایی تطابق دارند که پیکان‌ها را در شبکه‌ی (d') برجسب می‌زنند، یعنی عملگرهای i, t^{10} و i^{10} (I_0 و I_{10}).



شکل ۶. دو K-شبکه یک‌نگار منفی از لمبرت [۱۴]

Figure 6: Two negatively isographic K -nets from Lambert 2002

۱۰.۴. حال، می‌توانیم ابر-عملگری را تعیین کنیم که شبکه‌ی (b) را به شبکه‌ی (d') مرتبط می‌کند؛ این کار را با استفاده از ابر-عملگرهایی انجام می‌دهیم که شبکه‌ی (a) را به (b) و (a) را به (d') متصل می‌کنند. پیش از این می‌دانستیم که $\langle t^2 \rangle$ ، (a) را به (b) وصل می‌کند؛ از این رو، عکس آن یعنی $\langle t^2 \rangle^{-1}$ ، (b) را به (a) وصل می‌کند.^{۲۸} همچنین، می‌دانیم که (a) تحت $\langle it^3 \rangle$ به (d') نگاشته می‌شود. بنابراین، (b) تحت ترکیب $\langle it^3 \rangle$ به (d') با همان مرتبه متصل می‌شود. خروجی زیر، عمل مربوط به این ترکیب را بر برجسب‌های پیکان (b) نشان می‌دهد که دورهای (d') را تولید می‌کند:

```
gap > (t^2)^(((hyp_t^2)^-1) * hyp_i * hyp_t^3);
(1,11,9,7,5,3)(2,12,10,8,6,4)
gap > (i * t^5)^(((hyp_t^2)^-1) * hyp_i * hyp_t^3);
(1,11)(2,10)(3,9)(4,8)(5,7)
gap > (i * t^7)^(((hyp_t^2)^-1) * hyp_i * hyp_t^3);
(1,9)(2,8)(3,7)(4,6)(10,12)
```

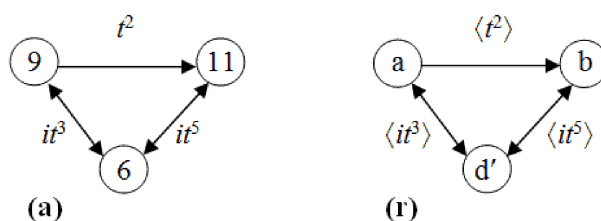
دورهای درون این خروجی با دورهای عملگرهایی تطابق دارند که پیکان‌ها را در (d') برجسب می‌زنند، یعنی عملگرهای i, t^{10} و i^{10} (I_0 و I_{10}).

۱۱.۴. $\langle it^5 \rangle$ (ابر- I_5) از ترکیب ابر-عملگرهای $\langle t^2 \rangle^{-1}$ و $\langle it^3 \rangle$ به دست می‌آید:

```
gap > ((hyp_t^2)^-1) * hyp_i * hyp_t^3 = hyp_i * hyp_t^5;
true
```

ممکن است از این همراه با $\langle t^2 \rangle$ و $\langle it^3 \rangle$ برای نمایش رابطه‌ی بازگشتی لمبرت استفاده کنیم؛ این رابطه بین شبکه‌ی (a) و یک ابر-شبکه (لمبرت [۱۴، مثال ۵]) توصیف می‌شود که آن را با (r) برجسب زده‌ایم. در شکل ۷، (r) گره‌هایی دارد که توسط شبکه‌های (a)، (b) و (d') اشغال شده‌اند و دارای پیکان‌هایی است که در $Hyp(T/I)$ برجسب زده شده است.

^{۲۸} می‌توانیم وجود یک وارون را تضمین کنیم، زیرا خودریختی از یک گروه با عضو وارون برای هر عنصر تشکیل شده‌اند.



شکل ۷. تکرار بین شبکه (a) و ابر-شبکه (r)

Figure 7: Recursion between network (a) and hyper-network (r)

۱۲.۴. به علاوه، می‌توانیم عملگر ابر- m (M_Δ -ابر) $\langle m \rangle$ را بر اساس تحقیق لویین [۱۷] تعریف کنیم^{۲۹} که t^n را به $t^{\Delta n}$ (T_n را به $T_{\Delta n}$) و it^n را به $it^{\Delta n+\circ}$ (I_n را به $I_{\Delta n+\circ}$) می‌نگارد:

```
gap > hyp_m := m^alpha * ((t^4)^alpha);
[ (1,2,3,4,5,6,7,8,9,10,11,12), (1,11)(2,10)(3,9)(4,8)(5,7) ] ->
[ (1,6,11,4,9,2,7,12,5,10,3,8), (1,11)(2,10)(3,9)(4,8)(5,7) ]
```

همانند قبل، خروجی عمل ابر-عملگر بر مولدهای (دلخواه) گروه T/I یعنی t (T_1) و i ($I_\circ$) نشان می‌دهد که به ترتیب به سمت t^Δ (T_Δ) و i (I_Δ) نگاشته می‌شوند.

۱۳.۴. حال، عملگر ابر- mi را به صورت زیر تعریف کنید:

```
gap > hyp_mi := hyp_m * hyp_i;
[ (1,2,3,4,5,6,7,8,9,10,11,12), (1,11)(2,10)(3,9)(4,8)(5,7) ] ->
[ (1,8,3,10,5,12,7,2,9,4,11,6), (1,11)(2,10)(3,9)(4,8)(5,7) ]
```

همان‌طور که به وسیله‌ی مولدهای T/I در این خروجی نشان داده شده، این ابرعملگر، t^n را به $t^{\Delta n}$ و it^n را به $it^{\Delta n+\circ}$ می‌نگارد.

۱۴.۴. با این حال، کلمپن‌هاور [۱۲] مشاهده کرد که عمل عملگرهای $Hyp(T/I)$ لمبرت بر شبکه‌ها الزاماً با عمل T/I بر گروه‌های شبکه‌های هم‌نهشت تطابق ندارد. برای مثال، شبکه‌ای که با (b) در شکل ۸ برچسب زده شده است، به وسیله‌ی $\langle t^1 \rangle$ (ابر- T_1) به شبکه‌ای متصل می‌شود که با (c) برچسب زده شده است (در شکل ۴، متناظر با مجموعه‌های سه نغمه‌ای است که به ترتیب با “b” و “c” برچسب زده شده‌اند)، اما گروه‌های هم‌نهشت شبکه‌ها به وسیله‌ی t^Δ (T_Δ) به یکدیگر متصل می‌شوند نه t^1 . یعنی، نغمه‌ی هم‌نام ۵ در بالاترین گره سمت چپ (b) به وسیله‌ی t^Δ به نغمه‌ی هم‌نام ۱۰ در بالاترین گره سمت چپ (c) نگاشته می‌شود، اما این نغمه به وسیله‌ی t^1 به نغمه‌ی هم‌نام ۱۰ نگاشته نمی‌شود؛ همین روند برای سایر جفت گره‌های هم‌نهشت مرتب‌شده برقرار است. (شکل ۸ را ببینید). بنابراین، کلمپن‌هاور کاربرد ابر-عملگرهایی را به اثبات رساند که از گروه خودریختی T/I ، $Inn(T/I)$ به دست آمده‌اند،

