CZU: 371.3:004

ALGORITMUL VERIFICĂRII IZOMORFISMELOR DE GRUPOIZI DIN PERSPECTIVA INFORMATICII

Liubomir CHIRIAC, dr. hab., prof. univ.
Aureliu DANILOV, drd.
Universitate de Stat Tiraspol

Rezumat. În acest articol, sunt examinate unele abordări metodice privind studierea și aplicarea noțiunilor de omotopism și izomorfizm la rezolvarea problemelor din informatică. Pe baza noțiunilor de *omotopism* s-a elaborat un algoritm de verificarea izomorfismelor de grupoizi. În expunerea subiectului sau folosit o abordare inderdisplinară: noțiuni din fundamentale algebrice, algoritmică și programare.

Cuvinte cheie: grupoid, omotopism, izotopism, izomorfism, algoritm.

Abstract. In this article, we examine some methodological approaches for studying and applying the notions of homotopism and isomorphism to solve problems in computer science. Based on the notions of homotopism, an algorithm for verifying goupoids isomorphisms was developed. In the presentation of the subject, an indisciplinarity approach is used: notions from algebraic fundamentals, algorithms and programming.

Keywords: groupoid, homotopism, isotopism, isomorphism, algorithm.

1. Noțiuni și concepte de bază

La studierea informaticii, înțelegerea și aplicarea noțiunilor și conceptelor de bază din algebra abstractă este un proces absolut necesar, în mod special pentru învățarea compartimentelor moderne din criptografie. Astfel, la studierea cursului pentru ciclul II, ”Structuri alegbrice pe calculator” sunt examinate diverse noțiuni algebrice care au o aplicație semnificativă la soluționarea problemelor practice. În articolul respectiv vom examina metodologia tratării noțiunii de izomorfism de grupoizi din perspectiva informaticii care, în opnia noastră, reprezintă o conexiune interdisciplinară pronunțată.

Prin izomorfism (din limba greacă: ἴσος isos "egal", și μορφή morphe "formă") [3] în matematică se înțelege o funcție între două mulțimi peste care s-au definit câte o structură algebrică care satisface următoarele două condiții [1, 2]:

- este morfism (adică păstrează structura algebrică, în sensul că orice relație ar exista între niște elemente din prima mulțime, relația respectivă se regăsește între elementele corespunzătoare - imagini prin funcția studiată - din a doua mulțime);
- admite un alt morfism care o "inversează" (formal, pentru $\mathbf{f} : \mathbf{A} \to \mathbf{B}$, să existe $\mathbf{g} : \mathbf{B} \to \mathbf{A}$ morfism astfel încât $\mathbf{g} \circ \mathbf{f} = 1_A$ și $\mathbf{f} \circ \mathbf{g} = 1_B$).

Notă: această condiție necesită ca **f** să fie bijectivă, dar cere în plus ca inversa ei să fie tot morfism.

Cuplul (**Q**, •) format dintr-o mulțime nevidă **Q** și o operația algebrică „ • ” pe Q se numește *grupoid*. Fie (**A**, *) și (**B**, ∘) doi grupoizi. Vom numi *morfism* (ori omomorfism) al grupoidului (**A**, *) în grupoidul (**B**, ∘) orice funcție $\mathbf{f} : \mathbf{A} \to \mathbf{B}$ care satisface condiția:

$$\mathbf{f}(\mathbf{x} * \mathbf{y}) = \mathbf{f}(\mathbf{x}) \circ \mathbf{f}(\mathbf{y}), \forall\, \mathbf{x}, \mathbf{y} \in \mathbf{A}.$$

Un morfism bijectiv se numește *izomorfism*.

2. Izotopisme și izomorfisme de grupoizi

Reamintim următoarele noțiuni algebrice.

Grupoidul (**Q**, *) se numește *quasigrup* dacă fiecare din ecuațiile **a** * **x** = **b** și **y** * **a** = **b** are exact o singură soluție în **Q** pentru orice **a**, **b** ∈ **Q**.

Quasigrupul cu unitate se numește *buclă* (în engleză *loop*). Bucla asociativă se numește *grup*.

