COMPUTATION OF FUNDAMENTAL GROUP OF A FINITE HYPERGROUP

Karim ABBASI¹, Reza AMERI*², Yahya TALEBI-ROSTAMI³

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

Received: 01.01.2018   Accepted: 16.04.2018

ABSTRACT

Let (H, °) be a hypergroup and β* be the fundamental relation on H, that is, the smallest equivalence relation on H such that the quotient (H/β*, ⊗) 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/β*, ⊗). 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/β*, ⊗). 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/β*, ⊗). 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 $o : H \times H \rightarrow P^*(H)$. A couple $(H, o)$ is said to be a hypergroupoid. Let $A, B$ be subsets of $H$. The hyperproduct $A \circ B$ is defined as

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

Also if there is no confusion, $\{a\}, A \circ \{a\}$ fag and $\{a\} \circ A$ are denoted by $a, A \circ a$ and $a \circ A$, respectively. From now on, when there is no ambiguity, we write $ab$ instead of $a \circ o b$ 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 quasihipergroup if for all $x \in H$, we have $xH = Hx = H$ (reproductivity). We say that a hypergroupoid $(H, o)$ is a hypergroup if $(H, o)$ verifies both associativity and reproductivity. Let $(H, o)$ be a finite hypergroup. We define the relation $\beta = \cup_{n \geq 1} \beta_n$, in which $\beta_n = \{(a, a) | a \in H\}$ and for all $(a, b) \in H^2$, $a\beta b$ iff $(a, b) \subseteq \Pi_{m=1}^{\infty} Z_m$, 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 $z_1 = a \beta 2\beta \beta \ldots \beta z_m = b$, for some $z_1, z_2, \ldots, z_m \in H^r$. The relation $\beta$ was introduced in 1970 by Koskas [11] and was mainly studied by Corsini [4] and Vougiouklis [18] and others. A useful tool in the study of theory of hyperstructures is the relation $\beta^*$, which is defined in a hypergroup or semi-hypergroup $(H, o)$, as the smallest equivalence relation on $H$ such that the quotient $H/\beta^*$ be a group (semi-group) under the hyperoperation $\beta^*(a) \oplus \beta^*(b) = \beta^*(c), c \in a \circ b$. This relation named as fundamental relation and $(H/\beta^*, \oplus)$ is called fundamental group. It is seen that, in hypergroups, $\beta^* = \beta^*$, in which $\beta^*$ is the transitive closure of $\beta$. In [10], Freni proved that $\beta$ is transitive on hypergroups. Therefore, $\beta^* = \beta^*$. In [1], Ameri introduced a process to obtain a group via the fundamental relation. The equivalence relation $\beta^*$ (or its extended forms in hyperrings and hypermodules and hyperlgebras) 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 $a \circ o b = H$, 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 $H/\beta^*$ 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 $|a \circ o b| \geq 2$. Set $|a \circ o b| = A$. 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 $(H_2, \ast)$ be two hypergroups. A map $f : H_1 \rightarrow H_2$ is called:

1. a homomorphism if $f(x \circ o y) \subseteq f(x) \ast f(y)$; for all $x, y \in H_1$.
2. a good homomorphism if $f(x \circ o y) = f(x) \ast f(y)$; 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 $\beta^*$ EQUIVALENCE CLASSES AND FUNDAMENTAL GROUP**

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

Let \( (a, b) \in \beta \) be a hypergroup of order \( n \) and \( P_m \subseteq \bigcup_{i=2}^{m-1} P_i \), for some \( m \geq 3 (m \in \mathbb{N}) \). Then for all \( k > m \), we have \( P_k \subseteq \bigcup_{i=2}^{m-1} P_i \).

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

Proof. Since \( |P^k(H)| = 2^n \cdot 2k \), \( P^k(H) \) for \( k \leq n \) is an upper bound for the sequence \( P_2 \subseteq P_2 \cup P_3 \subseteq \ldots \subseteq \bigcup_{i=2}^{m-1} P_i \subseteq \ldots \), that is, for all \( k \geq 2 \), we have \( P_k \subseteq \bigcup_{i=2}^{2^n} P_i \). 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 \( (a, b) \in \bigcup_{i=2}^{j} \beta_i \), for some \( 2 \leq j < 2^n \).