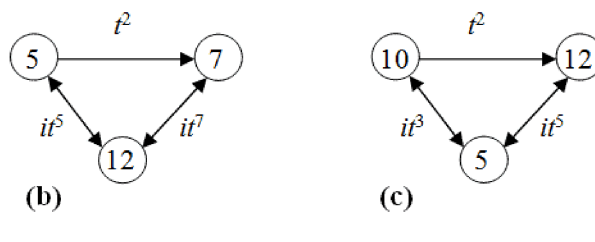
```
gap > InnTI := InnerAutomorphismsAutomorphismGroup(AutTI);
<group with 2 generators>
```

که دارای عمل طبیعی T/I بر خودش است. در اینجا، فرمان

“ $InnerAutomorphismsAutomorphismGroup(AutTI)$ ”

^{۲۹} تعمیم نظریه‌ی K -شبکه در مقاله لویین [۱۷، پیوست B] منتشر شده است. در اینجا، وی از نمادگذاری قبلی خود برای این خودریختی‌ها استفاده می‌کند: از این رو در اینجا، $F\langle \Delta, \circ \rangle$ و $F\langle \Delta, \circ \rangle$ به ترتیب نمایانگر عملگر ابر- m ($\langle m \rangle$) و عملگر ابر- mi ($\langle mi \rangle$) هست.

به GAP دستور می‌دهد تا زیرمجموعه‌ی خوردیختی‌های داخلی را در گروه خوردیختی $Aut(T/I)$ قرار دهد.



شکل ۸. شبکه‌ی (b) با t10 به (c) مربوط می‌شود، اما گره‌های همنهشت با t^5 مرتبط هستند.

Figure 8: Network (b) relates to (c) by t10, but congruent nodes relate by t^5 .

۱۵.۴. در حقیقت، خوردیختی‌های داخلی یک زیرگروه نرمال از گروه خوردیختی کامل، یعنی $Aut(T/I)$ شامل خوردیختی‌های حاصل از مزدوج‌ها توسط عناصر گروه هستند، که در آن مزدوج عنصر a با b برابر با ترکیب $b^{-1}ab$ است (همچنین، GAP از نمادگذاری نمایی برای مزدوج استفاده می‌کند؛ از این رو، a^b به این معنی است که a با b مزدوج می‌شود). از این رو، کلمپین‌هاور عملگرهای ابر- t یعنی، $[t^x]$ را در گروه توصیف می‌کند که اعضای T/I را با t^x (مزدوج با T_x) مزدوج می‌کنند^{۳۰}. چنین مزدوج‌هایی، t^n را به T_n (یا به T_n) $t^n(T_n)t^x = t^n$ و $t^{-x}(t^n)t^x = t^n$ را به it^n و $(it^x)^{-1}(t^n)it^x = t^n$ که $[it^x]$ یعنی، it^x عملگرهای ابر- t می‌نگارند. بنابراین، شبکه‌ی (b) در شکل ۸ به وسیله‌ی $[t^5]$ (مزدوج با T_5) به شبکه‌ی (c) وصل می‌شود که مطابق با عمل t^5 روی PCS است؛ در این عمل، گره‌های همنهشت به یکدیگر وصل می‌شدند:

```
gap > (t^2)^(t^5); (i*t^5)^(t^5);
(i*t^7)^(t^5); (1,3,5,7,9,11)(2,4,6,8,10,12)
(1,2)(3,12)(4,11)(5,10)(6,9)(7,8)
(1,4)(2,3)(5,12)(6,11)(7,10)(8,9)
```

خروجی، دورهای مربوط به عملگرهایی را نشان می‌دهد که پیکان‌ها را در (c) برچسب می‌زنند. همچنین، به کاربرد نمادگذاری نمایی در این فرمان (برای مثال، “ $(t^2)^{(t^5)}$ ”) توجه کنید که به مزدوج با $t^5(T_5)$ اشاره دارد.

۱۶.۴. اگر اندازه‌ی $Hyp(T/I)$ برابر با ۲۴ و اندازه‌ی $Inn(T/I)$ فقط برابر با ۱۲ باشد، آنگاه نمی‌توانیم یک یک‌ریختی بین این دو توصیف کنیم. با این حال، ممکن است یک هم‌ریختی گروه $beta$ را توصیف کنیم که $Hyp(T/I)$ را به $Inn(T/I)$ می‌نگارد:

```
gap > beta := ActionHomomorphism(HypTI, InnTI);
<action homomorphism>
```

خروجی این فرمان، صرفاً حضور یک هم‌ریختی را نشان می‌دهد. هسته‌ی $beta$ (آن دسته از عناصر $Hyp(T/I)$ که به عنصر همانی $Inn(T/I)$ نگاشته می‌شوند)، نابدیهی است:

^{۳۰}توجه داشته باشید که کلمپین‌هاور [۱۲]، گروه‌ها $([I])$ را برای ابر-عملگرهایی به کار گرفت که از گروه خوردیختی داخلی (یعنی، آن‌هایی که تحت مزدوج حاصل شده‌اند) به دست آمده‌اند، در مقابل، لوبین [۱۸]، قلاب‌های زاویه‌ای $\langle\langle\rangle\rangle$ را برای ابر-عملگرهایی استفاده کرد که از گروه $Hyp(T/I)$ به دست آمده‌اند.

```
gap > ker_beta := Kernel(beta);
```

```
<group with 1 generators>
```

```
gap > Size(ker_beta);
```

```
[2]
```

همان‌طور که خروجی نشان داد، این هسته شامل دو عنصر $Hyp(T/I)$ است. با استفاده از فرمان “List” پی می‌بریم که این دو عنصر شامل ابر-عملگرهای $\langle t^\circ \rangle$ و $\langle t^\epsilon \rangle$ (ابر- $T_\circ$ و ابر- T_ϵ) هستند.

```
gap > List(ker_beta);
```

```
[ IdentityMapping( Group([ (1,2,3,4,5,6,7,8,9,10,11,12),
```

```
(1,11)(2,10)(3,9)(4,8)(5,7) ] ),
```

```
[ (1,6,11,4,9,2,7,12,5,10,3,8), (1,10)(2,9)(3,8)(4,7)(5,6)(11,12) ] ->
```

```
[ (1,6,11,4,9,2,7,12,5,10,3,8), (1,4)(2,3)(5,12)(6,11)(7,10)(8,9) ] ]
```

