

УДК 512.548+519.179.1

Finite Representation of Classes of Isomorphic Groupoids

Maxim N. Nazarov*

National Research University of Electronic Technology — MIET
Bld. 1, Shokin Square, Zelenograd, Moscow, 124498

Russia

Received 29.04.2015, received in revised form 11.05.2015, accepted 20.06.2015

We consider an alternative representation of finite groupoids in the form of hypergraphs with three-vertex edges. Automorphism classes of vertices and edges of this hypergraphs are linearly ordered by a natural indexing algorithm based on a maxi-code for three-dimensional adjacency matrix of the hypergraph. With respect of this indexing is constructed a finite set description for the classes of isomorphic groupoids.

Keywords: finite groupoids, hypergraphs, automorphism classes of groupoid elements.

DOI: 10.17516/1997-1397-2015-8-3-312–319

Introduction

Let us remind that groupoid D is a set with binary operation. We will call $[D]$ the class of isomorphic groupoids, if it is composed of all groupoids which are isomorphic to D . This class obviously has infinite number of elements, and, also, the individual groupoids from class $[D]$ are indistinguishable from the algebraic point of view. That in turn leads to the problem of choosing the etalon representative of class $[D]$, when we consider the practical application of groupoids (see for example [1]). It also seems logical that there could be a way to describe the class of isomorphic groupoids with some complete invariant in a similar way it is done with complete invariants in graph theory. Moreover, in the framework of [2] it was shown that for classical graphs one can define a complete invariant, which will be very close to graphs in its properties. If we construct a similar invariant for groupoids, then it can be used in practice as an alternative finite representation for the classes of isomorphic groupoids $[D]$, as well as a tool for choosing the etalon representative in class $[D]$.

We can associate with any arbitrary groupoid D an oriented[†] hypergraph $G = (D, E)$ by using a rule: $xy = z \Leftrightarrow (x, y, z) \in E$. Thus, in this hypergraph G the edges connect only three vertices and the set of edges E is uniquely defined by the operation on groupoid D . As a result, if we generalize the method of construction of complete graph invariant from the article [2] on this oriented hypergraphs with three-vertex edges, then we will in fact transfer this method on any arbitrary finite groupoids.

1. The indexation of automorphism classes

Definition 1. *Let $G = (D, E)$ be any arbitrary hypergraph, in which the number of vertices is $n = |D|$ and all the edges are a three-vertex ones $(v_i, v_j, v_k) \in E$. If we choose some order on*

*nazarov-maximilian@yandex.ru

© Siberian Federal University. All rights reserved

[†]We will call a hypergraph G oriented, if its edges are ordered tuples (x, y, z, \dots) . More information about the theory of classical (not oriented) hypergraphs can be obtained in [3].

the set of vertices D , then we can associate a **three-dimensional adjacency matrix** A with this hypergraph G by using a following rule:

$$A(i, j, k) = 1 \Leftrightarrow (v_i, v_j, v_k) \in E \quad \wedge \quad A(i, j, k) = 0 \Leftrightarrow (v_i, v_j, v_k) \notin E.$$

In the Fig. 1 is represented an example of two adjacency matrices which were constructed for two different orders on the set of vertices D of some hypergraph G . Let us note, that adjacency