O noțiune importantă pentru teoria grupoizilor care generalizează noțiunea de izomorfism este noțiunea de izotopism. Să examinăm esența acestei noțiuni.

Fie (**A**, *) și (**B**, ∘) doi grupoizi. Se numește *omotopism* (sau homotopism) al grupoidului (**A**,*) în grupoidul (**B**,∘) orice triplet ordonat (α, β, γ) de funcții de la **A** la **B** care satisface următoarea condiție: $\alpha(\mathbf{x}) \circ \beta(\mathbf{y}) = \gamma(\mathbf{x} * \mathbf{y})$ sau $\gamma^{-1}(\alpha(\mathbf{x}) \circ \beta(\mathbf{y})) = \mathbf{x} * \mathbf{y}$, pentru ∀ x, y ∈ A. (1)

Prin *izotopism* al lui (**A**, *) pe (**B**, ∘) se înțelege un omotopism (α, β, γ) în care toate cele trei componente α, β și γ sunt bijecții ale lui A pe B [1, 2].

Grupoidul (**A**, *) se zice că este *izotop* cu (**B**, ∘) dacă există măcar un izotopism (α, β, γ) al lui (**A**, *) pe (**B**, ∘). Evident, dacă există două operații „ * ” și „ ∘ ” definite pe una și aceeași mulțime **Q**, atunci vom spune că operația „ * ” este un izotop a operației „ ∘ ” dacă există așa un triplet ordonat de permutări (α, β, γ) a mulțimii **Q** astfel, încât $\alpha(\mathbf{x}) \circ \beta(\mathbf{y}) = \gamma(\mathbf{x} * \mathbf{y})$, pentru ∀ **x**, **y** ∈ **Q**.

Tripletul ordonat **T** = (α, β, γ) se numește, după cum am mai spus, izotopie.

EXEMPLUL 1. Fie (**Q**, *) un quasigrup, determinat de următorul tabel Cayley:

*	1	2	3	4
1	4	1	2	3
2	3	4	1	2
3	2	3	4	1
4	1	2	3	4

Fie α, β și γ sunt trei permutări arbitrare ale mulțimii **Q**. Atunci aplicând permutarea α elementelor de pe linia de bordare, permutarea β elementelor de pe coloana de bordare și permutarea γ^{-1} a elementelor din interiorul tabelei, se obține o nouă lege de compoziție „ ∘ ” pe **Q** și este clar că (**Q**, ∘) este izotop cu quasigrupul (**Q**, *).

Astfel, considerăm:

$$\alpha = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix} \uparrow, \beta = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}, \gamma^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}.$$

În rezultat, obținem următoarele transformări:

*	1	2	3	4
1	4	1	2	3
2	3	4	1	2
3	2	3	4	1
4	1	2	3	4

$\xrightarrow{\alpha}$

$\circ_\alpha$	1	2	3	4
1	3	4	1	2
2	2	3	4	1
3	1	2	3	4
4	4	1	2	3

$\xrightarrow{\beta}$

$\circ_\beta$	1	2	3	4
1	2	1	4	3
2	1	4	3	2
3	4	3	2	1
4	3	2	1	4

$\xrightarrow{\gamma^{-1}}$

$\circ_{\gamma^{-1}}$	1	2	3	4
1	1	2	3	4
2	2	3	4	1
3	3	4	1	2
4	4	1	2	3

Dacă toate 3 permutări coincid: α = β = γ, atunci izotopia se transformă în *izomorfism* [1]. În acest caz vom scrie

$$\alpha(\mathbf{x}) \circ \alpha(\mathbf{y}) = \alpha(\mathbf{x} * \mathbf{y}).$$

EXEMPLUL 2. Fie (**Q**, *) un quasigrup, determinat de următorul tabel Cayley:

*	1	2	3
1	1	2	3
2	2	3	1
3	3	1	2

Fie α = β = γ sunt trei substituții echivalente ale mulțimii **Q** și

$$\alpha = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

Observăm că α(**1**)=2, α(**2**)=3 și α(**3**)=1. Atunci aplicând consecutiv substituțiile α, β și γ^{-1} asupra grupoidului (**Q**, *) obținem o nouă lege de compoziție „ ∘ ” pe **Q** și este clar că (**Q**, ∘) este izomorf cu quasigrupul (**Q**, *).