ابتدا، $\langle t^\circ \rangle$ در خروجی به صورت “IdentityMapping(Group([1,2,3,4,5,6,7,8,9,10,11,12), (1,11)(2,10)(3,9)(4,8)(5,7))” ظاهر می‌شود که خودریختی T/I است و هر عنصر گروه را به خودش می‌نگارد؛ سپس، خروجی باقیمانده، $\langle t^\epsilon \rangle$ را بر حسب مولدهای T/I یعنی $t^5(T_5)$ و $it^{11}(I_{11})$ ارائه می‌دهد که به ترتیب به t^5 و $it^{11+\epsilon} = it^5$ (T_5 و I_5) نگاشته می‌شوند. توجه داریم که هسته با مرکز $Hyp(T/I)$ برابر است. این نکته اخیر، از ویژگی جابه‌جایی‌پذیری به دست آمده است: اعضای مرکز ($Zentrum$)^{۳۱} یک گروه عناصری هستند که با تمام عناصر گروه جابه‌جا می‌شوند. در این حالت، $\{ \langle t^\circ \rangle, \langle t^\epsilon \rangle \} = Z(Hyp(T/I))$. هنگامی که مزدوج هر عنصر گروه با عنصر دیگری که با آن جابه‌جا می‌شود، بدیهی است (از طریق خاصیت شرکت‌پذیری گروه‌ها)، هر دو عضو $Z(Hyp(T/I))$ به عضو همانی $Inn(T/I)$ نگاشته می‌شوند:

```
gap > ker_beta = Center(HypTI);
```

```
true
```

۱۷.۴. در نتیجه، نمی‌توانیم بین جایگشت‌های واقع در T/I تمایزی قائل شویم؛ این جایگشت‌ها یا از مزدوج با t° یا t^ϵ ($T_\circ$ یا T_ϵ) به دست آمده‌اند که در اینجا روی مولدهای t و i نشان داده شده‌اند:

```
gap > t^(t^0) = t^(t^6); i^(t^0) = i^(t^6);
```

```
true
```

```
true
```

یا با اعضای مختلف هر کلاس هم‌ارزی (چپ یا راست) هسته به دست آمده‌اند^{۳۲}:

```
gap > t^(t^1) = t^(t^7); i^(t^1) = i^(t^7);
```

```
true
```

```
true
```

```
gap > t^(i*t^4) = t^(i*t^10); i^(i*t^4) = i^(i*t^10);
```

```
true
```

```
true
```

و همین روند را برای هر کلاس هم‌ارزی دیگر داریم.

^{۳۱} به معنی مرکز به زبان آلمانی است؛ مترجم.

^{۳۲} از آنجایی که هسته زیرگروهی نرمال است (`gap > IsNormal(HypTI, ker_beta); true`)، پس مهم نیست که آن را در چپ یا راست برای تشکیل هم‌رده‌ها ضرب کنیم.

۵. نظریه‌ی نئوریسمانی

۱.۵. نظریه‌ی نئوریسمانی، حوزه‌ی دیگری از تحقیق موسیقی نظری است که GAP در آن شاخه کاربرد دارد. این نظریه به‌طور عمده با عمل‌های محتوایی مربوط به مجموعه‌های از گروه‌های سه‌تایی هم‌آوای کلانگ^{۳۳} سروکار دارد. برای این منظور، مجموعه‌ی سه‌تایی هم‌آوای دلخواه (گام ماژور (بزرگ) یا گام مینور (کوچک)) را به‌صورت یک مجموعه‌ی نغمه‌ی هم‌نام نامرتب تعریف کنید^{۳۵}:

$gap > CM := [4, 7, 12];$

$[4, 7, 12]$

مدار مجموعه‌ی نغمه‌ی هم‌نام را تحت T/I ، یک مجموعه‌ی رده (در این حالت، مجموعه‌ی نغمه‌های هم‌نام مربوط به سه‌تایی گام C ماژور تحت عناصر گروه T/I نگاشته می‌شود) بنامید و مجموعه‌ی رده‌ی گروه‌های سه‌تایی هم‌آوا را با K برچسب بزنید:

$gap > K := Orbit(TI, CM, OnSets);$

$[[4, 7, 12], [1, 5, 8], [5, 8, 12], [2, 6, 9], [4, 7, 11],$

$[1, 6, 9], [3, 7, 10], [3, 6, 10], [2, 7, 10], [3, 6, 11],$

$[4, 8, 11], [2, 5, 9], [3, 8, 11], [2, 5, 10], [5, 9, 12],$

$[1, 4, 8], [4, 9, 12], [1, 4, 9], [1, 6, 10], [3, 7, 12],$

$[1, 5, 10], [3, 8, 12], [2, 7, 11], [2, 6, 11]]$

مجموعه‌ی K به‌عنوان یک مجموعه‌ی دارای ۲۴ عضو است (۱۲ گروه سه‌تایی گام ماژور و ۱۲ گروه سه‌تایی گام مینور که تمام آن‌ها در بالا نمایش داده شده‌اند) که در آن T/I به‌صورت زیر عمل می‌کند^{۳۶}:

$gap > TI_K := Action(TI, K, OnSets);$

$Group([(1,2,4,7,11,15,19,23,22,18,14,10) (3,6,9,13,17,21,24,20,16,12,8,5),$

$(1,3)(2,5)(4,8)(6,10)(7,12)(9,14)(11,16) (13,18)(15,20)(17,22)(19,24)(21,23)]$

در خروجی مشاهده می‌کنیم که GAP، ۲۴ عضو از K را به اعداد صحیح [۲۴۰۰۰۱] می‌نگارد. در حقیقت، نگاشت K برای این اعداد صحیح دارای پایه‌ای در حافظه تعریف ارائه شده در بخش ۱۳.۳ است که برای گروه T/I تولیدشده به‌وسیله‌ی t و i (T_1 و I_0) آمده است. اگر اعداد مرتبه را به این ۲۴ مجموعه‌ی سه‌تایی واقع در خروجی فرمان قبلی اختصاص دهیم، آن‌گاه گام C ماژور [۴، ۷، ۱۲] برابر با ۱، گام C# ماژور [۱، ۵، ۸] برابر با ۲، گام F مینور [۵، ۸، ۱۲] برابر با ۳، گام D ماژور [۲، ۶، ۹] برابر با ۴ و گام E مینور [۴، ۷، ۱۱] برابر با ۵ خواهد بود. بنابراین، مولد t (T_1)، ۱ را به ۲، ۲ را به ۴ و ۴ را به ۷ می‌نگارد و این روند به‌همین صورت ادامه پیدا می‌کند؛ و i (I_0)، ۱ را به ۳، ۳ را به ۵ می‌نگارد و با همین روند جابه‌جایی را ادامه می‌دهیم. جدول ۲، فهرستی از این گروه‌های سه‌تایی و برچسب‌های متناظر آن‌ها را ارائه می‌دهد. عمل گروه T/I بر مجموعه‌های از ۲۰ نغمه‌ی هم‌نام، هم‌ریخت با عمل آن بر ۲۴ عضو K است که آن را $gamma$ یک‌ریختی می‌نامیم:

^{۳۳}مقاله لوبین [۱۶] شامل اولین کار عمده در نظریه‌ی نئوریسمانی است، هر چند جنبه‌های معینی از کار قبلی وی [۱۵] در این زمینه به‌کار رفته است. کوهن [۴] نیز، مقدمه‌ی بسیار خوبی را برای این نظریه پیشنهاد می‌کند؛ در واقع، شماره‌ی اصلی مجله‌ی

Neo-Riemannian Theory, *Journal of Music Theory* 42, no. 2 (1998)

به موضوعات نئوریسمانی اختصاص داده شده است (علاقه‌مندان می‌توانند به تارنمای <https://www.jstor.org/stable/i235025> مراجعه کنند: مترجم).