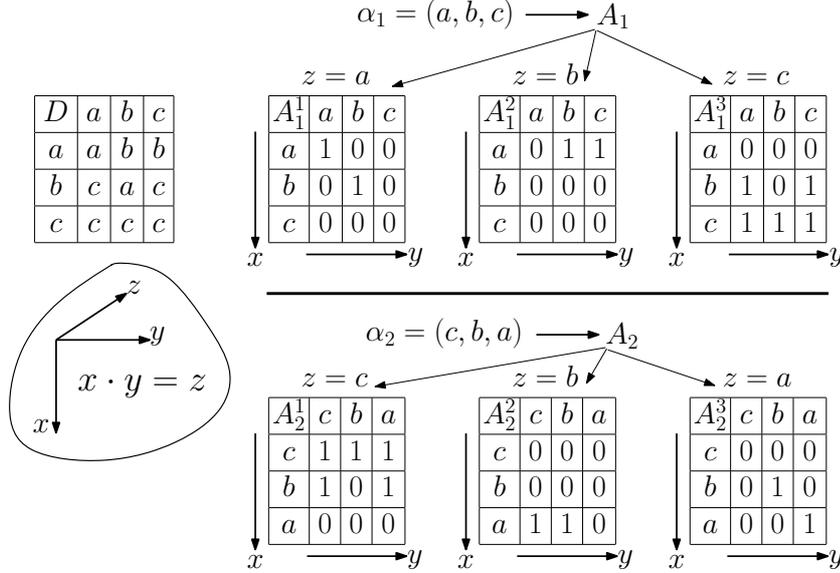


Fig. 1. Two adjacency matrices A_1 and A_2 for two ordering of the set D

matrices A will be equal for such permutations of vertices, which correspond to automorphisms of groupoid D (vertex automorphisms of its hypergraph G).

Definition 2. We will call **the code of adjacency matrix** A of finite hypergraph G such a number $\mu(A)$, which is obtained by converting adjacency matrix to binary number format $A(1, 1, 1) + A(1, 2, 1) \cdot 2^1 + \dots + A(1, n, 1) \cdot 2^{n-1} + \dots + A(n, n, n) \cdot 2^{n^3-1}$. This formula can be expressed in a compact form as $\mu(A) = \sum_{i,j,k=1}^n A(i, j, k) \cdot 2^{j-1+n(i-1)+n^2(k-1)}$.

Definition 3. The maximum of all adjacency matrix codes of hypergraph G we will call a **maxi-code** of hypergraph $\mu_{\max}(G)$. If for some order of vertices α the code of adjacency matrix $\mu(A) = \mu_{\max}(G)$, then order α **corresponds** to the maxi-code $\mu_{\max}(G)$ of hypergraph.

Maxi-code is a complete invariant of hypergraphs, since it can be used to reconstruct the adjacency matrix. As a result, maxi-code is also a complete invariant of groupoids D , and it can be viewed as a unique numerical identifier for the class of isomorphic groupoids $[D]$.

Position 1. If a two orderings of the set of vertices $\alpha_1 = (v_1^1, \dots, v_n^1)$ and $\alpha_2 = (v_1^2, \dots, v_n^2)$ correspond to the maxi-code $\mu_{\max}(G)$ of hypergraph G , then all the vertices in those orderings will be pairwise automorphic $v_k^1 \sim v_k^2$ for all $k = \overline{1, n}$.

Proof. The equality of code $\mu(A)$ for a two orderings $\alpha_1 = (v_1^1, \dots, v_n^1)$ and $\alpha_2 = (v_1^2, \dots, v_n^2)$ of the set of vertices D is possible only if the adjacency matrices are equal for this orderings. This in turn will yield that permutation of vertices $\psi = \begin{pmatrix} v_1^1 & \dots & v_n^1 \\ v_1^2 & \dots & v_n^2 \end{pmatrix}$ is an automorphism on G . \square

We will use a standard index notation $v_i = \alpha(i)$ for the sequences of vertices $\alpha = (v_1, \dots, v_n)$.

Definition 4. Let α be an ordering of the set of vertices, which corresponds to the maxi-code $\mu_{\max}(G)$. We will call the **number of automorphism class** \bar{v} such a natural number $N(\bar{v})$, which equals to the minimum index of vertex from class \bar{v} in the ordering α : $N(\bar{v}) = \min_{i: v \sim \alpha(i)} i$.

The correctness of Definition 4 for numbers of automorphism classes is a simple corollary from proposition 1. More strictly, if maxi-code of hypergraph corresponds to some different vertex sequences $\alpha_1 = (v_1^1, \dots, v_n^1)$ and $\alpha_2 = (v_1^2, \dots, v_n^2)$, then $v_k^1 \sim v_k^2$ for all $k = \overline{1, n}$. Hence, the minimum index of vertex from class \bar{v} would be the same for both α_1 and α_2 .