Pe baza formulei (1) calculăm:

x	**y**	α(**x**)	α(**y**)	α(**x**) ∘ α(**y**)	α(**x** * **y**)	α(**x**) ∘ α(**y**) = α(**x** * **y**)
1	1	α(**1**)=2	α(**1**)=2	2 ∘ 2	α(**1** * **1**) = α(**1**) = 2	2 ∘ 2 = 2
1	2	α(**1**)=2	α(**2**)=3	2 ∘ 3	α(**1** * **2**) = α(**2**) = 3	2 ∘ 3 = 3
1	3	α(**1**)=2	α(**3**)=1	2 ∘ 1	α(**1** * **3**) = α(**3**) = 1	2 ∘ 1 = 1
2	1	α(**2**)=3	α(**1**)=2	3 ∘ 2	α(**2** * **1**) = α(**2**) = 3	3 ∘ 2 = 3
2	2	α(**2**)=3	α(**2**)=3	3 ∘ 3	α(**2** * **2**) = α(**3**) = 1	3 ∘ 3 = 1
2	3	α(**2**)=3	α(**3**)=1	3 ∘ 1	α(**2** * **3**) = α(**1**) = 2	3 ∘ 1 = 2
3	1	α(**3**)=1	α(**1**)=2	1 ∘ 2	α(**3** * **1**) = α(**3**) = 1	1 ∘ 2 = 1
3	2	α(**3**)=1	α(**2**)=3	1 ∘ 3	α(**3** * **2**) = α(**1**) = 2	1 ∘ 3 = 2
3	3	α(**3**)=1	α(**3**)=1	1 ∘ 1	α(**3** * **3**) = α(**2**) = 3	1 ∘ 1 = 3

Obținem operația:

$\circ_{\gamma^{-1}}$	1	2	3
1	**3**	**1**	**2**
2	**1**	**2**	**3**
3	**2**	**3**	**1**

Sau folosind metoda: aplicând permutarea α elementelor de pe linia de bordare, permutarea β elementelor de pe coloana de bordare și permutarea γ^{-1} a elementelor din interiorul tabelei. În rezultat, obținem următoarele transformări:

*	1	2	3
1	1	2	3
2	2	3	1
3	3	1	2

$\xrightarrow{\alpha}$

$\circ_\alpha$	1	2	3
1	3	1	2
2	1	2	3
3	2	3	1

$\xrightarrow{\beta}$

$\circ_\beta$	1	2	3
1	2	3	1
2	3	1	2
3	1	2	3

$\xrightarrow{\gamma^{-1}}$

$\circ_{\gamma^{-1}}$	1	2	3
1	3	1	2
2	1	2	3
3	2	3	1

3. Algoritm privind verificarea izomorfismelor de grupoizi

Fie că avem doi grupoizi (**Q**, *) și (**Q**, ∘) de același ordin **n**. Să se verifice dacă (**Q**, *) este izomorf cu (**Q**, ∘).

Algoritmul de verificare al grupoizilor la izomorfizm este următorul:

1. Se introduce dimensiunea n a grupoizilor;
2. Se introduce grupoidul (**Q**, *) din n elemente.
3. Se introduce grupoidul (**Q**, ∘) din n elemente.
4. Se generează toate substituțiile **α** (**n**!) și se testează condiția: **α** (**x**) ∘ **α** (**y**) = **α** (**x** * **y**).
5. Se afișează la monitor acele substituții **α**, pentru care obținem izomorfizm.
6. Se afișează la monitor rezultatul obținut: grupoizii dați (**Q**, *), (**Q**, ∘) sunt izomorfi pentru substituțiile α_1, α_2,... , α_n, sau grupoizii respectivi nu sunt izomorfi.

EXEMPLUL 3. Fie că avem doi grupoizi (**Q**, *) și (**Q**, ∘) de același ordin **n**=3, cu operațiile binare respective.

*	1	2	3
1	1	2	3
2	2	3	1
3	3	1	2

și

∘	1	2	3
1	3	1	2
2	1	2	3
3	2	3	1

Să se verifice dacă (**Q**, *) este izomorf cu (**Q**, ∘), aplicând algoritmul de mai sus.