**Algorithm 2.5** Next, we come to the construction of the algorithm, which computes \( \beta \)-equivalence classes, according to the following description:

Step 1:
for each \( i \in H \) begin from 1
\[
Y(A) = \bigcup_{i \in H} \{ B_i \in T \mid \forall B_i \exists B_j \neq B_i : B_i \cap B_j \neq \emptyset \text{ and } \exists B_k : B_k \cap A \neq \emptyset \}
\]
\[\beta(i) = Y(A)\]
\[H = H \setminus Y(A)\]
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 \( P_3 \subseteq P_2 \), then by Lemma 2.1 the routine finishes. Otherwise, by Proposition 2.3, we have \( P_j \subseteq j-1 \cup_{i=2}^j P_i \) for some \( j \geq 4 \). Hence we move to the next step.
Step 2:
Let \( B = \{\beta(i1), \beta(i2), \ldots, \beta(ik)\} \) be the output of Step 1. We set:
\[P' = P_3\]
for \( j = 4 \) to \( 2^n - 1 \)
\[\text{while } \neg (P_j \subseteq \bigcup_{k=2}^{j-1} P_k)\]
\[P' = P' \cup P_j\]
\[T' = \{ B_i \in B \mid \exists B_j \neq B_i, \exists k_1 \in B_1, \exists k_2 \in B_2, \exists C \in P' : \{k_1, k_2\} \subseteq C \}\]
And we continue as bellow:
At the end of the process we obtain $\beta$-equivalence classes. Now in order to obtain the Fundamental group, $(H=\beta^*, \otimes)$, we have the following theorem:

**Theorem 2.6** Let $(H, \circ)$ be a hypergroup, such that $|H|=n$ and $k \in \mathbb{N}$ be the number of all $\beta$-equivalence classes. Let $\otimes$ be the operation on $H/\beta^*$ 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 $\beta$-equivalence classes, as the following:

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

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

**Proof.** In group $(B, \otimes)$ for one and only one $t_0 \in \{i_1, i_2, \ldots, i_k\}$ we have:

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

Since $t \in \beta(t)$ is an arbitrary member of $j \circ h$, so we have $j \circ h \cap \beta(t) \neq \emptyset$. Thus $j \circ h \cap \beta(t_0) \neq \emptyset$. Now it is sufficient to set $i_t = t_0$ 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 $\beta$-equivalence classes obtained through Algorithm 2.5. We present an algorithm, in brief, for computation of fundamental group, as follows:

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

    \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\}
}
```
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 \( n \leq 3 \) (\( n \in \mathbb{N} \)). 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 \( n \leq 3 \) (\( n \in \mathbb{N} \)) and isomorphism classes of them. This sub-program also enumerates quasi-hypergroups of order \( n \) and all \( \beta \)-equivalence classes.

```java
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 \( \exists B \in \mathbb{B}, B \cap j \circ h \neq \phi \)
            print ("\( \beta(i_j) \otimes \beta(i_h) = \) ", B)
         else
            \( h = h + 1 \)
      }
      \( j = j + 1 \)
   }
else
   print trivial case
end
```

```java
public class HyperGroupGenerator {
    public void hyperGroupGenerator(int n, int numberOfHyperGroup) {
        while (numberOfHyperGroup > 0) {
            for (int i = 1; i <= hyperStructure.size(); i++) {
                Point point1 = hyperStructure.getId(i);
                HashMap<Point, Set<Point>> productTable = new HashMap<>();
                for (int j = 1; j <= hyperStructure.size(); j++) {
                    Point point2 = hyperStructure.getId(j);
                    Set<Point> set = new HashSet<Point>(
                        powerSet.get(index[(i - 1) * n + (j - 1)]));
                    productTable.put(point2, set);
```
Computation of Fundamental Group of a Finite Hypergroup

\[ \text{Sigma J Eng & Nat Sci 9 (1), 107-125, 2018} \]