Corollary 1. The numbers $N(\bar{v})$ can be used to define a linear ordering on the set of automorphism classes of vertices: $\bar{v}_1 \leq \bar{v}_2 \Leftrightarrow N(\bar{v}_1) \leq N(\bar{v}_2)$.

Proof. By definition $N(v)$ are natural numbers, and according to proposition 1 they are equal only for automorphic vertices $v \sim u$. We can conclude from this that a set of numbers $N(\bar{v}_1), \dots, N(\bar{v}_k)$ is linearly ordered as a subset of natural numbers. As a result, the ordering on the set of automorphism classes $\bar{v}_1 \leq \bar{v}_2 \Leftrightarrow N(\bar{v}_1) \leq N(\bar{v}_2)$ would also be a linear one. \square

Definition 5. Let us define the **indexes of automorphism classes of vertices** $I(\bar{v})$ by using a following inductive rule:

1. $N(\bar{v}) = \min_{\bar{u}} N(\bar{u}) \Rightarrow I(\bar{v}) = 1$ – induction basis to define the first index.
2.
$$\left. \begin{array}{l} (N(\bar{v}) > N(\bar{u})) \wedge \\ (\forall u^* \neq u \ N(\bar{v}) > N(\bar{u}^*) \Rightarrow N(\bar{u}) > N(\bar{u}^*)) \end{array} \right\} \Rightarrow I(\bar{v}) = I(\bar{u}) + 1$$
 – inductive step.

We should note, that in the framework of [2] the definition of indexes $I(\bar{v})$ was not given in full, and in fact the numbers of classes $N(\bar{v})$ were used in all theorems. Wherein, in all illustrations and examples of article [2] $I(\bar{v})$ were used, but not $N(\bar{v})$, which can be considered a minor error.

By analogy with graph theory we will call two edges $(u_1, u_2, u_3), (v_1, v_2, v_3)$ of hypergraph automorphic, if there exists such an automorphism ψ , that $\forall i \ \psi(u_i) = v_i$. The fact that it is necessary for hypergraphs and groupoids to define the classes of automorphic edges $(\bar{u}_1, \bar{u}_2, \bar{u}_3)$ in addition to the classes of automorphic vertices \bar{v} is illustrated in Fig. 2.

D	a	b	c	d	e	f	$(a, b, a) \sim (b, a, b) \sim (a, c, a)$
a	a	a	a	a	e	f	$(a, a, a) \sim (b, b, b) \sim (c, c, c)$
b	b	b	b	d	b	f	
c	c	c	c	d	e	c	$(a, b, a) \approx (a, d, a)$
d	d	b	c	d	d	d	$(a, b, a) \approx (a, e, e)$
e	a	e	c	e	e	e	
f	a	b	f	f	f	f	

$$a \sim b \sim c \sim d \sim e \sim f \Rightarrow \bar{a} = D$$

Fig. 2. An example of groupoid D , for which hypergraph G will have only one class of automorphic vertices $\bar{a} = D$ and more than one classes of automorphic edges $(a, b, a) \approx (a, e, e)$

Definition 6. Let α be an ordering of the set of vertices, which corresponds to the maxi-code $\mu_{\max}(G)$ of G . We will call the **number of automorphism class** $(\bar{x}, \bar{y}, \bar{z})$ such a natural number $N(\bar{x}, \bar{y}, \bar{z})$, which equals to $N(\bar{x}, \bar{y}, \bar{z}) = \min_{\substack{(i,j,k): A(i,j,k)=1 \\ (\alpha(i), \alpha(j), \alpha(k)) \sim (x,y,z)}} j + (i-1)n + (k-1)n^2$.

Definition 7. By analogy with Definition 5 we will introduce the **index of automorphism class of edges** $I(\bar{x}, \bar{y}, \bar{z})$ up to replacing the classes of vertices with the classes of edges.