Soluție. Conform algoritmului de mai sus efectuăm următorii pași:

P_1. Introducem **n**=3;

P_2. Se introduce operația grupoidului (**Q**, *);

P_3. Se introduce operația grupoidului (**Q**, ∘);

O funcție de atribuire a valorilor pentru operația grupoidului **gr** de dimensiune **n** este Atribuire(n, gr), descrisă în punctul 4. Soluție pentru algoritmul de verificare a izomorfismelor de grupoizi, în continuare (4-SVI).

P_4. Se generează toate substituțiile **α** (**n**! = 3! = 6);

Funcțiile valid(p), StToMatrix(n), backtr(p, n) descrise în (4-SVI), sunt destinate pentru stocarea tuturor substituțiilor curente **s** în mărimea **perm** – lista tuturor substituțiilor, unde n – numărul de elemente în mulțimea **Q**, deci în cazul dat **n** = 3, p – poziția curentă în substituția **s**, valid(p) întoarce valoarea 1 (true) dacă valoarea **s[p]** nu se regăsește printre valorile din stânga ale lui **s**, StToMatrix(n) – salvează substituția curentă **s** în mărimea **perm**, backtr(p, n) – funcție recursivă care schimbă valoarea lui **p** de la **1** la **n**.

Pentru n = 3; kf = 1; backtr(1, n); obținem:								
n	**p**	**i**	**(i≤n)-?**	**s**	**(valid(p)=1)-?**	**(p=n)-?**	StToMatrix(n) perm={…}	backtr(p+1, n)
3	1	1	(1≤3) - da	{1}	valid(1)=1-da	1=3-nu		R_0
								backtr(2, 3), R_1
	2	1	(1≤3) - da	{1, 1}	valid(2)=1-nu			R_1

		2	(2≤3) – da	{1, 2}	valid(2)=1-da	2=3-nu		R_1
backtr(3, 3), R_2								
	3	1	(1≤3) - da	{1, 2, 1}	valid(3)=1-nu			R_2
		2	(2≤3) – da	{1, 2, 2}	valid(3)=1-nu			R_2
		3	(3≤3) – da	{1, 2, 3}	valid(3)=1-da	3=3-da	{1, 2, 3}	R_2
	2	3	(3≤3) – da	{1, 3}	valid(2)=1-da	2=3-nu		R_1
backtr(3, 3), R_3								
	3	1	(1≤3) - da	{1, 3, 1}	valid(3)=1-nu			R_3
		2	(2≤3) - da	{1, 3, 2}	valid(3)=1-da	3=3-da	{1, 2, 3; 1, 3, 2}	R_3
		3	(3≤3) – da	{1, 3, 3}	valid(3)=1-nu			R_3
	1	2	(2≤3) – da	{2}	valid(1)=1-da	1=3-nu		R_0
backtr(2, 3), R_4								
	2	1	(1≤3) - da	{2, 1}	valid(2)=1-da	2=3-nu		R_4
backtr(3, 3), R_5								
	3	1	(1≤3) - da	{2, 1, 1}	valid(3)=1-nu			R_5
		2	(2≤3) - da	{2, 1, 2}	valid(3)=1-nu			R_5
		3	(3≤3) - da	{2, 1, 3}	valid(3)=1-da	3=3-da	{1, 2, 3; 1, 3, 2; 2, 1, 3}	R_5
	2	2	(2≤3) - da	{2, 2}	valid(2)=1-nu			R_4
		3	(3≤3) - da	{2, 3}	valid(2)=1-da	2=3-nu		R_4
backtr(3, 3), R_6								
	3	1	(1≤3) - da	{2, 3, 1}	valid(3)=1-da	3=3-da	{1, 2, 3;1, 3, 2; 2, 1, 3; 2, 3, 1}	R_6
		2	(2≤3) - da	{2, 3, 2}	valid(3)=1-nu			R_6
		3	(3≤3) - da	{2, 3, 3}	valid(3)=1-nu			R_6
	1	3	(3≤3) - da	{3}	valid(1)=1-da	1=3-nu		R_0
backtr(2, 3), R_7								
	2	1	(1≤3) - da	{3, 1}	valid(2)=1-da	2=3-nu		R_7
backtr(3, 3), R_8								
	3	1	(1≤3) - da	{3, 1, 1}	valid(3)=1-nu			R_8
		2	(2≤3) - da	{3, 1, 2}	valid(3)=1-da	3=3-da	{1, 2, 3;1, 3, 2, 2, 1, 3; 2, 3, 1; 3, 1, 2}	R_8
		3	(3≤3) - da	{3, 1, 3}	valid(3)=1-nu			R_8
	2	2	(2≤3) - da	{3, 2}	valid(2)=1-da	2=3-nu		R_7
backtr(3, 3), R_9								
	3	1	(1≤3) - da	{3, 2, 1}	valid(3)=1-da	3=3-da	{1, 2, 3;1, 3, 2; 2, 1, 3; 2, 3, 1; 3, 1, 2; 3, 2, 1}	R_9
		2	(2≤3) - da	{3, 2, 2}	valid(3)=1-nu			R_9
		3	(3≤3) - da	{3, 2, 3}	valid(3)=1-nu			R_9
	2	3	(3≤3) - da	{3, 3}	valid(2)=1-nu			R_7
		4	(4≤3) - nu					R_0