```java
} point1.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){
    quasiHyperGroupWriter.write(countQuasiHyperGroup + "");
}
```
public void hyperGroupGeneratorUpToIso(int n) throws Exception {
    File theDir = new File("uptoiso");
    if (!theDir.exists()) {
        theDir.mkdirs();
    }

    HashMap<Integer, Integer> numberOfEveryClass = new HashMap<Integer, Integer>();
    BufferedReader br = new BufferedReader(new FileReader("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 = new String[];
            Set<Point> hashSet = null;
            for (int i = 1; i <= hyperStructure.size(); i++) {
                productTable = br.readLine().split(" ");
                for (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.getHyperStructure(k).getByIndex(1));
                    }
                }
            }
        }
        point.addProduct(hyperStructure.getByIndex(j + 1), hashSet);
    }
}
for (int i = 0; i < numberOfIsoHyperGroup; i++) {
    HyperStructure hyperStructure2 = 
        getHyperStructureFromIni("uptoiso" + 1 + ".ini");
    if (isIsomorphic(hyperStructure2, hyperStructure)) {
        numberOfEveryClass.put(1, numberOfEveryClass.get(1) + 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(1),
        tmp.get(numberOfEveryClass.get(1)) + 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");
}
public HyperStructure getHyperStructureFromFile(String fileName) {
    try (InputStreamReader file = new
         InputStreamReader(new FileInputStream(fileName));
         BufferedReader br = new BufferedReader(file)) {
        rStructure hyperStructure = new HyperStructure(
            Integer.parseInt(br.readLine()));
        String[] productTable;
        Set<Point> hashSet = null;
        for (int i = 1; i <= hyperStructure.size(); i++) {
            productTable = br.readLine().split(" ");
            Point point = hyperStructure.getViewById(i);
            for (int j = 0; j < productTable.length; j++) {
                String product[] = productTable[j].split(",");
                HashSet = new HashSet<Point>();
                for (int k = 0; k < product.length; k++) {
                    hsSet.add(hyperStructure.getViewById(Integer.parseInt(product[k])));
                    addProduct(hyperStructure.getViewById(j + 1),
                        hsSet);
                }
            }
        }
        return hyperStructure;
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

public boolean isIsomorphic(HyperStructure hyperStructure1,
                             HyperStructure hyperStructure2) {
    if (hyperStructure1.size() != hyperStructure2.size()) {
        return false;
    }
    return true;
}
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.getByIdx(temp
                [point11
                .getId() - 1]);
            Point point22 = hyperStructure2.getByIdx(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;
                        }}}}?>
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 1032979 quasihypergroups, 23192 hypergroups and 3999 hypergroups (up to isomorphism) of order $n = 3$, distributed on 5 classes of cardinality. Moreover, the fundamental groups of these hypergroups are as follows:

One group is isomorphic to $\mathbb{Z}_3$, 9 of them are isomorphic to $\mathbb{Z}_2$ and the rest are trivial groups (0).

In the following we examine for $(H, \circ)$ given in Table 1.

**Table 1. Hypergroup (Example 1)**

<table>
<thead>
<tr>
<th>$\circ$</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1}</td>
<td>(2,3)</td>
<td>{1}</td>
</tr>
<tr>
<td>2</td>
<td>(2,3)</td>
<td>{1}</td>
<td>(2,3)</td>
</tr>
<tr>
<td>3</td>
<td>{1}</td>
<td>(2,3)</td>
<td>{1}</td>
</tr>
</tbody>
</table>

Equivalence Classes

$\beta(1) = \{1\}$, $\beta(2) = \{2, 3\}$
That is, \((H/\beta^n, \otimes) = \mathbb{Z}_2\).

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