Theorem 1. If $G \cong H$, then for any two vertices $u \in V(G)$, $v \in V(H)$ the equality $I(\bar{u}) = I(\bar{v})$ holds if and only if there exists isomorphism $\phi : V(G) \rightarrow V(H)$, such that $\phi(u) = v$.

Proof. Let us prove first, that from $\phi(u) = v$, where ϕ is an isomorphism, will follow the equality of indexes $I(\bar{u}) = I(\bar{v})$. If two hypergraphs are isomorphic, then maxi-codes of this hypergraphs are equal $\mu_{\max}(G) = \mu_{\max}(H)$. In this case we can represent the isomorphism of hypergraphs in the form $\phi = \begin{pmatrix} u_1 \dots u_n \\ v_1 \dots v_n \end{pmatrix}$, where $\alpha_1 = (u_1, \dots, u_n)$ is some order, which corresponds to maxi-code of G and $\alpha_2 = (v_1, \dots, v_n)$ corresponds to maxi-code of H . We know that $N(\bar{u})$ is a natural number, and hence $\alpha_1(N(\bar{u}))$ defines some vertex $u^* = \alpha_1(N(\bar{u}))$ in hypergraph G . This vertex u^* has two properties: $u \sim u^*$ and it has minimum index in the ordering α_1 among all the vertices from automorphism class \bar{u} . If we assume, that $N(\bar{u}) \neq N(\bar{v})$, then we will get $\phi(u^*) \approx \phi(u)$, which results in contradiction: $\phi(u^*) \approx \phi(u)$ and $u \sim u^*$. As a result, for all isomorphic vertices $\phi(u) = v$ their numbers of automorphism classes will be equal $N(\bar{u}) = N(\bar{v})$, and therefore their indexes of automorphism classes will also be equal $I(\bar{u}) = I(\bar{v})$.

Let us prove the second part of theorem, that from $G \cong H$ and the equality of class indexes $I(\bar{u}) = I(\bar{v})$ will follow the existence of such isomorphism $\phi : \phi(u) = v$. Since $G \cong H$ we know that there must exist some isomorphism $\phi_0 : V(G) \rightarrow V(H)$. For the image $v^* = \phi_0(u)$ of u by using the first part of this theorem we obtain $I(\bar{v}^*) = I(\bar{u}) = I(\bar{v})$. We can conclude from it that vertices v, v^* are automorphic $v \sim v^*$ on hypergraph H . Let us denote the automorphism, which maps this vertices one onto another as $\psi : \psi(v^*) = v$. As a result, the final isomorphism can be defined as a composition $\phi = \phi_0 \circ \psi$, and we get for vertices u and v the equality $v = \psi(\phi_0(u))$. Since the mapping ϕ is a composition of two isomorphisms, then it is also an isomorphism. \square

Theorem 2. If $G \cong H$, then for any two edges $(u_1, u_2, u_3) \in E(G)$, $(v_1, v_2, v_3) \in E(H)$ the equality $I(\bar{u}_1, \bar{u}_2, \bar{u}_3) = I(\bar{v}_1, \bar{v}_2, \bar{v}_3)$ holds if and only if there exists hypergraph isomorphism $\phi : V(G) \rightarrow V(H)$ such, that $\phi(u_1) = v_1$, $\phi(u_2) = v_2$ and $\phi(u_3) = v_3$.

Proof. The sufficiency can be proven by full analogy with Theorem 1. If we assume, that there exists isomorphism $\phi : V(G) \rightarrow V(H)$, such that $\phi(u_1) = v_1$, $\phi(u_2) = v_2$ and $\phi(u_3) = v_3$, then automorphism class numbers would be equal $N(\bar{u}_1, \bar{u}_2, \bar{u}_3) = N(\bar{v}_1, \bar{v}_2, \bar{v}_3)$. This equality of numbers $N(\bar{u}_1, \bar{u}_2, \bar{u}_3) = N(\bar{v}_1, \bar{v}_2, \bar{v}_3)$ for all classes of isomorphic edges will in turn imply the equality of indexes $I(\bar{u}_1, \bar{u}_2, \bar{u}_3) = I(\bar{v}_1, \bar{v}_2, \bar{v}_3)$.