^{۳۵} کاربرد قبلی عدد صحیح ۱۲ را برای نغمه‌ی هم‌نام C در مطالعه‌ی فرمان بعدی فراخوانی کنید. به‌علاوه، نظریه‌ی نئوریسمانی معمولاً از نماد “+” برای مجموعه‌های سه‌تایی گام ماژور و از “-” برای مجموعه‌های سه‌تایی گام مینور استفاده می‌کند. نمی‌توانیم از این نمادها با نام‌های متغیر در GAP استفاده کنیم، در صورتی‌که GAP آن‌ها را برای عملگرهای حسابی در نظر می‌گیرد. بنابراین، از “M” برای گام ماژور استفاده می‌کنیم (برای مثال، “CM” در فرمان بعدی “گام C ماژور” را می‌خواند).

^{۳۶}مشابه با موقعیت ۴.۳، GAP نگاشت مختلفی از K را به [۲۴۰۰۰۱] در هنگام مطالعه‌ی “SampleGAPprogram.txt” اختصاص می‌دهد. در نتیجه، مرتب‌سازی متفاوتی را در این برنامه برای عمل‌های نئوریسمانی شریث s و w به‌دست می‌آوریم که در ۲.۵ ظاهر می‌شوند. بنابراین، مجدداً باید عمل‌های p ، r و l موجود در ۴.۵ را در این برنامه با $gap > p := w * s^9$ ، $gap > r := w$ ؛ و $gap > l := w * s^5$ تعریف کنیم. یک‌بار دیگر، موقعیت مشابهی در کتابچه‌ی راهنمای Sage “GAP-MTO” به‌دست می‌آید.

جدول ۲. نگاشت K به $[24 \dots 1]$

Table 2: Mapping of K to $[1 \dots 24]$

C+	$\mapsto$	۱	G-	$\mapsto$	۹
C#+	$\mapsto$	۲	Ab-	$\mapsto$	۱۳
D+	$\mapsto$	۴	A-	$\mapsto$	۱۷
Eb+	$\mapsto$	۷	Bb-	$\mapsto$	۲۱
E+	$\mapsto$	۱۱	B-	$\mapsto$	۲۴
F+	$\mapsto$	۱۵	C-	$\mapsto$	۲۰
F#+	$\mapsto$	۱۹	C#-	$\mapsto$	۱۶
G+	$\mapsto$	۲۳	D-	$\mapsto$	۱۲
Ab+	$\mapsto$	۲۲	Eb-	$\mapsto$	۸
A+	$\mapsto$	۱۸	E-	$\mapsto$	۵
Bb+	$\mapsto$	۱۴	F-	$\mapsto$	۳
B+	$\mapsto$	۱۰	F#-	$\mapsto$	۶

```
gap > gamma := IsomorphismGroups(TI, TI_K);
[ (1,2,3,4,5,6,7,8,9,10,11,12), (1,11)(2,10)(3,9)(4,8)(5,7) ] ->
[ (1,2,4,7,11,15,19,23,22,18,14,10)(3,6,9,13,17,21,24,20,16,12,8,5),
(1,8)(2,12)(3,14)(4,16)(5,10)(6,18)(7,20)(9,22)(11,24)(13,23)(15,21)(17,19) ]
```

تصویر t را تحت $gamma$ به صورت زیر:

```
gap > t_K := Image(gamma, t);
(1,2,4,7,11,15,19,23,22,18,14,10)(3,6,9,13,17,21,24,20,16,12,8,5)
```

و تصویر i تحت $gamma$ را به صورت زیر تعریف کنید:

```
gap > i_K := Image(gamma, i);
(1,8)(2,12)(3,14)(4,16)(5,10)(6,18)(7,20)(9,22)(11,24)(13,23)(15,21)(17,19)
```

۲۰۵. عمل T/I بر K ، نمایشی جایگشتی روی $[24 \dots 1]$ دارد که زیرگروهی از گروه متقارن روی آن مجموعه است ^{۳۷}:

```
gap > S_24 := SymmetricGroup(24);
Sym( [ 1 .. 24 ] )
gap > IsSubgroup(S_24, TI_K);
true
```

یک نتیجه کلاسیک در نظریه موسیقی تبدیلی [۱۶] بیان می‌کند که مرکزساز در S_{24} برای عمل T/I بر K برابر با گروه شریث/وشسل (S/W) نظریه‌ی نئوریمانی است ^{۳۸}:

^{۳۷} S_{24} ، یک گروه بزرگ از مرتبه $24!$ (620448401733239439360000) $(gap > Size(S_{24});)$ است. با این حال، GAP می‌تواند فرمان‌های بعدی را پردازش کند که این را تقریباً بلافاصله به‌کار می‌برد.
^{۳۸} مرکزساز در S_{24} برای عمل T/I بر K شامل تمام جایگشت‌های روی $[24 \dots 1]$ است که با هر عضو از T/I جابه‌جا می‌شوند.

```
gap > SW := Centralizer(S_24, TI_K);
Group([ (1,2,4,7,11,15,19,23,22,18,14,10) (3,5,8,12,16,20,24,21,17,13,9,6),
(1,3)(2,6)(4,9)(5,10)(7,13)(8,14)(11,17) (12,18)(15,21)(16,22)(19,24)(20,23) ])
```

در اینجا، خروجی یک جفت شریث و ویشل خاص را به عنوان مولدهای S/W نشان می‌دهد. این پیامد مربوط به نتیجه‌ی نظریه‌ی گروه جایگشتی است؛ این نظریه بیان می‌کند که از آنجایی که عمل T/I بر K منظم (به طور ساده، تقارنی) است،

```
gap > IsRegular(TI_K);
true
```

با مرکزساز آن یکریخت است [۶]. این یکریختی را δ بنامید:

```
gap > delta := IsomorphismGroups(TI_K, SW);
[ (1,2,4,7,11,15,19,23,22,18,14,10)(3,6,9,13,17,21,24,20,16,12,8,5),
(1,3)(2,5)(4,8)(6,10)(7,12)(9,14)(11,16)(13,18)(15,20)(17,22)(19,24)(21,23) ] ->
[ (1,2,4,7,11,15,19,23,22,18,14,10)(3,5,8,12,16,20,24,21,17,13,9,6),
(1,3)(2,6)(4,9)(5,10)(7,13)(8,14)(11,17)(12,18)(15,21)(16,22)(19,24)(20,23) ]
```

توجه داریم که δ ، T/I را به S/W می‌نگارد به طوری که GAP، عبارات خروجی مولدها را برای گروه‌های مربوطه در هنگام عمل آن‌ها بر K نشان می‌دهد.