Am obținut matricea **perm** ce include toate substituțiile pentru **n**=3,

perm={(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)}.

Următorul pas este testarea relației $\alpha(\mathbf{x}) \circ \beta(\mathbf{y}) = \gamma(\mathbf{x} * \mathbf{y})$ pentru fiecare din substituțiile evidențiate.

1) Fie $\alpha_1 = \beta_1 = \gamma_1 = (1, 2, 3)$. Verificăm dacă obținem ori nu izomorfism pentru substituția dată.

x	**y**	**α(x)**	**β(y)**	**α(x) ∘ β(y)**	**γ(x * y)**	**α(x)∘β(y)=γ(x*y)**
1	1	α(1) = 1	β(1) = 1	α(1) ∘ β(1) = 1 ∘ 1 = 3	γ(1 * 1) = γ(1) = 1	3 ≠ 1

Deci, $\alpha(\mathbf{x}) \circ \alpha(\mathbf{y}) \neq \alpha(\mathbf{x} * \mathbf{y})$ și condiția pentru izomorfism nu se îndeplinește.

2) Fie $\alpha_2 = \beta_2 = \gamma_2 = (1, 3, 2)$. Verificăm dacă obținem ori nu izomorfism pentru substituția dată.

x	**y**	**α(x)**	**β(y)**	**α(x) ∘ β(y)**	**γ(x * y)**	**α(x)∘β(y)=γ(x*y)**
1	1	α(1) = 1	β(1) = 1	α(1) ∘ β(1) = 1 ∘ 1 = 3	γ(1 * 1) = γ(1) = 1	3 ≠ 1

Deci, $\alpha(\mathbf{x}) \circ \alpha(\mathbf{y}) \neq \alpha(\mathbf{x} * \mathbf{y})$ și condiția pentru izomorfism nu se îndeplinește.

3) Fie $\alpha_3 = \beta_3 = \gamma_3 = (2, 1, 3)$. Verificăm dacă obținem ori nu izomorfizm pentru substituția dată.

x	**y**	**α(x)**	**β(y)**	**α(x) ∘ β(y)**	**γ(x * y)**	**α(x)∘β(y)=γ(x*y)**
1	1	α(1) = 2	β(1) = 2	α(1) ∘ β(1) = 2 ∘ 2 = 2	γ(1 * 1) = γ(1) = 2	2 = 2
1	2	α(1) = 2	β(2) = 1	α(1) ∘ β(2) = 2 ∘ 1 = 1	γ(1 * 2) = γ(2) = 1	1 = 1
1	3	α(1) = 2	β(3) = 3	α(1) ∘ β(3) = 2 ∘ 3 = 3	γ(1 * 3) = γ(3) = 3	3 = 3
2	1	α(2) = 1	β(1) = 2	α(2) ∘ β(1) = 1 ∘ 2 = 1	γ(2 * 1) = γ(2) = 1	1 = 1
2	2	α(2) = 1	β(2) = 1	α(2) ∘ β(2) = 1 ∘ 1 = 3	γ(2 * 2) = γ(3) = 3	3 = 3
2	3	α(2) = 1	β(3) = 3	α(2) ∘ β(3) = 1 ∘ 3 = 2	γ(2 * 3) = γ(1) = 2	2 = 2
3	1	α(3) = 3	β(1) = 2	α(3) ∘ β(1) = 3 ∘ 2 = 3	γ(3 * 1) = γ(3) = 3	3 = 3
3	2	α(3) = 3	β(2) = 1	α(3) ∘ β(2) = 3 ∘ 1 = 2	γ(3 * 2) = γ(1) = 2	2 = 2
3	3	α(3) = 3	β(3) = 3	α(3) ∘ β(3) = 3 ∘ 3 = 1	γ(3 * 3) = γ(2) = 1	1 = 1