The necessity could also be proven by analogy with Theorem 1. Let $G \cong H$ and $I(\bar{u}_1, \bar{u}_2, \bar{u}_3) = I(\bar{v}_1, \bar{v}_2, \bar{v}_3)$. For the images of isomorphism $v_1^* = \phi_0(u_1)$, $v_2^* = \phi_0(u_2)$ and $v_3^* = \phi_0(u_3)$ using the first part of this theorem we obtain $I(\bar{u}_1, \bar{u}_2, \bar{u}_3) = I(\bar{v}_1, \bar{v}_2, \bar{v}_3) = I(\bar{v}_1^*, \bar{v}_2^*, \bar{v}_3^*)$. We can conclude from it that edges (v_1, v_2, v_3) and (v_1^*, v_2^*, v_3^*) are automorphic $(v_1, v_2, v_3) \sim (v_1^*, v_2^*, v_3^*)$. Let us denote the automorphism, which maps this vertices one onto another as $\psi : \forall i \psi(v_i^*) = v_i$. As a result, the final isomorphism would be a composition $\phi = \phi_0 \circ \psi$. \square

2. The construction of linear notation for groupoids

The direct consequence of Theorems 1 and 2 is the fact, that indexes $I(\bar{v})$ and $I(\overline{v_1, v_2, v_3})$ are unique identifiers for the automorphism classes. Thus, when defining an invariant for classes of isomorphic groupoids, this indexes could be used as a replacement for vertices and edges.

In constructing the announced invariant we will use an algorithmic[‡] approach, which is quite similar to the definition of the linear notation $I[G]$ of graphs from article [2].

Definition 8. We will call the **symmetric linear notation** of groupoid D such a string of symbols $\mathfrak{L}(D)$, which is defined on the basis of four rules:

Rule 1: The string $\mathfrak{L}(D)$ begins with the element v for which automorphism index is $I(v) = 1$.

Rule 2: When any element $x \in D$ is included into the notation $\mathfrak{L}(D)$, then together with x we will include the results ($z = x \cdot y$) of multiplication of x on all the groupoid elements $y \in D$.

For this we will use a following string: $S = x \left[\underset{n}{y} \left[\underset{n+1}{\dots} \dots \right] \xrightarrow{i} z[\dots] \dots \right]$, if x, y and z have not appeared in the substring of linear notation $\mathfrak{L}(D)$ to the left of this entry S .

Rule 3: When for some vertex u , which is already present in the left substring of linear notation $\mathfrak{L}(D) = \left[\dots \left[\underset{n}{\dots} \dots \right] \dots \right]$, it is required to add one more instance of u to $\mathfrak{L}(D)$, then instead of it we will use a special replacement codes $\#1, \#2, \#3, \dots, \#m$. The designation $\#1$ defines the vertex v to the left of the first opening bracket $\left[\underset{1}{\dots} \right]$ in the linear notation $\mathfrak{L}(D)$, while code $\#2$ defines the vertex before the second opening bracket and so on for the last opening bracket.

Rule 4: For any element $x \in D$, when choosing the order in which we will include the elements $y_1 \xrightarrow{i_1} z_1, \dots, y_2 \xrightarrow{i_2} z_2, \dots$ and special symbols $\#m \xrightarrow{i} \dots$ in the notation $x[\dots; \dots; \dots]$, we will use a following list of priorities.