۳.۵ s را (شریث واحد) تصویر t تحت δ بنامید:

```
gap > s := Image(delta, t_K);
(1,2,4,7,11,15,19,23,22,18,14,10)(3,5,8,12,16,20,24,21,17,13,9,6)
```

این، گروه‌های سه‌تایی گام ماژور را به وسیله‌ی یک نیم‌پرده (گام C ماژور، گام $C\#$ ماژور و غیره) به سمت بالا و گروه‌های سه‌تایی گام مینور را به وسیله‌ی همان مقدار (گام C مینور، گام b مینور و غیره) به پایین می‌نگارد. علاوه بر این، w را تصویر i تحت δ بنامید:

```
gap > w := Image(delta, i_K);
(1,9)(2,13)(3,14)(4,17)(5,18)(6,10)(7,21)(8,22)(11,24)(12,23)(15,20)(16,19)
```

این نگاشت در جدول ۲ ذکر شده است. توجه داریم که w یک ویشل است. این، هر مجموعه‌ی سه‌تایی گام ماژور را به یک مجموعه‌ی سه‌تایی گام مینور می‌فرستد و بالعکس که در آن ریشه‌های سه‌گانه‌ی مربوطه یک بازه‌ی باوفا را تشکیل می‌دهند: ۱ به ۹ (گام C ماژور را به گام G مینور) و ۲ به ۱۳ (گام $C\#$ ماژور را به گام Ab مینور [گام $G\#$ مینور]) می‌نگارد و این روند به همین صورت ادامه پیدا می‌کند. بنابراین، s و ws^2 برجسب‌های شریث و ویشل خواهند بود که در ورودی این فرمان ظاهر می‌شوند که این فرمان، S/W موجود در ۲.۵ را تعریف می‌کند.

۴.۵. نئوریسمانی متعارفی که با مهمترین صدای کوتاه (پرده موازی، پرده نسبی و پرده برجسته) تعویض می‌شود را می‌توان به صورت زیر تعریف کرد:

```
gap > p := w * s^7;
(1,20)(2,16)(3,15)(4,12)(5,11)(6,19)(7,8)(9,23)(10,24)(13,22)(14,21)(17,18)
gap > r := w * s^10;
(1,17)(2,21)(3,22)(4,24)(5,23)(6,18)(7,20)(8,19)(9,14)(10,13)(11,16)(12,15)
gap > l := w * s^3;
(1,5)(2,3)(4,6)(7,9)(8,10)(11,13)(12,14)(15,17)(16,18)(19,21)(20,22)(23,24)
```

به علاوه، عمل‌های مفید دیگری مثل قطب هگزاتونیک کوهن [۵] ممکن است به صورت ترکیبی از این تعویض‌ها تعریف شوند:

```
gap > hp := l * p * l;
(1,13)(2,17)(3,18)(4,21)(5,22)(6,14)(7,24)(8,23)(9,10)(11,20)(12,19)(15,16)
```

۵.۵. حال، گروه T دوری را بسازید که روی گروه‌های سه‌تایی هم‌آوا عمل می‌کند:

```
gap > T_K := Group(t_K);
Group([ (1,2,4,7,11,15,19,23,22,18,14,10) (3,6,9,13,17,21,24,20,16,12,8,5) ])
```

دوباره، خروجی دورهای مولد t (T_1) در هنگام عمل آن بر K نشان می‌دهد. مرکز ساز در S_{24} برای عمل T بر K ، گروه معروف دیگری در نظریه‌ی نئوریمانی به نام مجموعه‌ی ۲۸۸ انتقال سه‌تایی یکنواخت (UTT های) ^{۳۹} هوک [۹] است:

```
gap > CinS_24ofT_K := Centralizer(S_24, T_K);
Group([ (1,2,4,7,11,15,19,23,22,18,14,10) (3,6,9,13,17,21,24,20,16,12,8,5),
(3,5,8,12,16,20,24,21,17,13,9,6), (1,3)(2,6)(4,9)(5,10)(7,13)(8,14)(11, 17)
(12,18)(15,21)(16,22)(19,24)(20,23) ])
gap > Size(CinS_24ofT_K);
[288]
```

که S/W زیرگروهی از آن است:

```
gap > IsSubgroup(CinS_24ofT_K, SW);
true
```

۶.۵. می‌توانیم یک ساختار یک‌ریخت را روی $[24 \dots 1]$ به‌دست آوریم که احتمالاً شهودی‌تر است. فرض کنیم $[12 \dots 1]$ نغمه‌های هم‌نامی را نشان می‌دهد که به‌عنوان ریشه‌های مجموعه‌های سه‌تایی گام مازور عمل می‌کنند و فرض کنید $[24 \dots 13]$ ریشه‌های مجموعه‌های سه‌تایی گام مینور را نشان می‌دهد که در آن x و $x+12$ به همان نغمه‌ی هم‌نام تعلق دارند (یعنی، ۱ و ۱۳ به نغمه‌ی هم‌نام $C\#$ ، ۲ و ۱۴ به نغمه‌ی هم‌نام D ، ۱۲ و ۲۴ و $\dots$ به نغمه‌ی هم‌نام C نگاشته می‌شوند، جدول ۳ را ببینید).

جدول ۳. نگاشت جایگزین K به $[24 \dots 1]$

Table 3: Alternate mapping of K to $[1 \dots 24]$

$C\#+$	$\mapsto$	۱	$C\#-$	$\mapsto$	۱۳
$D+$	$\mapsto$	۲	$D-$	$\mapsto$	۱۴
$Eb+$	$\mapsto$	۳	$Eb-$	$\mapsto$	۱۵
$E+$	$\mapsto$	۴	$E-$	$\mapsto$	۱۶
$F+$	$\mapsto$	۵	$F-$	$\mapsto$	۱۷
$F\#+$	$\mapsto$	۶	$F\#-$	$\mapsto$	۱۸
$G+$	$\mapsto$	۷	$G-$	$\mapsto$	۱۹
$Ab+$	$\mapsto$	۸	$Ab-$	$\mapsto$	۲۰
$A+$	$\mapsto$	۹	$A-$	$\mapsto$	۲۱
$Bb+$	$\mapsto$	۱۰	$Bb-$	$\mapsto$	۲۲
$B+$	$\mapsto$	۱۱	$B-$	$\mapsto$	۲۳
$C+$	$\mapsto$	۱۲	$C-$	$\mapsto$	۲۴

^{۳۹} مخفف Uniform triadic transformations: مترجم

۷.۵. UTT های هوک به شکل $\langle \sigma, t^+, t^- \rangle$ ارائه شده‌اند که در آن $\sigma \in \{+, -\}$ که نمای $(+)$ الگو را حفظ و $(-)$ آن را معکوس می‌کند.^{۴۰}

```
gap > minus := (1, 13)(2, 14)(3, 15)(4, 16)(5, 17)(6, 18)
> (7, 19)(8, 20)(9, 21)(10, 22)(11, 23)(12, 24);
(1, 13)(2, 14)(3, 15)(4, 16)(5, 17)(6, 18)(7, 19)(8, 20)(9, 21)(10, 22)(11, 23)(12, 24)
gap > plus := minus^2;
()
```

بنابراین، t^+ عامل ترانهش (انتقال) را روی T -مدار مجموعه‌های سه‌تایی گام ماژور نشان می‌دهد:

```
gap > t^plus;
(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
```

که در آن t (T_1 روی $[120001]$ عمل می‌کند) را با عمل “جمع” مزدوج می‌کنیم؛ و t^- روی مجموعه‌های سه‌تایی گام مینور

```
gap > t^minus;
(13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24)
```

ارائه شده توسط مزدوج کردن t با عمل “منها” عمل می‌کند که $[120001]$ را همانند یک مجموعه به $[2400013]$ جابه‌جا می‌کند.