```java
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();
    }
```
Here, we observe some examples of hypergroups, used by sub-program, Main, and the outputs:

**Example 2.** Consider hypergroupoids \((H_1, \circ)\) and \((H_2, \ast)\) as in Table 2:

<table>
<thead>
<tr>
<th>(\circ)</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1}</td>
<td>{2, 3}</td>
<td>{2, 3}</td>
</tr>
<tr>
<td>2</td>
<td>{2, 3}</td>
<td>{1}</td>
<td>{1}</td>
</tr>
<tr>
<td>3</td>
<td>{2, 3}</td>
<td>{1}</td>
<td>{1}</td>
</tr>
</tbody>
</table>

**Table 2.** Hyperstructures (Example 2)

For these two hyperstructures, we get the following outputs:

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

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

Hence \((H_1/\beta, \circ) = \mathbb{Z}_2\), while \((H_2, \ast)\) is not a hypergroup and actually is a quasihypergroup.

**Example 3.** Consider two hyperstructures of order 4 as follows
For hyperstructure \((H_1, \circ)\), it concludes that \((H_1, \circ)\) is a hypergroup, with the following equivalence classes

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

Hence \((H_1 / \beta^*, \&\)\) = \(Z_2\). For hyperstructure \((H_2, \ast)\) we get the following output:

**Hypergroup**

**Equivalence Classes**

\[
\begin{align*}
\beta(1) &= \{1\} \\
\beta(2) &= \{2, 3\} \\
\beta(4) &= \{4\}
\end{align*}
\]

That is, \((H_2 / \beta^*, \&\)\) = \(Z_3\).

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

**Table 4.** Hyperstructure \((H_1, \circ)\) (Example 4)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>{2,3}</td>
<td>{2,3}</td>
<td>{4}</td>
<td>{5}</td>
</tr>
<tr>
<td>2</td>
<td>{2,3}</td>
<td>2</td>
<td>{4}</td>
<td>{5}</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>{2,3}</td>
<td>4</td>
<td>{4}</td>
<td>{5}</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>{4}</td>
<td>{5}</td>
<td>{5}</td>
<td>{1}</td>
<td>{2,3}</td>
</tr>
<tr>
<td>5</td>
<td>{5}</td>
<td>{1}</td>
<td>{1}</td>
<td>{2,3}</td>
<td>{4}</td>
</tr>
</tbody>
</table>

**Table 5.** Hyperstructure \((H_2, \ast)\) (Example 4)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1}</td>
<td>{2}</td>
<td>{3,4}</td>
<td>{4}</td>
<td>{5}</td>
</tr>
<tr>
<td>2</td>
<td>{2}</td>
<td>{1, 5}</td>
<td>{3,4}</td>
<td>{3,4}</td>
<td>{2}</td>
</tr>
<tr>
<td>3</td>
<td>{3}</td>
<td>{3,4}</td>
<td>{1, 2}</td>
<td>{1, 2}</td>
<td>{2}</td>
</tr>
<tr>
<td>4</td>
<td>{3,4}</td>
<td>{3,4}</td>
<td>{1, 2}</td>
<td>{1, 2}</td>
<td>{5}</td>
</tr>
<tr>
<td>5</td>
<td>{5}</td>
<td>{2}</td>
<td>{5}</td>
<td>{5}</td>
<td>{1}</td>
</tr>
</tbody>
</table>

Then for hyperstructure \((H_1, \circ)\), it results:

**Hypergroup**

**Equivalence Classes**

\[
\begin{align*}
\beta(1) &= \{1\}, \beta(2) = \{2, 3\}, \beta(4) = \{4\}, \beta(5) = \{5\}
\end{align*}
\]

Therefore, \((H_1 / \beta^*, \&\)\) = \(Z_4\) (see Table 6).
Table 6. Operations (Example 4)

<table>
<thead>
<tr>
<th>⊗</th>
<th>β(1)</th>
<th>β(2)</th>
<th>β(4)</th>
<th>β(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>β(1)</td>
<td>β(1)</td>
<td>β(2)</td>
<td>β(4)</td>
<td>β(5)</td>
</tr>
<tr>
<td>β(2)</td>
<td>β(2)</td>
<td>β(4)</td>
<td>β(5)</td>