1. The top priority will be associated with code $\#1$. The next in priority will be code $\#2$, and so on for all other codes up to the last $\#m$.
2. The elements $y \in D$ with minimum value of indexes $I(\bar{y})$ will be included after the codes $\#m$ in the notation $x[\dots; \dots; \dots]$.
3. If two elements y_1 and y_2 have the same index $I(\bar{y}_1) = I(\bar{y}_2)$, then we will choose the element y_1 , if the indexes of edges are $I(\bar{x}, y_1, z_1) < I(\bar{x}, y_2, z_2)$.
4. When indexes of edges and vertices are the same for y_1 and y_2 we will turn our attention to the potential right multiplication factors of y_1 and y_2 . Let y_1 yield a first result, which is different from the elements of codes $\#1, \dots, \#m$, by right factor multiplication on element of the code $\#i$. In addition, let the code $\#j$ hold the same property for y_2 . We will consider the element y_1 to be of higher priority than y_2 when $i < j$. In case of equality $i = j$ we will continue recursively this process for the second code value, and so on.
5. If all the positions of non-coded result elements on the step 4 have been the same for both y_1 and y_2 , then we will consider the order in which codes $\#1, \dots, \#m$ would appear as a results of multiplication of y_1 and y_2 on the elements of any other codes $\#1, \dots, \#m$. Let y_1 yield a code $\#1$ when multiplied ($y_1 \times \#i \rightarrow \#1$) on the element of code $\#i$, and let y_2 yield $\#1$ when multiplied ($y_2 \times \#j \rightarrow \#1$) on the element of code $\#j$. We will choose[§] the element y_1 to be of higher priority than y_2 when $i < j$. In case of equality $i = j$ we will continue recursively this process for code $\#2$, and so on up to the last code $\#m$.

[‡]It should also be stated, that this algorithm is similar in some extent to the algorithm of SMILES standard for encoding molecular structures (see [4]).

[§]If we have found i , but the value j does not exist, then we shall assume $j = \infty$.

6. With the failure of priority criteria 1–5 it can be guaranteed, that notation segments $y_1[\dots z_1^1[\dots]]$ and $y_2[\dots z_2^1[\dots]]$ will have the same position for the first elements z_1^1 and z_2^1 , which are different from codes $\#1, \dots, \#m$. We will then recursively apply all the priority criteria 1–5 to z_1^1 and z_2^1 . Moreover, we will temporally associate for y_1 a new code $\#(m+1)$ with z_1^1 and for y_2 we will associate the same code $\#(m+1)$ with z_2^1 . If such iteration process will end with overall equality in priorities, then we will do the same with the second ones z_1^2 and z_2^2 , and so on up to the last z_1^k and z_2^k , different from codes $\#1, \dots, \#m$.

Definition 9. We will call the **linear notation of class of isomorphic groupoids** $[D]$ such a string $I[D]$, which is obtained from any symmetric linear notation $\mathfrak{L}(D)$ by replacing all the elements v in $\mathfrak{L}(D)$ with their indexes $I(\bar{v})$.

Let us illustrate the definition of linear notation $I[D]$ with the examples in Fig. 3 of left-zero semigroup L and right-zero semigroups R , as well as some arbitrary groupoid D .

D	a	b	c
a	a	b	b
b	c	a	c
c	c	c	c

R	a	b	c
a	a	b	c
b	a	b	c
c	a	b	c

L	a	b	c
a	a	a	a
b	b	b	b
c	c	c	c

$$I(a) = 3, I(b) = 2, I(c) = 1$$

$$I(a) = I(b) = I(c) = 1$$

$$I(a) = I(b) = I(c) = 1$$

$$I[D]=1[_1\#1^1\rightarrow\#1; 2[_2\#1^4\rightarrow\#1; \#2^8\rightarrow 3[_3\#1^6\rightarrow\#2; \#2^7\rightarrow\#2; \#3^9\rightarrow\#3]; \#3^5\rightarrow\#1]^2\rightarrow\#1; \#3^3\rightarrow\#1]$$

$$I[R]=1[_1\#1^1\rightarrow\#1; 1[_2\#1^2\rightarrow\#1; \#2^1\rightarrow\#2; 1[_3\#1^2\rightarrow\#1; \#2^2\rightarrow\#2; \#3^1\rightarrow\#3]^2\rightarrow\#3]^2\rightarrow\#2; \#3^2\rightarrow\#3]$$

$$I[L]=1[_1\#1^1\rightarrow\#1; 1[_2\#1^2\rightarrow\#2; \#2^1\rightarrow\#2; 1[_3\#1^2\rightarrow\#3; \#2^2\rightarrow\#3; \#3^1\rightarrow\#3]^2\rightarrow\#2]^2\rightarrow\#1; \#3^2\rightarrow\#1]$$