۸.۵. ممکن است UTT ها را مانند رابطه‌ی غالب هایلر $[10]$ $d = \langle +, 5, 5 \rangle$ و قطب نزدیک به شش تونیک اسمیت $[23]$ $nhp = \langle -, 3, 9 \rangle$ در GAP به صورت زیر مدل‌سازی کنیم:

```
gap > d := (t^plus)^5 * (t^minus)^5 * plus;
(1, 6, 11, 4, 9, 2, 7, 12, 5, 10, 3, 8)(13, 18, 23, 16, 21, 14, 19, 24, 17, 22, 15, 20)
gap > nhp := (t^plus)^3 * (t^minus)^9 * minus;
(1, 16)(2, 17)(3, 18)(4, 19)(5, 20)(6, 21)(7, 22)(8, 23)(9, 24)(10, 13)(11, 14)(12, 15)
```

خروجی اولین فرمان، ۱ را در یک مدار به ۶ و سپس به ۱۱ و ۱۳ را به ۱۸ و سپس به ۲۳ در مدار دیگر می‌نگارد. این اعداد صحیح در دستگاه برچسب‌گذاری جدید به گام $C\#$ ماژور، گام $F\#$ ماژور، گام B ماژور در اولین دور و گام $C\#$ مینور، گام $F\#$ مینور و گام B مینور در دور دیگر اشاره دارند. در عوض، خروجی فرمان دوم مبادلات دوازده را نشان می‌دهد: ۱ به ۱۶ نگاشته می‌شود و ۱۶ به ۱ برمی‌گردد؛ ۲ به ۱۷ نگاشته می‌شود و ۱۷ به ۲ برمی‌گردد؛ و الی آخر. مجدداً، در هر دستگاه برچسب‌گذاری جدید توجه داریم که گام $C\#$ ماژور با گام E مینور، گام D ماژور با گام F مینور مبادله می‌شود و الی آخر.

۹.۵. به علاوه، هوک $[9]$ در سال ۲۰۰۲ یک گروه بزرگتر به نام QQT را توصیف کرد که عمل انعکاس معمولی را به UTT ها متصل می‌کند. می‌توانیم این گروه را به صورت ضرب حلقوی T/I تشکیل دهیم که بر $[120001]$ به وسیله‌ی S_2 (گروه متقارن درجه‌ی ۲ که شامل دو جایگشت $()$ و $(1, 2)$ است) عمل می‌کند:

```
gap > QTT := WreathProduct(TI, SymmetricGroup(2));
<permutation group of size 1152 with 5 generators>
```

QTT ، چندین زیرگروه آهنگین مربوطه از مرتبه‌ی ۲۴ دارد که شامل گروه S/I آبلی کلاف $[3]$ هستند.^{۴۱}

^{۴۰} عمل تفاضل ($minus$) که در فرمان بعدی تعریف شده، در اصل مبادله‌ای موازی است؛ توان‌یابی و بازگشتی است که $C\# +$ را به $C\# - (1, 13)$ ، $D +$ را به $D - (2, 14)$ ، ... و $C +$ را به $C - (12, 24)$ می‌نگارد و بالعکس. بنابراین، عمل جمع ($plus$) واضح است و در اینجا تنها برای ارائه‌ی سازگاری در برچسب‌گذاری با نمادگذاری هوک $[9]$ داده می‌شود.

^{۴۱} کلاف $[2]$ ، گروهی را ارائه می‌دهد که روی مجموعه‌های سه‌تایی هم‌آوای تولید شده با یک شریث واحد و یک انعکاس عمل می‌کند و “گروه S/I ” را برچسب می‌زند. از آنجایی که این گروه شامل دوازده شریث آشنا نیست، پس شامل تمام دوازده انعکاس نیست. (واضح است که دو انعکاس برای تشکیل یک ترانهش مقایسه می‌شوند نه برای تشکیل یک شریث.) به همین منوال، یک “گروه T/W ” یک‌ریخت را توصیف کرد. هوک $[9]$ ، این گروه‌ها را در موضوع گروه QTT بررسی کرد.

```
gap > schritt := t^plus * (t^minus)^-1 * plus;
(1,2,3,4,5,6,7,8,9,10,11,12)(13,24,23,22,21,20,19,18,17,16,15,14)
gap > inversion := i^plus * i^minus * minus;
(1,23)(2,22)(3,21)(4,20)(5,19)(6,18)(7,17)(8,16)(9,15)(10,14)(11,13)(12,24)
gap > SI := Group(schritt, inversion);
Group([ (1,2,3,4,5,6,7,8,9,10,11,12)(13,24,23,22,21,20,19,18,17,16,15,14),
(1,23)(2,22)(3,21)(4,20)(5,19)(6,18)(7,17)(8,16)(9,15)(10,14)(11,13)(12,24) ])
gap > IsAbelian(SI);
true
```

۱۰.۵. کوچاوی [۱۳] علاوه بر عمل انعکاس معمولی، پنج عملگر انعکاس همبافتی (cois) را توصیف کرد که می‌توانند بر مجموعه‌های از مجموعه‌های سه‌تایی هم‌آوا عمل کنند^{۲۳}:

```
gap > i215 := m^plus * m^minus * minus;
(1,17)(2,22)(3,15)(4,20)(5,13)(6,18)(7,23)(8,16)(9,21)(10,14)(11,19)(12,24)
gap > i216 := (m * i)^plus * (m * i)^minus * minus;
(1,19)(2,14)(3,21)(4,16)(5,23)(6,18)(7,13)(8,20)(9,15)(10,22)(11,17)(12,24)
gap > i217 := i^plus * i^minus * (t^minus)^6 * minus;
(1,23,7,17)(2,22,8,16)(3,21,9,15)(4,20,10,14)(5,19,11,13)(6,18,12,24)
gap > i218 := (m * i)^plus * (m * i)^minus * (t^minus)^6 * minus;
(1,19,7,13)(2,14,8,20)(3,21,9,15)(4,16,10,22)(5,23,11,17)(6,18,12,24)
gap > i219 := m^plus * m^minus * (t^minus)^3 * minus;
(1,17,4,20,7,23,10,14)(2,22,5,13,8,16,11,19)(3,15,6,18,9,21,12,24)
```

