Sigma J Eng & Nat Sci 9 (1), 2018, 107-125

Publications Prepared for the 13th Algebraic Hyperstructures and its Applications Conferences AHA 2017

Research Article

COMPUTATION OF FUNDAMENTAL GROUP OF A FINITE HYPERGROUP

Karim ABBASI[1], Reza AMERI*[2], Yahya TALEBI-ROSTAMI[3]

[1]*Department of Mathematics, University of Mazandaran, Babolsar-IRAN*; ORCID:0000-0003-2135-8864
[2]*School of Mathematics, University of Tehran, Tehran-IRAN;* ORCID:0000-0001-5760-1788
[3]*Department of Mathematics, University of Mazandaran, Babolsar-IRAN*; ORCID:0000-0003-2311-4628

Received: 01.01.2018 Accepted: 16.04.2018

ABSTRACT

Let (H, o) be a hypergroup and β^* be the fundamental relation on *H*, that is, the smallest equivalence relation on *H* such that the quotient $(H/\beta^*, \otimes)$ is a group. The purpose of this paper is to compute the fundamental group of a given finite hypergroup. In this regards, first obtain some algebraic results to obtain equivalence classes of the fundamental relation and the we introduce an algorithm to compute these classes. Also, based on these algorithms we develop an application to construct the equivalence classes of β^* and hence to compute the fundamental group, $(H/\beta^*, \otimes)$. Furthermore, we use a sub-program to produce all hypergroups (up to isomorphism) of order less than or equal 3 and obtain their fundamental groups. Finally, we examine the algorithm and program by some examples to compute the fundamental groups of hypergroups of various orders.
Keywords: Hypergroup, fundamental relation, fundamental group, algorithm, computation**.**
MSC number: 20N20, 68W30, 68W40.

1. INTRODUCTION

The theory of algebraic hyperstructures was born in 1934, when the notion of a hypergroup was defined by Marty [12], as a generalization of group. After that, several papers have been written (for example see [2], [5], [13], [15], [16], [17]) in order to construction of finite hypergroups. Since hypergroups are much more varied than groups, e.g. for the prime number 3, there is only one group, up to isomorphism, with cardinality 3, while there are 3999 non-isomorphic hypergroups with 3 elements. As it is well-known one of the main topics in study of hypergroup theory is the fundamental relation β^*, in fact this relation plays an important role in this theory. The aim of this paper is to find an application to compute the fundamental group of an arbitrary finite hypergroup. For this, we obtain two algorithms and then develop an application to compute the equivalence classes of β^* and the fundamental group, $(H/\beta^*, \otimes)$. Also, we illustrate these algorithms by some various examples.

This paper has been written in 3 sections. In section 2, we obtain some results and give two algorithms (Algorithms 2.5 and 2.7) to compute the equivalence classes of β^* and the fundamental group, $(H/\beta^*, \otimes)$. In section 3, we develop a comprehensive program in Java

* Corresponding Author: e-mail: rameri@ut.ac.ir, tel: +989122805056

programming environment to obtain the fundamental group of a given finite hypergroup as well as to produce all finite hypergroups of order less than or equal 3 with their fundamental groups. In the sequel of this section we briefly give some notions and results of hypergroup theory that we need to develop our paper (for more details see [3], [4]).

Let H be a non- empty set and $P^*(H)$ denotes the set of all non-empty subsets of H. A *hyperoperation* on H is a map o : $H \times H \to P^*(H)$. A couple (H , o) is said to be a *hypergroupoid*. Let A, B be subsets of H. The hyperproduct A o B is defined as

$$A \circ B = \bigcup_{(a,b)\in A\times B} a \circ b.$$

Also if there is no confusion, $\{a\}$, A o $\{a\}$ fag and $\{a\}$ o A are denoted by a, A o a and a o A, respectively. From now on, when there is no ambiguity, we write ab instead of a o b for $a, b \in H$. A hypergroupoid (H, o) is said to be a *semihypergroup* if o is associative. A hypergroupoid (H, o) is called *quasihypergroup* if for all $x \in H$, we have xoH =Hox= H (reproductivity). We say that a hypergroupoid (H, o) is a *hypergroup* if (H, o) verifies both associativity and reproductivity. Let (H, o) is a finite hypergroup. We define the relation $\beta = \cup_{n\geq 1} \beta_n$, in which $\beta_1 = \{(a, a) \mid a \in H\}$ and for all $(a, b) \in H^2$, $a\beta b$ iff $\{a, b\} \subseteq \Pi_{mi}=1\ Z_i$, for some $z_1, z_2, \ldots , z_m \in H$ and $m \in$ N. Indeed, for all $a, b \in H$, we have $a\beta b$ iff $z_1 = a\ \beta z2\beta \ldots {}_{zn-1}\ \beta_{zm} = b$, for some $z_1, z_2, \ldots z_m \in H^n$. The relation β was introduced in 1970 by Koskas [11] and was mainly studied by Corsini [4] and Vougiouklis [18] and others. A usefull tool in the study of theory of hyperstractures is the relation β^*, which is defined in a hypergroup or semi-hypergroup (H, o), as the smallest equivalence relation on H such that the quotient H/β^* be a group (semi-group) under the hyperoperation $\beta^*(a) \otimes \beta^*(b) = \beta^*(c)$, $c \in a$ o b. This relation named as *fundamental relation* and (H/β^*, ⊗) is called *fundamental group*. It is seen that, in hypergroups, $\beta^* = \beta$^ , in which β^ is the transitive closure of β. In [10], Freni proved that β is transitive on hypergroups. Therefore, $\beta^* = \beta$^. In [1], Ameri introduced a process to obtain a group via the fundamental relation. The equivalence relation β^*(or its extended forms in hyperrings and hypermodoules and hyperlagebras) also was studied in many other papers(for example see [3], [6], [7], [8], [9], and also [14], [17], [18]).