Fig. 3. The example of linear notations of classes of isomorphic groupoids for D , R and L

Definition 10. For any arbitrary linear notation $I[D]$ the **colouring** of any abstract **element** j **into colour** α would be such a string $I_{(j,\alpha)}[D]$, which is obtained from $I[D]$ by replacing the first[¶] appearance of automorphism index j with pair $\langle j, \alpha \rangle$. The colouring for a greater number of elements and colours could be defined by induction, as a consequential colouring of single elements. Thus, it is sufficient to require, that on each iteration of this process we would choose only the elements, which were not coloured on previous iterations.

By using the recursive colouring procedure one could reconstruct the original groupoid D^* from linear notation $I[D]$ up to isomorphism $D^* \cong D$. If the groupoid D has n elements, then for its reconstruction from $I[D]$ we would have to colour the abstract elements of $I[D]$ (the numbers of automorphism classes) into n different colours. For example, to reconstruct the groupoids from Fig. 3 we would colour elements of $I[D]$, $I[L]$ and $I[R]$ into three colours: a , b and c .

Theorem 3. Any arbitrary finite groupoids D_1 and D_2 are isomorphic if and only if their linear notations are equal ($I[D_1] = I[D_2]$).

[¶]It is important to note, that only the first appearance of index j is replaced, while all other instances of j would remain untouched.

Proof. Let us first prove the necessity, that from $D_1 \cong D_2$ will follow the equality of linear notation of classes of isomorphic groupoids $I[D_1] = I[D_2]$. We know, that in construction of $I[D]$ there is only one situation, which can possibly lead to ambiguity in the definition — it is the choice of the inclusion order for elements v_1 and v_2 with the rule 4 of definition 8, when all six priority criteria failed to distinguish v_1 and v_2 . Considering the recursive search on stage 6 of rule 4 we can assume, that after the replacement of first occurring elements v with $I(v)$, and all the reoccurring elements with codes $\#1, \dots, \#n$ the strings $I(v_1)[\dots]$ and $I(v_2)[\dots]$ would be just the same. If it is not, and some indexes $I(v)$ are different in $I(v_1)[\dots]$ and $I(v_2)[\dots]$, then vertices v_1 and v_2 would have a different priorities after criterion 2 of rule 4 is applied recursively. Analogously, if codes $\#1, \dots, \#n$ have different positions in strings $I(v_1)[\dots]$ and $I(v_2)[\dots]$, then v_1 and v_2 would have different priorities after criteria 4 and 5 of rule 4 are applied recursively. As a result, the definition of $I[D]$ is independent from the order in which we include this vertices v_1 and v_2 . Hence, the notation $I[D]$ would be same for all the isomorphic groupoids.

Let us prove the sufficiency, that from the equality $I[D_1] = I[D_2]$ of linear notation of classes of isomorphic groupoids will follow $D_1 \cong D_2$. To do this we get back from $I[D_1]$ to $\mathfrak{L}(D_1)$ by colouring the $I[D_1]$ with elements of (D_1, \cdot) . Moreover, we choose such colouring $\mathfrak{L}(D_1)$, which corresponds to the operation on (D_1, \cdot) , potentially exhausting all $n!$ options. Then we repeat the same procedure for the groupoid (D_2, \circ) and obtain $\mathfrak{L}(D_2)$ for it. Let us construct the $\psi : D_1 \rightarrow D_2$ explicitly by mapping all the elements from $\mathfrak{L}(D_1)$ to the elements with the same string positions in $\mathfrak{L}(D_2)$. If we assume that resulting mapping ψ is not an isomorphism, then we will immediately get a contradiction. Indeed, the violation of isomorphism condition would mean that positions of codes $\#1, \dots, \#n$ are different in $\mathfrak{L}(D_1)$ and $\mathfrak{L}(D_2)$, which is impossible due to the equality $I[D_1] = I[D_2]$ and unambiguous^{||} definition of $I[D]$ on the basis of $\mathfrak{L}(D)$. \square

Remark 1. The basic algorithms for manipulation of linear notations $I[D]$ belong to the following classes of complexity.