<td>β(1)</td>
</tr>
<tr>
<td>β(4)</td>
<td>β(4)</td>
<td>β(5)</td>
<td>β(1)</td>
<td>β(2)</td>
</tr>
<tr>
<td>β(5)</td>
<td>β(5)</td>
<td>β(1)</td>
<td>β(2)</td>
<td>β(4)</td>
</tr>
</tbody>
</table>

For hyperstructure \((H_2, \ast)\) we get the following output:

Hypergroupoid

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

Table 7. Hyperstructure \((H, \circ)\) (Example 5)

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1}</td>
<td>{2}</td>
<td>{3}</td>
<td>{4}</td>
<td>{5}</td>
</tr>
<tr>
<td>2</td>
<td>{2}</td>
<td>{1}</td>
<td>{3}</td>
<td>{4}</td>
<td>{5}</td>
</tr>
<tr>
<td>3</td>
<td>{3}</td>
<td>{3}</td>
<td>{1,2}</td>
<td>{4}</td>
<td>{5}</td>
</tr>
<tr>
<td>4</td>
<td>{4}</td>
<td>{4}</td>
<td>{4}</td>
<td>{6}</td>
<td>{1,2,3}</td>
</tr>
<tr>
<td>5</td>
<td>{5}</td>
<td>{5}</td>
<td>{1,2,3}</td>
<td>{6}</td>
<td>{4}</td>
</tr>
<tr>
<td>6</td>
<td>{6}</td>
<td>{6}</td>
<td>{6}</td>
<td>{5}</td>
<td>{4}</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>⊗</th>
<th>β(1)</th>
<th>β(4)</th>
<th>β(5)</th>
<th>β(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>β(1)</td>
<td>β(1)</td>
<td>β(4)</td>
<td>β(5)</td>
<td>β(6)</td>
</tr>
<tr>
<td>β(4)</td>
<td>β(4)</td>
<td>β(6)</td>
<td>β(1)</td>
<td>β(5)</td>
</tr>
<tr>
<td>β(5)</td>
<td>β(5)</td>
<td>β(1)</td>
<td>β(6)</td>
<td>β(4)</td>
</tr>
<tr>
<td>β(6)</td>
<td>β(6)</td>
<td>β(5)</td>
<td>β(4)</td>
<td>β(1)</td>
</tr>
</tbody>
</table>

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 \((H, \circ)\) of order 7 (as shown in Table 9).
Table 9. Hyperstructure (H, o) (Example 6)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1}</td>
<td>{2}</td>
<td>{3}</td>
<td>{4}</td>
<td>{5}</td>
<td>{6,7}</td>
<td>{6,7}</td>
</tr>
<tr>
<td>2</td>
<td>{2}</td>
<td>{1}</td>
<td>{5}</td>
<td>{6,7}</td>
<td>{3}</td>
<td>{4}</td>
<td>{4}</td>
</tr>
<tr>
<td>3</td>
<td>{3}</td>
<td>{6,7}</td>
<td>{1}</td>
<td>{5}</td>
<td>{4}</td>
<td>{2}</td>
<td>{2}</td>
</tr>
<tr>
<td>4</td>
<td>{4}</td>
<td>{5}</td>
<td>{6,7}</td>
<td>{1}</td>
<td>{2}</td>
<td>{3}</td>
<td>{3}</td>
</tr>
<tr>
<td>5</td>
<td>{5}</td>
<td>{4}</td>
<td>{2}</td>
<td>{3}</td>
<td>{6,7}</td>
<td>{1}</td>
<td>{1}</td>
</tr>
<tr>
<td>6</td>
<td>{6,7}</td>
<td>{3}</td>
<td>{4}</td>
<td>{2}</td>
<td>{1}</td>
<td>{5}</td>
<td>{5}</td>
</tr>
<tr>
<td>7</td>
<td>{6,7}</td>
<td>{3}</td>
<td>{4}</td>
<td>{2}</td>
<td>{1}</td>
<td>{5}</td>
<td>{5}</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>⊗</th>
<th>(\beta(1))</th>
<th>(\beta(2))</th>
<th>(\beta(3))</th>
<th>(\beta(4))</th>
<th>(\beta(5))</th>
<th>(\beta(6))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\beta(1))</td>
<td>(\beta(1))</td>
<td>(\beta(2))</td>
<td>(\beta(3))</td>
<td>(\beta(4))</td>
<td>(\beta(5))</td>
<td>(\beta(6))</td>
</tr>
<tr>
<td>(\beta(2))</td>
<td>(\beta(2))</td>
<td>(\beta(1))</td>
<td>(\beta(5))</td>
<td>(\beta(6))</td>
<td>(\beta(3))</td>
<td>(\beta(2))</td>
</tr>
<tr>
<td>(\beta(3))</td>
<td>(\beta(3))</td>
<td>(\beta(6))</td>
<td>(\beta(1))</td>
<td>(\beta(5))</td>
<td>(\beta(4))</td>
<td>(\beta(2))</td>
</tr>
<tr>
<td>(\beta(4))</td>
<td>(\beta(4))</td>
<td>(\beta(5))</td>
<td>(\beta(6))</td>