Definition 1.1 Let (H, o) be a hypergroup and a o $b = H$, for all $a, b \in H$. Then (H, o) is called a total hypergroup. It is seen that in this case for every $x, y \in H$, we have $x\beta y$. So $x\beta^*y$ and thus $\beta^*(x) = \beta^*(y)$. Hence H/β^* is a singleton.

Definition 1.2 A Hypergroup (H, o) is very thin if all its hyperoperations are operation except only one.

Let (H, o) be a very thin hypergroup. Hence for a unique $(a , b) \in H^2$, we have $|a$ o $b| \geq 2$. Set $|a$ o $b| = A$. So for all $x \in A$, $\beta^*(x) = A$. It is clear that if $|A|$ = m, then $|H/\beta^*| = n - m$.

Definition 1.3 Let(H_1, o) and (H_2, ∗) be two hypergroups. A map $f : H_1 \to H_2$ is called:

(1) a homomorphism if $f(x \circ y) \subseteq f(x) * f(y)$; for all $x, y \in H_1$.

(2) a good homomorphism if $f(x \circ y) = f(x) * f(y)$; for all $x, y \in H_1$.

(3) an isomorphism if it is a one to one and onto good homomorphism. If f is an isomorphism, then H_1 and H_2 are said to be isomorphic.

2. COMPUTATION OF β* EQUIVALENCE CLASSES AND FUNDAMENTAL GROUP

In the sequel we assume that (H, o) is a finite hypergroup of order n. Clearly,

$$a_i \in \beta\,(a_j) \Longleftrightarrow (a_i,\ a_j) \in \beta = \bigcup_{n\geq 1} \beta_n$$

$$\Longleftrightarrow (a_i,\ a_j) \in\ \beta_m,\ \text{for some}\ m \geq 1$$

$$\Longleftrightarrow \{a_i,\ a_j\} \subseteq \prod_{i=1}^{m} b_i,\ \text{for some}\ b_1,\ b_2\ ,\ldots, b_m \in H.$$

In H, the basic point on computation of equivalence classes of relation β is the existence of a least upper bound (or an upper bound) $m \in$ N such that for all $a, b \in H$, we have $b \in \beta(a)$ iff $(a, b) \in \cup^{m}_{i=1} \beta_i$. Let $P_m = \{ \Pi^{m}_{i=1} a_i \,|a_i \in H\}$. We start with the following lemma.

Lemma 2.1 Let (H, o) be a hypergroup of order n and $P_m \subseteq \cup^{m-1}_{i=2} P_i$, for some $m \geq 3$ ($m \in$ N) . Then for all $k > m$, we have $P_k \subseteq \cup^{m-1}_{i=2} P_i$.

Proof. Let c ∈ Pm+1. So we have:

$$c = (a_1 \text{ o } a_2 \text{ o } \ldots \text{ o } am) \text{ o } a_{m+1}\ , \text{ for some } a_1, a_2\ , \ldots\ , a_m, a_{m+1}.$$

Suppose $b = a_1 \text{ o } a_2 \text{ o } \ldots a_m$, then

$$c = b \circ a_{m+1} \Longrightarrow\ c\ \in (\bigcup_{i=2}^{m-1} P_i) \circ a_{m+1}$$

$$\Longrightarrow c\ \in \bigcup_{i=2}^{m-1} (\ P_i \circ a_{m+1})$$

$$\Longrightarrow c\ \in \bigcup_{i=2}^{m-1} P_{i+1}\ .$$

Since $P_m \subseteq \cup^{m-1}_{i=2} P_i$, it follows that $c \in \cup^{m-1}_{i=2} P_i$. Therefore, $P_{m+1} \subseteq \cup^{m-1}_{i=2} P_i$. So the assertion holds for $k = m + 1$. Similarly it is true for $k > m$, by induction.

Corollary 2.2 Let (H, o) be a hypergroup such that $|H| = n$ and for some $m \geq 3$(m 2 N) we have $P_m \subseteq \cup^{m-1}_{i=2} P_i$. Then for all $a, b \in H$ and $a \neq b$, we have $b \in \beta(a)$ iff$(a, b) \in \cup^{m-1}_{i=2} \beta_i$

Proof. Let $a, b \in$H. So $b \in \beta(a)$ iff $\{ a, b\} \subseteq P_k$, for some $k \in$ N. If $2 \leq k \leq m - 1$, then it is clear that $P_k \subseteq \cup^{m-1}_{i=2} P_i$. Now if $k \geq m$, Lemma 2.1 results $P_k \subseteq \cup^{m-1}_{i=2} P_i$. So $\{a, b\} \subseteq P_k \subseteq \cup^{m-1}_{i=2} P_i$, that is, $(a, b) \in \cup^{m-1}_{i=2} \beta_i$.

Proposition 2.3 Let (H, o) be a hypergroup, such that $|H| = n$. Then there exists j, $2 \leq j < 2^n$ such that $P_k \subseteq \cup^{j}{}_{i}{=}2\ P_i$ for all k.

Proof. Since $|P^*(H)| = 2^n - 1$, Then $\cup^{2n-1}_{i=2} P_i$ is an upper bound for the sequence $P_2 \subseteq P_2 \cup P_3 \subseteq \ldots \subseteq \cup^{2n-1}_{i=2} P_i \subseteq \ldots$, that is, for all $k \geq 2$, we have $P_k \subseteq \cup^{k}_{i=2} P_i \subseteq \cup 2^{n-1}{}_{i}{=}2\ P_i$. This complete the proof.