1. The transition from D to linear notation $I[D]$ is a problem from class NP, because it requires to compute the maxi-code of three dimensional adjacency matrix.
2. The test on isomorphism for two linear notations $I[D_1]$ and $I[D_2]$ is a problem of class P, as long as it can be reduced to the trivial check of equality $I[D_1] = I[D_2]$.
3. The transition from linear notation $I[D]$ to groupoid D is a problem from class P, because it could be performed with a recursive colouring of the abstract elements of $I[D]$.

Conclusion

The first and the most obvious practical application of linear notations $I[D]$ would be in optimization of isomorphism test for groupoids. In fact, if $I[D]$ is stored in computer memory instead of groupoid D , then this test is reduced to the trivial check of equality $I[D_1] = I[D_2]$. As a result, this algorithm for $I[D]$ will be of $O(|D|^2)$ complexity, which would be essentially better than its counterpart for groupoids D with factorial $O(|D|!)$ complexity.

The second application amounts to the adaptation of $I[D]$ for the algorithms of exhaustive search for finite groupoids of small order. As it is well known, the brute force enumeration of Cayley tables of groupoids D would be computationally ineffective starting from $|D| = 4$. However, if instead of tables we would do the exhaustive search for $I[D]$ of some fixed order

^{||}The indexes $I(\bar{v})$ are used only to replace the elements v , while codes $\#m$ are preserved in the notation $I[D]$.

$n = |D|$, then the task would be greatly simplified (the number of groupoids as a function of n is given in [5]).

Let us note, that it is an open problem, whether any correspondence exists between the structure of notation $I[D]$ and such classical algebraic notions as commutativity, associativity, presence of zeros and unities on groupoids. In addition to this, it could also be of some interest for the specialists in the field of n -ary algebras to obtain a generalization of $I[D]$ to the n -ary groupoids (D, f) , where $f : D \times \dots \times D \rightarrow D$.

The author is grateful to A. V. Reshetnikov and N. V. Suvorova for critical comments and discussion of this article.

References

- [1] M.N.Nazarov, A self-induced metric on groupoids and its application to the analysis of cellular interactions in biology, *J. Math. Sci.*, **206**(2015), no. 5, 561–569.
- [2] M.N.Nazarov, Alternative approaches to the description of classes of isomorphic graphs, *Prikl. Diskr. Mat.*, **3**(2014), 86–97 (in Russian).
- [3] A.A.Zykov, Hypergraphs, *Russian Mathematical Surveys*, **29**(1974), no. 6, 89–154.
- [4] D.Weininger, et al, SMILES. 2. Algorithm for generation of unique SMILES notation, *J. Chem. Inf. Comput. Sci.*, **29**(1989), no. 2, 97–101.
- [5] M.A.Harrison, The number of isomorphism types of finite algebras, *Proc. Amer. Math. Soc.*, **17**(1966), 731–737.

Представление классов изоморфных группоидов на основе конечных множеств

Максим Н. Назаров

Для конечных группоидов рассматривается альтернативное представление в виде гиперграфов с трёх-вершинными рёбрами. Классы автоморфных вершин и рёбер данных гиперграфов линейно упорядочиваются с помощью алгоритма естественной индексации на основе макси-кода для трёхмерной матрицы смежности гиперграфа. Опираясь на полученную индексацию, строится конечно-множественное описание для классов изоморфных группоидов.

Ключевые слова: конечные группоиды, гиперграфы, классы автоморфных элементов группоидов.