که cois ممکن است با t (که بر [۲۴۰۰۰۱] عمل می‌کند) برای ساخت گروه‌های غیرآبلی مرتبه‌ی ۲۴ همراه با یک عنصر مرتبه‌ی ۱۲ استفاده شود:

```
gap > TI215 := Group(t^plus * t^minus, i215);
Group([ (1,2,3,4,5,6,7,8,9,10,11,12)(13,14,15,16,17,18,19,20,21,22,23,24),
(1,17)(2,22)(3,15)(4,20)(5,13)(6,18)(7,23)(8,16)(9,21)(10,14)(11,19)(12,24) ])
gap > TI216 := Group(t^plus * t^minus, i216);
Group([ (1,2,3,4,5,6,7,8,9,10,11,12)(13,14,15,16,17,18,19,20,21,22,23,24),
(1,19)(2,14)(3,21)(4,16)(5,23)(6,18)(7,13)(8,20)(9,15)(10,22)(11,17)(12,24) ])
gap > TI217 := Group(t^plus * t^minus, i217);
Group([ (1,2,3,4,5,6,7,8,9,10,11,12)(13,14,15,16,17,18,19,20,21,22,23,24),
(1,23,7,17)(2,22,8,16)(3,21,9,15)(4,20,10,14)(5,19,11,13)(6,18,12,24) ])
gap > TI218 := Group(t^plus * t^minus, i218);
Group([ (1,2,3,4,5,6,7,8,9,10,11,12)(13,14,15,16,17,18,19,20,21,22,23,24),
(1,19,7,13)(2,14,8,20)(3,21,9,15)(4,16,10,22)(5,23,11,17)(6,18,12,24) ])
```

^{۲۲} مخفف Contextual inversion operators: مترجم

^{۲۳} برچسبگذاری انعکاس‌های مرسوم کوچاوی [۱۳] ($i_{215}, i_{216}, i_{217}, i_{218}$) از شمارگذاری شکل‌های ارائه‌شده در پایان‌نامه‌ی وی (شکل‌های ۱۵.۲، ۱۶.۲، ...، ۱۹.۲) نتیجه شده است (و در غیر این صورت، برچسبگذاری نشده‌اند).

```
gap > TI219 := Group(t^plus * t^minus, i219);
Group([ (1,2,3,4,5,6,7,8,9,10,11,12)(13,14,15,16,17,18,19,20,21,22,23,24),
(1,17,4,20,7,23,10,14)(2,22,5,13,8,16,11,19)(3,15,6,18,9,21,12,24) ])
```

حال، می‌توانیم (تا یک‌ریختی) هر یک از این گروه‌ها را با استفاده از کتابخانه‌ی گروه کوچک GAP مشخص کنیم (جدول ۴ را ببینید):^{۴۴}(۴۴)

```
gap > IdGroup(TI);
[ 24, 6 ]
gap > IdGroup(TI215);
[ 24, 5 ]
gap > IdGroup(TI216);
[ 24, 10 ]
gap > IdGroup(TI217);
[ 24, 4 ]
gap > IdGroup(TI218);
[ 24, 11 ]
gap > IdGroup(TI219);
[ 24, 1 ]
```

جدول ۴. طبقات یک‌ریختی زیرگروه‌های غیرآبلی درجه ۲۴ در M با عنصر درجه ۱۲

Table 4: Isomorphism classes of the nonabelian subgroups of order 24 in M with an element of order 12

T/I	$\cong$	$[24, 6]$	$\cong$	D_{24}
T/I_{215}	$\cong$	$[24, 5]$	$\cong$	$D_8 \times C_3$
T/I_{216}	$\cong$	$[24, 10]$	$\cong$	$D_8 \times C_3$
T/I_{217}	$\cong$	$[24, 4]$	$\cong$	Dic_8
T/I_{218}	$\cong$	$[24, 11]$	$\cong$	$Q_8 \times C_3$
T/I_{219}	$\cong$	$[24, 1]$	$\cong$	$C_3 \rtimes C_8$

^{۴۴} در جدول ۴، نماد " Dic_8 " به گروه دودوری مرتبه‌ی ۲۴ (Q_8) ، به گروه چهارگانی مرتبه‌ی ۸ و " $C_3 \rtimes C_8$ " به ضرب نیمه‌مستقیم C_3 در C_8 اشاره دارد.
^{۴۵} خروجی فرمان `gap > IdGroup(TI)` مقدار $[24, 6]$ را نشان می‌دهد که به معنی این است که T/I برای ششمین گروه فهرست‌شده در رده بندی GAP پانزده گروه مرتبه‌ی ۲۴ یک‌ریخت است. کتابخانه‌ی گروه کوچک برای Sage نصب نشده است؛ از این رو، هفت فرمان آخر ارائه شده در اینجا در کتابچه‌ی راهنمای Sage "GAP-MTO" گنجانده نشده‌اند.

۱۱.۵. با وجود این که QTT شامل زیرگروه‌هایی است که برای تمام گروه‌های مرتبه‌ی ۲۴ با یک عنصر مرتبه‌ی ۱۲ یک‌ریخت هستند^{۴۶}، اما شامل این زیرگروه‌های خاص نیست. بلکه، دوت و پک [۷] گروهی از مرتبه‌ی ۴۶۰۸ (M) را ارائه دادند که شامل $cois$ کوچاوی نبود^{۴۷}. M یک ضرب حلقوی از TTO است که به وسیله‌ی S_2 بر $[120001]$ عمل می‌کند:

```
gap > M := WreathProduct(t(TTO, SymmetricGroup(2));
<permutation group of size 4608 with 7 generators>
```

۱۲.۵. دو گروه آبدلی مرتبه‌ی ۲۴ با یک عنصر مرتبه‌ی ۱۲ نیز به عنوان زیرگروه‌های M ظاهر می‌شوند. گروه S/I کلاف (از بخش ۹.۵ بالا) یک مثال است:

```
gap > IdGroup(SI);
[ 24, 9 ]
```

و رده یک‌ریخت آن به صورت

$$S/I \cong [24, 9] \cong C_{12} \times C_2$$

است. حالت باقیمانده سمت چپ به عنوان تمرین به خواننده واگذار می‌شود^{۴۸}.

۶. نتیجه‌گیری

۱.۶. هدف بحث پیش رو، نمایش کارایی GAP در مطالعه‌ی نظریه‌ی موسیقی تبدیلی است. با وجود این که بسیاری از مثال‌ها، مدل‌های مشهوری را بازسازی کردند، اما هدف ما برای جست‌وجوی نتایج جدید نبوده است. با این حال، GAP ممکن است در پیگیری دستگاه‌های نظری جدید به طور کاملاً کارآمدی عمل کند. به علاوه، می‌تواند ابزار ارزشمندی در تعلیم نظریه‌ی انتقال باشد که به معلمان و دانش‌آموزان برای تهیه محیطی برای آزمایش با مفاهیم موجود یا اصلی کمک می‌کند.