The next result is an immediate consequence of Corollary 2.2 and Proposition 2.3.

Theorem 2.4 Let (H, o) be a hypergroup, in which $|H| = n$. For all $a, b \in H(a \neq b)$, we have $b \in \beta(a)$ iff $(a, b) \in \cup^{j}_{i=2} \beta_i$, for some $2 \leq j < 2^n$.

Algorithm 2.5 Next, we come to the construction the algorithm, which computes β- equivalence classes, according to the following description:
Step 1:

$$T = \left\{ A \in P_2 \,\middle|\, |A| \geq 2, \exists B \in P_2 : \ |B| \geq 2, \ \ A \cap B \neq \emptyset \right\}$$

$for\ each\ \ i \in\ H\ \ begin\ from\ 1$

$\{$

$if\ \ i \in A\ \ \ for\ some\ A \in T$

$$Y(A) = \bigcup_{i \in I_0} \left\{ B_i \in T \,\middle|\, \forall B_i, \exists B_j \neq B_i : \ B_i \cap B_j \neq \emptyset \ and\ \exists B_k : \ B_k \cap A \neq \emptyset \right\}$$

$\beta(i) = Y(A)$

$H = H \setminus Y(A)$

$else\ if\ i \in B\ \ for\ some\ B \in\ P_2\ , |B| \geq\ 2$

$\beta(i) = B$

$H = H \setminus B$

$else$

$\beta(i) = \{i\}$

$\}$

If P3 ⊆ P2, then by Lemma 2.1 the routine finishes. Otherwise, by Proposition 2.3, we have Pj ⊆ j-1∪i=2 Pi for some j ≥ 4. Hence we move to the next step.

Step 2:

Let B = {β(i1), _(i2), … , β (ik)} be the output of Step 1. We set:

$P' = P_3$

$for\ j = 4\ to\ 2^n - 1$

$while\ \ Not(P_j \subseteq \bigcup_{k=2}^{j-1} P_k)$

$P' = P' \bigcup P_j$

$$T' = \left\{ B_i \in \mathbb{B} \,\middle|\, \exists B_j \neq B_i\,,\ \exists k_1 \in B_1\,,\ \exists k_2 \in B_2\,,\ \exists C \in P' : \ \{k_1\,, k_2\} \subseteq C \right\}$$

And we continue as bellow:

$$for\,each\ i \in H\ \ begin\ from\ 1$$
$$\{$$
$$if\ \ i \in A\,, for\,some\ A \in T'$$

$$Y'(A) = \bigcup_{i \in I_0} \Big\{ B_i \in T' \big| \forall B_i\, \exists B_j \neq B_i,\, k_1 \in B_i,\, k_2 \in B_j,\, C \in P' :\ \{k_1, k_2\} \subseteq C$$
$$and\ \exists B_k,\ k_3 \in B_k,\, k_4 \in A,\, C' \in P' :\ \{k_3, k_4\} \subseteq C' \Big\}$$

$$\beta(i) = Y'(A)$$
$$H = H \setminus Y'(A)$$
$$print\ \beta(i)$$
$$else\ if\ i \in B\ for\ some\ B \in Y(A)$$
$$\beta(i) = B$$
$$print\ \beta(i)$$
$$H = H \setminus B$$
$$else\ if\ \ i \in C\ \ for\ \ some\ \ C \in P_2,\ |C| \geq 2\ ,$$
$$\beta(i) = C$$
$$print\ \beta(i)$$
$$H = H \setminus C$$
$$else$$
$$\beta(i) = \{i\}$$
$$\}$$

At the end of the process we obtain β-equivalence classes. Now in order to obtain the Fundamental group, (H=/β*, ⊗), we have the following theorem:

Theorem 2.6 Let (H, o) be a hypergroup, such that $|H| = n$ and $k \in$ N be the number of all β-equivalence classes. Let ⊗ be the operation on H/β^* and $\beta(j) \otimes \beta(h) = \beta(t)$, for some $t \in j \circ h \subseteq \beta(j) \circ \beta(h)$. If we rewrite the set of β- equivalence classes, as the following:

$$\mathbb{B} = \{\beta(i_1),\, \beta(i_2),\, \ldots, \beta(i_k)\},\ i_1 = \min\ \beta(i_1),\ i_2 = \min\ \beta(i_2),\ \ldots,\ i_k = \min\ \beta(i_k),$$

then we have $\beta(t) = \beta(i_l)$, $\beta(i_l) \cap j \circ h \neq \varnothing$, for one and only one $\beta(i_l) \in$ B.

Proof. In group (B, ⊗) for one and only one t0 $\in \{i_1, i_2, \ldots, i_k\}$ we have:

$$\beta(j) \otimes \beta(h) = \beta(t) = \beta(t'),\ \ t' = \min \beta(t)$$

Since $t \in \beta(t)$ is an arbitrary member of $j \circ h$, so we have $j \circ h \cap \beta(t) \neq \varnothing$. Thus $j \circ h \cap \beta(t0) \neq \varnothing$. Now it is sufficient to set $i_l = t0$ and this complete the proof.