<td>(\beta(1))</td>
<td>(\beta(2))</td>
<td>(\beta(3))</td>
</tr>
<tr>
<td>(\beta(5))</td>
<td>(\beta(5))</td>
<td>(\beta(4))</td>
<td>(\beta(2))</td>
<td>(\beta(3))</td>
<td>(\beta(6))</td>
<td>(\beta(1))</td>
</tr>
<tr>
<td>(\beta(6))</td>
<td>(\beta(6))</td>
<td>(\beta(3))</td>
<td>(\beta(4))</td>
<td>(\beta(2))</td>
<td>(\beta(1))</td>
<td>(\beta(5))</td>
</tr>
</tbody>
</table>

Which \(|\beta(2)| = 2, |\beta(6)| = 3, \ldots\) Hence the fundamental group \((H/\beta^*, \otimes)\) is isomorphic 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)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1}</td>
<td>{2}</td>
<td>{3}</td>
<td>{4}</td>
<td>{5}</td>
<td>{6}</td>
<td>{7}</td>
<td>{8}</td>
</tr>
<tr>
<td>2</td>
<td>{2,6}</td>
<td>{1,5}</td>
<td>{4,8}</td>
<td>{3,7}</td>
<td>{2,6}</td>
<td>{1,5}</td>
<td>{4,8}</td>
<td>{3,7}</td>
</tr>
<tr>
<td>3</td>
<td>{3}</td>
<td>{4}</td>
<td>{5}</td>
<td>{6}</td>
<td>{7}</td>
<td>{8}</td>
<td>{1}</td>
<td>{2}</td>
</tr>
<tr>
<td>4</td>
<td>{4,8}</td>
<td>{3,7}</td>
<td>{2,6}</td>
<td>{1,5}</td>
<td>{4,8}</td>
<td>{3,7}</td>
<td>{2,6}</td>
<td>{1,5}</td>
</tr>
<tr>
<td>5</td>
<td>{5}</td>
<td>{6}</td>
<td>{7}</td>
<td>{8}</td>
<td>{1}</td>
<td>{2}</td>
<td>{3}</td>
<td>{4}</td>
</tr>
<tr>
<td>6</td>
<td>{2,6}</td>
<td>{1,5}</td>
<td>{4,8}</td>
<td>{3,7}</td>
<td>{2,6}</td>
<td>{1,5}</td>
<td>{4,8}</td>
<td>{3,7}</td>
</tr>
<tr>
<td>7</td>
<td>{7}</td>
<td>{8}</td>
<td>{1}</td>
<td>{2}</td>
<td>{3}</td>
<td>{4}</td>
<td>{5}</td>
<td>{6}</td>
</tr>
<tr>
<td>8</td>
<td>{4,8}</td>
<td>{3,7}</td>
<td>{2,6}</td>
<td>{1,5}</td>
<td>{4,8}</td>
<td>{3,7}</td>
<td>{2,6}</td>
<td>{1,5}</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>⊗</th>
<th>β(1)</th>
<th>β(2)</th>
<th>β(3)</th>
<th>β(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>β(1)</td>
<td>β(1)</td>
<td>β(2)</td>
<td>β(3)</td>
<td>β(4)</td>
</tr>
<tr>
<td>β(2)</td>
<td>β(2)</td>
<td>β(1)</td>
<td>β(4)</td>
<td>β(3)</td>
</tr>
<tr>
<td>β(3)</td>
<td>β(3)</td>
<td>β(4)</td>
<td>β(1)</td>
<td>β(2)</td>
</tr>
<tr>
<td>β(4)</td>
<td>β(4)</td>
<td>β(3)</td>
<td>β(2)</td>
<td>β(1)</td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>o</th>
<th>1</th>
<th>2, 3, 4, 5, 6</th>
<th>7, 8, 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1}</td>
<td>{2, 3}</td>
<td>H - {1}</td>
</tr>
<tr>
<td>2</td>
<td>{2}</td>
<td>{1, 3}</td>
<td>H - {2}</td>
</tr>
<tr>
<td>3</td>
<td>{3}</td>
<td>{1, 2}</td>
<td>H - {3}</td>
</tr>
<tr>
<td>4</td>
<td>{4}</td>
<td>{1, 2, 3, 5, 6}</td>
<td>H - {4}</td>
</tr>
<tr>
<td>5</td>
<td>{5}</td>
<td>{2, 3, 4, 5, 6}</td>
<td>H - {5}</td>
</tr>
<tr>
<td>6</td>
<td>{6}</td>
<td>{1, 2, 3, 4, 5}</td>
<td>H - {6}</td>
</tr>
<tr>
<td>7</td>
<td>{7}</td>
<td>H - {7}</td>
<td>H - {7}</td>
</tr>
<tr>
<td>8</td>
<td>{8}</td>
<td>H - {8}</td>
<td>H - {8}</td>
</tr>
<tr>
<td>9</td>
<td>{9}</td>
<td>H - {9}</td>
<td>H - {9}</td>
</tr>
</tbody>
</table>

And we get the following output:

HyperGroup
Equivalence Classes
\(β(1) = H\)

Trivial case, that is, \((H/β^∗, ⊗) = (0)\).

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


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