Observăm că avem izomorfism pentru substituția $\alpha_3 = \beta_3 = \gamma_3 = (2, 1, 3)$ și $\alpha(\mathbf{x}) \circ \alpha(\mathbf{y}) = \alpha(\mathbf{x} * \mathbf{y})$.

4) Fie $\alpha_4 = \beta_4 = \gamma_4 = (2, 3, 1)$. Verificăm dacă obținem ori nu izomorfizm pentru substituția dată.

x	**y**	**α(x)**	**β(y)**	**α(x) ∘ β(y)**	**γ(x * y)**	**α(x)∘β(y)=γ(x*y)**
1	1	α(1) = 3	β(1) = 3	α(1) ∘ β(1) = 3 ∘ 3 = 1	γ(1 * 1) = γ(1) = 3	1 ≠ 3

Deci, $\alpha(\mathbf{x}) \circ \alpha(\mathbf{y}) \neq \alpha(\mathbf{x} * \mathbf{y})$ și condiția pentru izomorfism nu se îndeplinește.

5) Fie $\alpha_5 = \beta_5 = \gamma_5 = (3, 1, 2)$. Verificăm dacă obținem ori nu izomorfizm pentru substituția dată.

x	y	α(x)	β(y)	α(x) ∘ β(y)	γ(x * y)	α(x)∘β(y)=γ(x*y)
1	1	α(1) = 2	β(1) = 2	α(1) ∘ β(1) = 2 ∘ 2 = 2	γ(1 * 1) = γ(1) = 2	2 = 2
1	2	α(1) = 2	β(2) = 3	α(1) ∘ β(2) = 2 ∘ 3 = 3	γ(1 * 2) = γ(2) = 3	3 = 3
1	3	α(1) = 2	β(3) = 1	α(1) ∘ β(3) = 2 ∘ 1 = 1	γ(1 * 3) = γ(3) = 1	1 = 1
2	1	α(2) = 3	β(1) = 2	α(2) ∘ β(1) = 3 ∘ 2 = 3	γ(2 * 1) = γ(2) = 3	3 = 3
2	2	α(2) = 3	β(2) = 3	α(2) ∘ β(2) = 3 ∘ 3 = 1	γ(2 * 2) = γ(3) = 1	1 = 1
2	3	α(2) = 3	β(3) = 1	α(2) ∘ β(3) = 3 ∘ 1 = 2	γ(2 * 3) = γ(1) = 2	2 = 2
3	1	α(3) = 1	β(1) = 2	α(3) ∘ β(1) = 1 ∘ 2 = 1	γ(3 * 1) = γ(3) = 1	1 = 1
3	2	α(3) = 1	β(2) = 3	α(3) ∘ β(2) = 1 ∘ 3 = 2	γ(3 * 2) = γ(1) = 2	2 = 2
3	3	α(3) = 1	β(3) = 1	α(3) ∘ β(3) = 1 ∘ 1 = 3	γ(3 * 3) = γ(2) = 3	3 = 3

Observăm că avem izomorfism pentru substituția $\alpha_3 = \beta_3 = \gamma_3 = (2, 1, 3)$ și α (**x**) ∘ α (**y**) = α (**x** * **y**).