Algorithm 2.7 Let B = $\{\beta(i_1)\ ,\ \beta(i_2)\ ,\ \beta(i_3)\ ,\ \ldots\ ,\ \beta(i_k)\}$ be the set of all β-equivalence classes obtained through Algorithm 2.5. We present an algorithm, in brief, for computation of fundamental group, as follows:

$if\ |\beta| \geq 2$
 $for\ all\ i_j \in \{i_1, i_2, ..., i_k\}$
   {
     $for\ all\ i_h \in \{i_1, i_2, ..., i_k\}$
     {
       $if\ \exists B \in \mathbb{B}, B \bigcap j \circ h \neq \phi$
         $print\ (" \beta(i_j) \otimes \beta(i_h) = ", B)$
       $else$
         $h = h + 1$
     }
     $j = j + 1$
   }
$else$
   $print\ trivial\ case$
end

3. JAVA PACKAGE AND EXAMPLES

In this section we use a comprehensive program written totally in Java to compute the fundamental group of a given finite hypergroup. This program also produces all hypergroups of order $n \leq 3$ ($n \in$ N) . It consists of two sub-programs (Hypergroupgenerator and Main). For summarizing, those part of program that are devoted to data entry and output, memory allocation and file management, are deleted. In the following, the Hypergroup-generator, counts all hypergroups of order $n \leq 3$ ($n \in$ N) and isomorphism classes of them. This sub-program also enumerates quasihypergroups of order n and all β-equivalence classes.

```
public class HyperGroupGenerator {
public void hyperGroupGenerator(int n, int
numberOfHyperGroup)
              while (numberOfHyperGroup > 0) {
              for $(int i = 1; i <= hyperStructure.size
              (); i++){
              Point point^1 = hyperStructure.getById(i);
              HashMap<Point, Set<Point>> productTable =
              new HashMap<>();
              for (int j = 1; j <= hyperStructure.size(
              ); j++) {
              Point point^2 = hyperStructure.getById(j;
              Set<Point> set = new HashSet<Point>(
              powerSet.get(index[(i - 1) * n + (j - 1)
              ]));
              productTable.put(point2, set);
```

```
        }
       point^1.setProductTable(productTable);
 }
 boolean isReproduction = hyperStructure
 .isReproduction();
 if (isReproduction) {
            if (hyperStructure.isAssociative
            ()) {
                   hyperGroupWriter.write
                   ("HyperGroup");
 hyperGroupWriter.write(hyperStructure
 .printHyperStructure());
   hyperStructure.getEquivalenceClasses
   (hyperGroupWriter);
        numberOfHyperGroup--;
        countHyperGroup++;
  } else if (calcQuasi) {
        quasiHyperGroupWriter.write
        ("QuasiHyperGroup");
        quasiHyperGroupWriter.write
        (hyperStructure
         .printHyperStructure());
           countQuasiHyperGroup++;
              }
          }
          boolean finish = true;
          for (int i = 0; i < index.length;
          i++) {
                if (index[i] > 0) {
                finish = false;
                index[i] = index[i] - 1;
                for (int j = 0; j< i; j++) {
                  index[j] =  powerSet.size()-1;
                }
                break;
             }
         }
         if (finish) {
                  break;
         }
      }
hyperGroupWriter.write(countHyperGroup + "");
if(calcQuasi){
  uasiHyperGroupWriter.write
  (countQuasiHyperGroup+"");
}
```

```
     hyperGroupWriter.close();
     quasiHyperGroupWriter.close();
}
public void hyperGroupGeneratorUpToIso(int n) throws
 Exception {
            File theDir = new File("uptoiso");
            if (!theDir.exists()) {
            theDir.mkdir();
            }
HashMap$<$Integer, Integer> numberOfEveryClass =
new HashMap<Integer,
 Integer>();
        BufferedReader br = new BufferedReader
        (new FileReader(new File(
                     "Hypergroup.txt")));
        int numberOfIsoHyperGroup = 0;
        String line = "";
        while ((line = br.readLine()) != null) {
               if (line.equals("HyperGroup")) {
                        boolean isExists = false;
                      HyperStructure hyperStructure=
                         new HyperStructure(n);
        String[] productTable;
        Set$<$Point$>$ hashSet = null;
        for (int i = 1; i <= hyperStructure.size();
        i++) {
                productTable = br.readLine().split
                (" ");
              nt point = hyperStructure.getById(i);
              or (int j = 0; j< productTable.length;j++){
                        String product[] = productTable[j]
                        .split(",");
                        hashSet = new HashSet<Point>();
                        for (int k = 0; k< product.length
                        ; k++) {
                           hashSet.add(hyperStructure
                           .getById
                           (Integer

                                   .parseInt(product[k])));
               }
                                         point.addProduct
                                          (hyperStructure
                                         .getById(j + 1),
                                               hashSet);
       }
```

```
}
for (int i = 0; i < numberOfIsoHyperGroup; i++) {
                  HyperStructure hyperStructure2 =
                getHyperStructureFromFile("uptoiso"
                + i + ".txt");
  if (isIsomorphic(hyperStructure2, hyperStructure)) {
  numberOfEveryClass.put(i, numberOfEveryClass.get(i)+1);
        isExists = true;
        break;
         }
}
if (!isExists) {
 BufferedWriter writer = null;
 File output;
 output = new File("uptoiso/" + numberOfIsoHyperGroup
    + ".txt");
 writer = new BufferedWriter(new FileWriter(output));
 writer.write(n + "{\textbackslash}n"+ hyperStructure
 .printHyperStructure());
 numberOfEveryClass.put(numberOfIsoHyperGroup, 1);