۲.۶. علاوه بر موضوعات گروه نظری استاندارد که تا اینجا مطرح کردیم، GAP کاربردهای فراوان دیگری نیز دارد. GAP می‌تواند برای مطالعه‌ی سایر ساختارهای جبری گسسته مانند تکواره‌ها، نیم‌گروه‌ها، میدان‌ها، حلقه‌ها، فضاها، برداری، ماتریس‌ها و موارد مشابه استفاده شود. یکی از کارهایی که GAP قادر به انجام آن نیست (حداقل به طور کارآمد نمی‌تواند انجام دهد)، تحقیقات در فضاها پیوسته و گسسته است؛ برای مشاهده چنین موضوعی می‌توانید به [۲] مراجعه کنید. همچنین، این موضوعات ممکن است به طور محاسباتی با استفاده از برنامه‌هایی به جز GAP بررسی شوند.

مراجع

- [1] M. Aschbacher, *Finite Group Theory*, Cambridge: Cambridge University Press, 1986.
- [2] C. Callender, I. Quinn and D. Tymoczko, Generalized Voice Leading Spaces, *Science*, **320** (2008) 346–348.
- [3] J. Clough, A Rudimentary Geometric Model for Contextual Transposition and Inversion, *J. Music Theory*, **42** no. 2 (1998) 297–306.
- [4] R. Cohn, An Introduction to Neo-Riemannian Theory: A Survey and Historical Perspective, *J. Music Theory*, **42** no. 2 (1998) 167–180.

^{۴۶}پانزده گروه مرتبه‌ی ۲۴ (تا یک‌ریخت) وجود دارند، اما هفت‌تای آن‌ها دارای یک عنصر مرتبه‌ی ۱۲ نیستند که در صورت مدل‌سازی یک عمل مانند ترانهش در این گروه‌ها الزامی است.
^{۴۷}در تحقیق دوت و پک [۷]، M "گروه مادر" را نشان می‌دهد. با عمل M اشاره شده در موريس [۱۹] اشتباه گرفته نشود.
^{۴۸}از سوی دیگر، می‌توانید به [۹، قضیه ۸.۳] مراجعه کنید.

- [5] R. Cohn, Uncanny Resemblances: Tonal Signification in the Freudian Age, *J. A. Music Sos.*, **57(2)** (2004) 285–323.
- [6] J. D. Dixon and B. Mortimer, *Permutation Groups*, New York: Springer-Verlag, 1996.
- [7] J. Douthett and R. Peck, An Order 1152 Group of Triadic Transformations and Its Relevance to Existing Musical Theoretical Structures, Paper presented at the Special Session on Mathematical Techniques in Musical Analysis, American Mathematical Society/Mathematical Association of America Joint National Meeting, New Orleans, Louisiana, 2007.
- [8] The GAP Group, *GAP—Groups, Algorithms, and Programming, Version 4.4.12*, 2008. <https://www.gap-system.org>
- [9] J. Hook Uniform Triadic Transformations, *J. Music Theory*, **46(1–2)** (2002) 57–126.
- [10] B. Hyer, Reimag(in)ing Riemann, *J. Music Theory*, **39(1)** (1995) 101–138.
- [11] H. Klumpenhouwer, *A Generalized Model of Voice-Leading for Atonal Music*, Ph.D. diss., Harvard University, 1991.
- [12] H. Klumpenhouwer, The Inner and Outer Automorphisms of Pitch-Class Inversion and Transposition: Some Implications for Analysis with Klumpenhouwer Networks. *Intégral*, **12** (1998) 81–93.
- [13] J. H. Kochavi, *Contextually Defined Musical Transformations*, Ph.D. diss., State University of New York, Buffalo, 2002.
- [14] P. Lambert, Isographies and Some Klumpenhouwer Networks They Involve, *Music Theor. Spectrum*, **24(2)** (2002) 165–195.
- [15] D. Lewin, Amfortas's Prayer to Titurél and the Role of D in Parsifal: The Tonal Spaces of the Drama and the Enharmonic C-flat/B, *Ninet-Cent. Music Rev.* **8(3)**, (1984) 336–349.
- [16] D. Lewin, *Generalized Musical Intervals and Transformations*, New Haven: Yale University Press, 1987.
- [17] D. Lewin, Klumpenhouwer Networks and Some Isographies that Involve Them, *Music Theor. Spectrum*, **12(1)** (1990) 83–120.
- [18] D. Lewin, A Tutorial on Klumpenhouwer Networks, Using the Chorale in Schoenberg's Opus 11, No. 2, *J. Music Theory*, **38(1)** (1994) 79–101.
- [19] R. D. Morris, Set Groups, Complementation, and Mappings among Pitch-Class Sets, *J. Music Theory*, **26(1)** (1982) 101–144.
- [20] R. D. Morris, *Composition with Pitch-Classes*, New Haven: Yale University Press, 1987.
- [21] R. Peck, *GAP (Groups, Algorithms, and Programming): A Tool for Computer-Assisted Research in Music Theory*, Poster presented at Society for Music Theory National Meeting, Cambridge, Massachusetts, 2005.
- [22] R. Peck, Wreath Products in Transformational Music Theory, *Perspect. New Music*, **47(1)** (2009) 193–210.
- [23] D. Smyth, More About Wagner's Chromatic Magic, *31st Annual Meeting of Society for Music Theory*, Nashville, Tennessee, 2008.
- [24] D. V. Starr, Sets, Invariance, and Partitions, *J. Music Theory*, **22(1)** (1978) 136–183.
- [25] D. Starr and Robert Morris, The Structure of All-Interval Series, *J. Music Theory*, **18(2)** (1974) 364–389.

- [26] D. Starr and R. Morris. A General Theory of Combinatoriality and the Aggregate, *Perspect. New Music*, **16(1)** 3-35 (1977); and **16(2)** (1977) 50-84.
- [27] W. Stein, et al. *Sage Mathematics Software, Version 3.4*, The Sage Development Team, 2009. <http://www.sagemath.org/>

مهدی شمس

گروه آمار، دانشکده علوم ریاضی، دانشگاه کاشان، کاشان، ایران
mehdishams@kashanu.ac.ir

مهدی شمس متولد آذر ماه ۱۳۵۷ در شهر اصفهان است. وی در سال ۱۳۷۵ وارد مقطع کارشناسی رشته ریاضی کاربردی در دانشگاه اصفهان شد. در سال ۱۳۸۰ وارد مقطع کارشناسی ارشد رشته آمار محض در دانشگاه صنعتی اصفهان شد. وی در سال ۱۳۸۷ وارد مقطع دکتری رشته آمار، گرایش استنباط آماری در دانشگاه فردوسی مشهد شد. او از سال ۱۳۹۲ تاکنون عضو هیات علمی گروه آمار دانشگاه کاشان است.



سیدعلی محمدیه

گروه ریاضی محض، دانشکده علوم ریاضی، دانشگاه کاشان، کاشان، ایران
alim@kashanu.ac.ir, max@std.kashanu.ac.ir

سید علی محمدیه متولد خرداد ماه ۱۳۷۹ در شهر کاشان است. وی در سال ۱۴۰۳ در مقطع کارشناسی رشته ریاضیات و کاربردها از دانشگاه کاشان فارغ التحصیل شده است. او پانزده سال است که در زمینه برنامه نویسی و کار با رایانه فعالیت دارد و از سال ۱۴۰۰ تاکنون به عنوان مدیر فنی در شرکت نرم افزاری خود مشغول به کار است.