6) Fie $\alpha_6 = \beta_6 = \gamma_6 = (3, 2, 1)$. Verificăm dacă obținem ori nu izomorfizm pentru substituția dată.

x	y	α(x)	β(y)	α(x) ∘ β(y)	γ(x * y)	α(x)∘β(y)=γ(x*y)
1	1	α(1) = 3	β(1) = 3	α(1) ∘ β(1) = 3 ∘ 3 = 1	γ(1 * 1) = γ(1) = 3	1 ≠ 3

Deci, α (**x**) ∘ α (**y**) ≠ α (**x** * **y**) și condiția pentru izomorfism nu se îndeplinește.

P_5. Deci pentru substituțiile $\alpha_3 = (2, 1, 3)$ și $\alpha_5 = (3, 1, 2)$, obținem că grupoidul (**Q**, *) este izomorf cu (**Q**, ∘) .

P_6. Grupoizii dați (**Q**, *) și (**Q**, ∘) sunt izomorfi pentru substituțiile α_3, α_5.

Funcția VerificareIzomorf(q1, a, b, g, q2, n) descrisă în (4-SVI), reîntoarce valoarea 1 dacă grupoidul q1 și q2 sunt izomorfi în raport cu substrituțiile a, b, g, unde n – număr de elemente în mulțimea grupoidului.

PROBLEMĂ. Folosind algoritmul de mai sus, pentru fiecare din exemplele de mai jos, să se verifice dacă există izomorfism între grupoizii (**Q**,•) și (**Q**, *).

Ex. 1

(**Q**, •)

	1	2	3	4
1	1	2	3	4
2	2	3	4	1
3	4	1	2	3
4	3	4	1	2

(**Q**, *)

•	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	4	3	2	1
4	3	1	4	2

Ex. 2

(**Q**, •)

•	1	2	3	4
1	1	2	3	4
2	2	1	4	3
3	3	4	1	2
4	4	3	2	1

(**Q**, *)

•	1	2	3	4
1	2	1	4	3
2	1	2	3	4
3	4	3	2	1
4	3	4	1	2

Ex. 3

(**Q**, •)

•	1	2	3
1	1	2	3
2	3	1	2
3	2	3	1

(**Q**, *)

•	1	2	3
1	3	2	1
2	1	3	2
3	2	1	3

Ex. 4

(**Q**, •)

•	1	2	3
1	1	2	3
2	2	3	1
3	3	1	2

(**Q**, *)