numberOfIsoHyperGroup++;
// System.out.println(numberOfIsoHyperGroup);
writer.flush();
 writer.close();
    }
  }
}
HashMap<Integer, Integer> tmp = new HashMap<Integer,
Integer>();
   for(int i = 0 ; i< numberOfIsoHyperGroup ;i++){
       if(!tmp.containsKey(numberOfEveryClass.get(i))){
                   tmp.put(numberOfEveryClass.get(i), 0);
   }
                   tmp.put(numberOfEveryClass.get(i),
           tmp.get(numberOfEveryClass.get(i))+1);
   }
   BufferedWriter writer = null;
   File output;
   output = new File("uptoiso/numberOfEveryClass.txt");
   writer = new BufferedWriter(new FileWriter(output));
   for (Iterator<Integer> iterator = tmp.keySet()
   .iterator(); iterator
   .hasNext();) {
             Integer key = iterator.next();
   writer.write(key+" := "+ tmp.get(key)+"{
   textbackslash}n");
```

```
    }
    writer.flush();
    writer.close();
    br.close();
}
public HyperStructure getHyperStructureFromFile
(String fileName) {
        try (InputStreamReader file = new
        InputStreamReader(
                new FileInputStream(fileName));
          feredReader br = new BufferedReader(file)) {
         rStructure hyperStructure = new HyperStructure(
                  Integer.parseInt(br.readLine()));
           String[] productTable;
           Set<Point> hashSet = null;
       or (int i = 1; i <= hyperStructure.size(); i++) {
                productTable = br.readLine().split(" ");
                Point point = hyperStructure.getById(i);
          for (int j = 0; j < productTable.length; j++) {
           tring product[] = productTable[j].split(",");
                    hashSet = new HashSet<$Point>();
              r (int k = 0; k < product.length; k++) {
               hSet.add(hyperStructure.getById(Integer
                                  .parseInt(product[k])));
                    }
               .addProduct(hyperStructure.getById(j + 1),
                    hashSet);
                }
              }
                      return hyperStructure;
              } catch (FileNotFoundException e) {
                     e.printStackTrace();
              } catch (IOException e) {
                     e.printStackTrace();
              } catch (Exception e){
                     e.printStackTrace();
              }
              return null;
            }
            public boolean isIsomorphic(HyperStructure
             hyperStructure^1,
                     HyperStructure hyperStructure2) {
               if (hyperStructure1.size() != hyperStructure2
               .size()){
                      return false;
                }
```

```
PermutationGenerator pg = new
 PermutationGenerator(
         hyperStructure1.size(), 1);
permute: while (pg.hasMore()) {
       int[] temp = pg.getNext();
       boolean iso = true;
       for (Iterator<Point> iterator
       = hyperStructure1
       .iterator(); iterator
               .hasNext();) {
           Point point11 = iterator.next();
           for (Iterator<Point> iterator2
           = hyperStructure1
           .iterator(); iterator2
                .hasNext();) {
    int point21 = (Point) iterator2.next();\
     hSet<Point> product1 = (HashSet<Point>)
               point11
                .productOfPoint(point21);
 Point point12 = hyperStructure2.getById(temp
              [point11
               .getId() - 1]);
Point point22 = hyperStructure2.getById(temp
              [point21
                   .getId() - 1]);
HashSet<Point> product2 = (HashSet<Point>)
              point12\\
                 .productOfPoint(point22);\\
    if (product1.size() != product2.size()) {
               // System.out.println(product1
                       .size() + "," +
                       // product2.size());
                       iso = false;
                       continue permute;
              } else {
    for (Iterator<Point> iterator3 = product1
                     .iterator();
                       iterator3.hasNext();){
             Point point1 = iterator3.next();
                     boolean terminate = true;
     for (Iterator<Point> iterator4 = product2
          .iterator();iterator4.hasNext();) {
               Point point2 = iterator4.next();
                if (temp[point1.getId() - 1] =
                             point2.getId()){
                            terminate = false;
```

```
                                        break;
                                      }
                              }
                              if (terminate) {
                                  iso = false;
                                  continue permute;
                               }
                          }
                      }
                 }
            }
            if (iso) {
                          return true;
            }
        }
         return false;
    }
  public static void main(String args[]) throws Exception {
         int n = Integer.parseInt(JOptionPane
                .showInputDialog("Size Of Hypergroup:"));
   int numberOfHyperGroup = Integer.parseInt(JOptionPane
              .showInputDialog("Number Of HyperGroups:"));
             String calcQuasi = JOptionPane
              .showInputDialog("Calculate QuasiHyperGroup,
                       y or n?");
    HyperGroupGenerator hgg = new HyperGroupGenerator();
                    String calcIsomorphism = JOptionPane
                 .showInputDialog("Calculate Isomorphism,
                       y or n?");
       gg.hyperGroupGenerator(n, numberOfHyperGroup, "y"
          .equals(calcQuasi));
        if ("y".equals(calcIsomorphism.toLowerCase())) {
             hgg.hyperGroupGeneratorUpToIso(n);
          }
       }
    }
```

The following example illustrates the output of sub-program, Hypergroup-generator, after enumeration of hypergroups of order 3:

Example 1. If set $n = 3$ in the above sub-program, It results there are 7^9 hyperstructures. Then we find 10323979 quasihypergroups, 23192 hypergroups and 3999 hypergroups(up to of isomorphism) of order $n = 3$, distributed on 5 classes of cardinality. Moreover, the fundamental groups of these hypergroups are as follows:

One group is isomorph to Z_3, 9 of them are isomorph to Z_2 and the rest are trivial groups (0). In the following we examine for (H, o) given in Table 1.

Table 1. Hypergroup (Example 1)

o	1	2	3
1	{1}	{2,3}	{1}
2	{2,3 }	{1}	{2,3}
3	{1}	{2,3}	{1}

Equivalence Classes

$\beta(1) = \{1\}$, $\beta(2) = \{2, 3\}$

That is, $(H/\beta^{*}, \otimes) = Z_2$:

Now by the next sub-program (Main), first it is checked that is an arbitrary hypergroupoid $(H, \circ)$ of order $n \in N$ a hypergroup or not. If it is a hypergroup, this subprogram computes its β-equivalence classes and fundamental group. This can be done by the following written lines:

```
public class Main{
   public HyperStructure createHyperStructure() {
   try (InputStreamReader file = new InputStreamReader(
   HyperStructure hyperStructure = new HyperStructure(
   for (int i = 1; i $<$= hyperStructure.size(); i++) {
       productTable = br.readLine().split(" ");
       Point point = hyperStructure.getById(i);
       for (int j = 0; j $<$ productTable.length; j++) {
           for (int k = 0; k $<$ product.length; k++) {
              hashSet.add(hyperStructure.getById(Integer
                      .parseInt(product[k])));
           }
nt.addProduct(hyperStructure.getById(j + 1), hashSet);
       }
    }
    return hyperStructure;
} catch (FileNotFoundException e) {
  e.printStackTrace();
} catch (IOException e) {
  e.printStackTrace();
} catch (Exception e) {
 e.printStackTrace();
}
return null;
}
public static void main(String[] args) throws IOException {
try {
   HyperStructure hyperStructure = new Main()
   .createHyperStructure();
```

```
boolean isReproduction = hyperStructure.isReproduction();
 boolean isAssociative = hyperStructure.isAssociative();
 if (isReproduction && isAssociative) {
     writer.write("HyperGroup");
     ArrayList<HashSet<Point>>(equvalenceClasses)
     = (hyperStructure)
          .getEquivalenceClasses(writer);
     System.out.println(equvalenceClasses.get(0).size());
     String calcFundamentalGroup = JOptionPane
     .showInputDialog("Calculate Fundamental Group,
     y or n?");
     if("y".equals(calcFundamentalGroup.toLowerCase())){
      if (equvalenceClasses.size() > 1) {
        hyperStructure.calculateFundamentalGroup(
        equvalenceClasses, writer);
      } else {
               writer.write("\nTrivial Case");
             }
      }
   } else if (isReproduction) {
      writer.write("QuasiHypergroup");
     } else {
      writer.write("Hypergroupoid");
      }
       } catch (Exception e) {
       } finally {
       writer.close();
       }
   }
}
```

Here, we observe some examples of hypergroups, used by sub-program, Main, and the outputs:

Example 2. Consider hypergroupoids (H_1, o) and (H_2, o) as in Table 2:

Table 2. Hyperstructures (Example 2)

o	1	2	3
1	{1}	{2,3}	{2,3}
2	{2,3 }	{1}	{1}
3	{2,3}	{1}	{1}

Hyperstructure (H_1, o)

*	1	2	3
1	{1,3}	{3}	{2}
2	{2 }	{1,2,3}	{2}
3	{2}	{1}	{1,2,3}

Hyperstructure (H_2, $*$)

For these two hyperstructures, we get the following outputs:

(H_1, o) is a hypergroup, with the following equivalence classes

$$\beta(1) = \{1\},\ \beta(2) = \{2, 3\}$$

Hence (H_1/β^* , $\otimes$) $=Z_2$, while (H_2, $*$) is not a hypergroup and actually is a quasihypergroup.

Example 3. Consider two hyperstructures of order 4 as follows

Table 3. Hyperstructures (Example 3)

o	1	2	3	4
1	{1,2}	{1,2}	{3,4}	{3,4}
2	{1,2}	{1,2}	{3,4}	{3,4}
3	{3,4}	{3,4}	{1}	{2}
4	{3,4}	{3,4}	{2}	{1}

Hyperstructure (H_1, o)

*	1	2	3	4
1	{1}	{2,3}	{2,3}	{4}
2	{2,3}	{4}	{4}	{1}
3	{2,3}	{4}	{4}	{1}
4	{4}	{1}	{1}	{2,3}

Hyperstructure (H_2, ∗)

For hyperstructure (H_1, o), it concludes that (H_1, o) is a hypergroup, with the following equivalence classes

$$\beta(1) = \{1, 2\},\ \beta(2) = \{3, 4\}$$

Hence (H_1/β^*, ⊗) $=Z_2$. For hyperstructure (H_2, ∗) we get the following output:

Hypergroup
Equivalence Classes

$$\beta(1) = \{1\}$$
$$\beta(2) = \{2, 3\}$$
$$\beta(4) = \{4\}$$

That is, (H_2/β^*, ⊗) $=Z_3$.

Example 4. Give two hyperstructures of order 5 as follows:

Table 4. Hyperstructure (H_1, o) (Example 4)

o	1	2	3	4	5
1	{1}	{2, 3}	{2, 3}	{4}	{5}
2	{2, 3}	{4}	{4}	{5}	{1}
3	{2, 3}	{4}	{4}	{5}	{1}
4	{4}	{5}	{5}	{1}	{2, 3}
5	{5}	{1}	{1}	{2, 3}	{4}

Table 5. Hyperstructure (H_2, ∗) (Example 4)

*	1	2	3	4	5
1	{1}	{2}	{3, 4}	{4}	{5}
2	{2}	{1, 5}	{3, 4}	{3, 4}	{2}
3	{3}	{3, 4}	{1, 2}	{1, 2}	{5}
4	{3, 4}	{3, 4}	{1, 2}	{1, 2}	{5}
5	{5}	{2}	{5}	{5}	{1}

Then for hyperstructure (H_1, o), it results:

Hypergroup
Equivalence Classes

$$\beta(1) = \{1\},\ \beta(2) = \{2, 3\},\ \beta(4) = \{4\},\ \beta(5) = \{5\}$$

Therefore, (H_1/β^*, ⊗) = Z_4 (see Table 6).