•	1	2	3
1	3	1	2
2	2	3	1
3	1	2	3

4. Soluție pentru algoritmul de verificarea izomorfismelor de grupoizi

```
//C-Free 5.0
#include<iostream.h>
#include<iomanip.h>
const int nn = 100;
const int ng = 10;
int kf = -1, perm[nn][10], s[nn];
int valid(int p) {//1-true 0-false
   int i, ok = 1;
   for (i = 1; i <= p - 1; i++)
      if (s[p] == s[i]) ok = 0;
   return ok;}
void StToMatrix(int n) {    int i;
   for (i = 1; i <= n; i++) {
      perm[kf][i] = s[i];    }
   kf++;    }
void backtr(int p, int n) {    int i;
   for (i = 1; i <= n; i++) {  s[p] = i;
      if (valid(p) == 1) {
    if (p == n) { StToMatrix(n); }
    else { backtr(p + 1, n); }   }  } }
void Atribuire(int n1, int gr[nn][nn]) {
   int i, j;
   for (i = 1; i <= n1; i++)
      for (j = 1; j <= n1; j++) {
         cout << "gr[" << i << ", " << j <<
"]="; cin >> gr[i][j];       }  }
void Afisare(char NumeGr[ng], char
Operatie, int n1, int gr[nn][nn]) {
   int i, j;
   cout << "Grupoidul: (" << NumeGr <<
"," << Operatie << ")={";
  for (i=1; i<n1; i++) cout <<i<< ", ";
   cout << n1 << "};" << endl;
   cout << "Cu operatia binara:" << endl;
   cout << setw(2) << Operatie << "|";
   for (i = 1; i <= n1; i++) cout << setw(3)
<< i; cout << endl;
   cout << "--+";
   for (i = 1; i <= n1; i++) cout << "---";
cout << endl;
   for (i = 1; i <= n1; i++) {
      cout << setw(2) << i << "|";
      for (j = 1; j <= n1; j++) {
         cout << setw(3) << gr[i][j]; }
      cout << endl;    }  }
int VerificareIzomorf(int q1[nn][nn], int
a[nn], int b[nn], int g[nn], int q2[nn][nn],
int n) {
   int i, j, k, iz = 1, ax, by, gq2;
   for (i = 1; i <= n; i++) {
      for (j = 1; j <= n; j++) {
         ax = -1, by = -1, gq2 = -1;
         for (k = 1; k <= n; k++) {
            if (a[k] == i) ax = k;
            if (b[k] == j) by = k;
      if (g[k] == q1[i][j]) gq2 = k;   }
 if ((ax > 0) && (by > 0) && (gq2 >> 0))
{
            if (q2[ax][by] != gq2) { iz = 0; j
= n; i = n; };      }    }  }
   return iz;  }
int main() {  int q1[nn][nn], q2[nn][nn];
   int iz,  i, j, k, p, n;
   cout << "Dimensiune (Q,.) n=";cin>>n;
   cout << "Introdu grupoidul (Q,.):\n";
Atribuire(n, q1);
   cout << "Introdu grupoidul (Q,*):\n";
Atribuire(n, q2); Afisare("Q",'.',n,q1);
Afisare("Q",'*',n,q2); kf=1; backtr(1,n);
  iz = 0;  cout << "perm[k][i] = \n";
   for (k = 1; k < kf; k++) {
      for (i = 1; i <= n; i++)
         cout << perm[k][i] << ", ";
      cout << endl;    }
```

```
   for (k = 1; k < kf; k++) {
      p = VerificareIzomorf(q1, perm[k],
perm[k], perm[k], q2, n);
      if (p == 1) {       iz = 1;
         cout << "Avem izomorfizm pentru
substitutia ALFA = ";
         for (i = 1; i <= n; i++) cout <<
setw(4) << perm[k][i];
         cout << endl;      }    }
   if (iz == 0) {       cout << "Grupoizii nu
sunt izomorfi !\n";    }
   return 0;
}
```

5. Rezultate

Rezultate obținute privind verificarea izomorfismelor de grupoizi:

	Ex 1		Ex 2		Ex 3		Ex 4	
	(Q, •)	(Q, *)	(Q, •)	(Q, *)	(Q, •)	(Q, *)	(Q, •)	(Q, *)
Avem izomorfizm pentru substituția	α = (1 2 4 3)		α = (2 1 3 4) α = (2 1 4 3) α = (2 3 4 1) α = (2 4 1 3)		Nu există izomorfism		Nu există izomorfism	

Concluzii. Înțelegerea noțiunilor algebrice pentru studenții informaticieni și aplicarea lor la soluționarea unor probleme practice, nu întotdeauna este un proces simplu și clar. În articolul respectiv sunt prezentate unele abordări metodice care se referă la algoritmul verificării izomorfismelor de grupoizi, algoritm necesar pentru studierea unor compartimente din criptografie. Astfel interpretate și redate conceptele fundamentale din algebra abstratcă devin mai clare pentru studenții informaticieni și, ulterior, le pot aplica cu succese în practică. Metodologia propusă, facilitează procesul de pregătire a informaticienilor în domeniul algebrei abstracte, algoritmică și programare.

Articol realizat în cadrul proiectului de cercetări științifice „Metodologia implementării TIC în procesul de studiere a științelor reale în sistemul de educație din Republica Moldova din perspectiva inter/transdisciplinarității (concept STEAM)”, inclus în „Program de stat” (2020-2023), Prioritatea IV: Provocări societale, cifrul 20.80009.0807.20, cu suportul financiar oferit de Agenția Națională pentru Dezvoltare și Cercetare

Bibliografie

1. CHIRIAC, L. *Structuri algebrice pe computer*. Chișinău, 2014. ISBN 978-9975-53-301-0, 60 p.
2. GALLIAN, J.A. *Contemporary Abstract Algebra*. Cengage Learning, 2017. 631 p. ISBN: 978-1-305-65796-0.
3. https://en.wikipedia.org/wiki/Isomorphism. *Isomorphism*. [accesat la data de 10.02.2020].