Table 6. Operations (Example 4)

$\otimes$	$\beta(1)$	$\beta(2)$	$\beta(4)$	$\beta(5)$
$\beta(1)$	$\beta(1)$	$\beta(2)$	$\beta(4)$	$\beta(5)$
$\beta(2)$	$\beta(2)$	$\beta(4)$	$\beta(5)$	$\beta(1)$
$\beta(4)$	$\beta(4)$	$\beta(5)$	$\beta(1)$	$\beta(2)$
$\beta(5)$	$\beta(5)$	$\beta(1)$	$\beta(2)$	$\beta(4)$

For hyperstructure (H_2, $*$) we get the following output:

Hypergroupoid

Example 5. Consider hyperstructure (H, o) of order 6 as follows:

Table 7. Hyperstructure (H, o) (Example 5)

o	1	2	3	4	5	6
1	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{5\}$	$\{6\}$
2	$\{2\}$	$\{1\}$	$\{3\}$	$\{4\}$	$\{5\}$	$\{6\}$
3	$\{3\}$	$\{3\}$	$\{1,2\}$	$\{4\}$	$\{5\}$	$\{6\}$
4	$\{4\}$	$\{4\}$	$\{4\}$	$\{6\}$	$\{1,2,3\}$	$\{5\}$
5	$\{5\}$	$\{5\}$	$\{5\}$	$\{1,2,3\}$	$\{6\}$	$\{4\}$
6	$\{6\}$	$\{6\}$	$\{6\}$	$\{5\}$	$\{4\}$	$\{1,2,3\}$

And we get the following output:

Hypergroup
Equivalence Classes

$\beta(1) = \{1, 2, 3\}$, $\beta(4) = \{4\}$, $\beta(5) = \{5\}$, $\beta(6) = \{6\}$

Table 8. Operations (Example 5)

$\otimes$	$\beta(1)$	$\beta(4)$	$\beta(5)$	$\beta(6)$
$\beta(1)$	$\beta(1)$	$\beta(4)$	$\beta(5)$	$\beta(6)$
$\beta(4)$	$\beta(4)$	$\beta(6)$	$\beta(1)$	$\beta(5)$
$\beta(5)$	$\beta(5)$	$\beta(1)$	$\beta(6)$	$\beta(4)$
$\beta(6)$	$\beta(6)$	$\beta(5)$	$\beta(4)$	$\beta(1)$

It is seen that ($H=/\beta^*$, $\otimes$) is isomorph to Klein quartet group, that is, ($H=/\beta^*$, $\otimes$) = $Z_2 \times Z_2$ (see Table 8).

Example 6. Consider hyperstructure (H, o) of order 7 (as shown in Table 9).

Table 9. Hyperstructure (H, o) (Example 6)

o	1	2	3	4	5	6	7
1	{1}	{2}	{3}	{4}	{5}	{6, 7}	{6, 7}
2	{2}	{1}	{5}	{6, 7}	{3}	{4}	{4}
3	{3}	{6, 7}	{1}	{5}	{4}	{2}	{2}
4	{4}	{5}	{6, 7}	{1}	{2}	{3}	{3}
5	{5}	{4}	{2}	{3}	{6, 7}	{1}	{1}
6	{6, 7}	{3}	{4}	{2}	{1}	{5}	{5}
7	{6, 7}	{3}	{4}	{2}	{1}	{5}	{5}

And we get the following output:

HyperGroup
Equivalence Classes

$\beta(1) = \{1\}, \beta(2) = \{2\}, \beta(3) = \{3\}$

$\beta(4) = \{4\}, \beta(5) = \{5\}, \beta(6) = \{7, 6\}$

Table 10. Operations (Example 6)

$\otimes$	$\beta(1)$	$\beta(2)$	$\beta(3)$	$\beta(4)$	$\beta(5)$	$\beta(6)$
$\beta(1)$	$\beta(1)$	$\beta(2)$	$\beta(3)$	$\beta(4)$	$\beta(5)$	$\beta(6)$
$\beta(2)$	$\beta(2)$	$\beta(1)$	$\beta(5)$	$\beta(6)$	$\beta(3)$	$\beta(4)$
$\beta(3)$	$\beta(3)$	$\beta(6)$	$\beta(1)$	$\beta(5)$	$\beta(4)$	$\beta(2)$
$\beta(4)$	$\beta(4)$	$\beta(5)$	$\beta(6)$	$\beta(1)$	$\beta(2)$	$\beta(3)$
$\beta(5)$	$\beta(5)$	$\beta(4)$	$\beta(2)$	$\beta(3)$	$\beta(6)$	$\beta(1)$
$\beta(6)$	$\beta(6)$	$\beta(3)$	$\beta(4)$	$\beta(2)$	$\beta(1)$	$\beta(5)$

Which $|\beta(2)| = 2$, $|\beta(6)| = 3$, …. Hence the fundamental group (H/β^*, $\otimes$) is isomorph to S_3.

Example 7. Give hyperstructure (*H*, o) of order 8 as it shown in Table 11.

Table 11. Hyperstructure (*H*, o) (Example 7)

o	1	2	3	4	5	6	7	8
1	{1}	{2}	{3}	{4}	{5}	{6}	{7}	{8}
2	{2, 6}	{1, 5}	{4, 8}	{3, 7}	{2, 6}	{1, 5}	{4, 8}	{3, 7}
3	{3}	{4}	{5}	{6}	{7}	{8}	{1}	{2}
4	{4, 8}	{3, 7}	{2, 6}	{1, 5}	{4, 8}	{3, 7}	{2, 6}	{1, 5}
5	{5}	{6}	{7}	{8}	{1}	{2}	{3}	{4}
6	{2, 6}	{1, 5}	{4, 8}	{3, 7}	{2, 6}	{1, 5}	{4, 8}	{3, 7}
7	{7}	{8}	{1}	{2}	{3}	{4}	{5}	{6}
8	{4, 8}	{3, 7}	{2, 6}	{1, 5}	{4, 8}	{3, 7}	{2, 6}	{1, 5}

Then for hyperstructure (*H*, o), it results:

Hypergroup
Equivalence Classes

$\beta(1) = \{1, 5\}, \beta(2) = \{2, 6\}, \beta(3) = \{3, 7\}, \beta(4) = \{4, 8\}$

Table 12. Operations (Example 7)

$\otimes$	$\beta(1)$	$\beta(2)$	$\beta(3)$	$\beta(4)$
$\beta(1)$	$\beta(1)$	$\beta(2)$	$\beta(3)$	$\beta(4)$
$\beta(2)$	$\beta(2)$	$\beta(1)$	$\beta(4)$	$\beta(3)$
$\beta(3)$	$\beta(3)$	$\beta(4)$	$\beta(1)$	$\beta(2)$
$\beta(4)$	$\beta(4)$	$\beta(3)$	$\beta(2)$	$\beta(1)$

As in example 5, here we also have (H/β^*, $\otimes$) is isomorph to Klein quartet group and (H/β^*, $\otimes$) = $Z_2 \times Z_2$ (see Table 12).

Example 8. We give a hyperstructure (H, o) of order 9 as follows:

Table 13. Hyperstructure (H, o) (Example 8)

o	1	2, 3, 4, 5, 6	7, 8, 9
1	{1}	{2,3}	H - {1}
2	{2}	{1,3}	H - {2}
3	{3}	{1,2}	H - {3}
4	{4}	{1,2,3,5,6}	H - {4}
5	{5}	{2,3,4,5,6}	H - {5}
6	{6}	{1,2,3,4,5}	H - {6}
7	{7}	H - {7}	H - {7}
8	{8}	H - {8}	H - {8}
9	{9}	H - {9}	H - {9}

And we get the following output:

HyperGroup
Equivalence Classes

$\beta(1) = H$

Trivial case, that is, (H=/β^*, $\otimes$) = (0).

4. CONCLUSION

We gave two algorithms to compute the fundamental group of a given finite hypergroup. One of this algorithm produces all hypergroups of order less than or equal 3 and another algorithm first check that a given hyperstructure is a hypergroup or no and the compute the fundamental group derived from a hypergroup via the fundamental relation.

Acknowledgements

The second author partially has been supported by “Algebraic Hyperstructure Excellence (AHETM), Tarbiat Modares University, Tehran, Iran” and “Research Center in Algebraic Hyperstructures and Fuzzy Mathematics, University of Mazandaran, Babolsar, Iran”.

REFERENCES

[1] R. Ameri, T. Nouzari, A new characterization of fundamental relation on hyperrings, *Int. J. Contemp. Sciences* 5(10)(2010) 721-738.

[2] R. Bayon, N. Lygeros, Advanced results in enumeration of hyperstructures, *Journal of Algebra* 320 (2008) 821-835.

[3] P. Corsini, Prolegomena of Hypergroup Theory, *second edition. Aviani Editor* (1993).

[4] P. Corsini, V. Leoreanu, Applications of Hyperstructures Theory, *in: Advanced in Mathematics, Kluwer Academic Publishers* (2003).

[5] I. Cristea, M. Jafarpour, S.S. Mousavi and A. Soleymani, *Enumeration of Rosenberg hypergroups, Computers and Mathematics with Applications*60 (2010) 2753-2763.

[6] B. Davvaz, M. Karimian, On the γ^*_n-complete Hypergroups, *European journal of Combinatorics* 28 (2007) 86-93.

[7] M. De Salvo, On the *n**-complete Hypergroups, *Discrete Mathematics* 208/209 (1990) 177-188.

[8] D. Freni, A new characterization of the derived hypergroup via strongly regular equivalences, *Communication in algebra* 30(8) (2002) 3977-3989.

[9] D. Freni, On a Strongly Regular Relation in Hypergroupoids, *Pure Math. Appl, Ser. A* 3-4 (1992) 191-198.

[10] D. Freni, Une note sur le coeur d'un hypergroupe et sur la cloture transitive β^* de β. (French) [A note on the core of a hypergroup and the transitive closure β^* of β], Rivista. di Mat. Pura Appl 8 (1991) 153-156.

[11] M. Koskas, Groupoides, Demi-hypergroupes et hypergroupes, *J. Math. Pures Appl* 49 (1970) 155-192.

[12] F. Marty, Sur une generalization de la notion de groupe, in: *8th Congress Math. Scandinaves, Stockholm, Sweden* (1934) 45-49.

[13] G. Nordo, An algorithm on number of isomorphism classes of hypergroups of order 3, Italian J. Pure Appl. Math 2 (1997) 37-42.

[14] S. Spartalis, T. Vougiouclis, The fundamental relations on H_v-rings, Rivista. di Math. Pura Appl 13 (1994) 7-20.

[15] Ch. Tsitouras, Ch. G. Massouros, Enumeration of Rosenberg-type hypercompositional structures defined by binary relations, *European Journal of Combinatorics* 33 (2012) 1777-1786.

[16] Ch. Tsitouras, Ch. G. Massouros, On enumeration of hypergroups of order 3, *Computers and Mathematics with Applications* 59 (2010) 519-523.

[17] T. Vougiouklis, Hyperstructures and Their Representations, *Hadronic Press, Palm Harbor, FL* (1994).

[18] T. Vougiouklis, The fundamental relations in hyperrings. The general hyperfield, *Proceeding of 4th International congress in Algebraic Hyperstructures and Its Applications AHA* (1990).