CONTRACŢII ALE GRUPOIZILOR ASOCIAŢI SISTEMELOR DINAMICE DISCRETE

Mădălina Roxana Buneci, *conf. dr, Universitatea Constantin Brâncuşi din Târgu-Jiu, ROMÂNIA*

GROUPOID REDUCTIONS ASSOCIATED TO DISCRETE DYNAMICAL SYSTEMS

Mădălina Roxana Buneci, *Assoc. Professor, University Constantin Brâncuşi of Târgu-Jiu, ROMANIA*

REZUMAT. Fie G(X,φ) grupoidul asociat sistemului dinamic discret (X,φ) ca în [4]. Stabilim condiţii necesare şi suficiente pentru ca un subgrupoid G al grupoidului trivial X×**Z**×X să provină dintr-un sistem dinamic discret (X,φ). În particular, utilizăm rezultatul pentru a stabili dacă reducerea (contracţia) la o submulţime de stări A⊂X a grupoidului G(X,φ) este de forma G(A,φ_A) precum şi pentru a determina φ_A.

În cazul unui spaţiu X cu un număr finit de stări, definim de proceduri pentru a testa dacă un subgrupoid G al grupoidului X×**Z**×X provine dintr-un sistem dinamic discret şi dacă răspunsul este pozitiv pentru a determina toate funcţiile φ cu proprietatea că G(X,φ)⊂G, precum şi toate funcţiile φ cu proprietatea că G(X,φ) = G şi care sunt constante pe clasele de echivalenţă ale relaţiei x~y <=> (x,0,y)∈G.

CUVINTE CHEIE: grupoid; sistem dinamic discret; contracţie a unui grupoid; orbită; CAS.

ABSTRACT. Let G(X,φ) be the groupoid associated to a discrete dynamical system (X,φ) as in [4]. We establish necessary and sufficient conditions for a subgroupoid G of the trivial groupoid X×**Z**×X to be associated with a discrete dynamical system (X,φ). Particularly, we use the result to check if the reduction to A⊂X of groupoid G(X,φ) is of the form G(A,φ_A) and to determine φ_A.

For the case of a finite state space X we provide a package of subroutines (Maple procedures) for testing if the subgroupoid G of X×**Z**×X arises from a discrete dynamical system and if the answer is positive to find all the time evolutions φ with the property that G(X,φ)⊂G, as well as all time evolutions φ with the property that G(X,φ)=G and that are constant on the equivalence classes of the relation x~y iff (x,0,y)∈G.

KEY WORDS: groupoid; discrete dynamical system; groupoid reduction; orbit; CAS.

1. INTRODUCERE

Cadrul matematic pentru studiul unui sistem dinamic discret este dat de un spaţiu X (spaţiul tuturor stărilor posibile ale sistemului) şi o aplicaţie φ care descrie evoluţia timpului: starea x∈X la timpul t = 0 evoluează în φ(x) la t = 1, φ(φ(x)) la t = 2, etc. După modelul din [4] putem asocia un grupoid G(X, φ) sistemului dinamic discret (X, φ). Considerăm o submulţime de stări A⊂X şi studiem dacă reducerea (contracţia) grupoidului G(X,φ) la A este de forma G(A,φ_A) şi în caz afirmativ determinăm funcţiile φ_A care au această proprietate.

1. INTRODUCTION

The mathematical setting for a discrete-time dynamical system is a space X (the space of all possible states of the system) and a map φ:X→X that describes time evolution: the state x∈X at time t = 0 evolves into φ(x) at t = 1, φ(φ(x)) at t = 2, etc. As in [4] we can associate a groupoid G(X,φ) to the discrete dynamical system (X,φ). We consider A ⊂ X and study if the reduction of G(X,φ) to A is of the form G(A,φ_A). If the answer is positive, we determine the functions φ_A having this property. We use the same terminology and notation as in [4] and [1].

Utilizăm aceeaşi terminologie şi notaţii ca în [4] şi [1].

2. SUBGRUPOIZI AI GRUPOIDULUI TRIVIAL

Fie X o mulţime şi **Z** grupul numerelor întregi. În raport cu operaţiile

$$(x, n, y)(y, m, z) = (x, n+m, y)$$
$$(x, n, y)^{-1} = (y, -n, x)$$

$X\times\mathbf{Z}\times X$ este un grupoid al cărui spaţiu al unităţilor $\{(x,0,x): x\in X\}$ poate fi identificat cu X.

Fie R o relaţie de echivalenţă pe X, şi $\{k_{x,y}\}_{(x,y)\in R}$ o familie de numere întregi ce satisfac următoarele condiţii:

1. $k_{x,x}\geq 0$ pentru orice $x\in X$.
2. Dacă $k_{x,x} \neq 0$, atunci $k_{x,y} + k_{y,z} = k_{x,z}$ (mod $k_{x,x}$), altfel $k_{x,y} + k_{y,z} = k_{x,z}$.

Atunci $k_{x,x} = k_{y,y}$ pentru orice $(x,y)\in R$. Într-adevăr, dacă $k_{x,x} = 0$, atunci $k_{x,y} + k_{y,y} = k_{x,y}$. Deci $k_{y,y} = 0$. Reciproc, dacă $k_{y,y} = 0$, deoarece $k_{y,x} + k_{x,x} = k_{y,x}$, rezultă că $k_{x,x} = 0$. Dacă $k_{x,x} \neq 0$, atunci $k_{x,y} + k_{y,y} = k_{x,y}$ (mod $k_{x,x}$) şi în consecinţă, $k_{y,y} = 0$ (mod $k_{x,x}$). Analog, $k_{x,x} = 0$ (mod $k_{y,y}$). Astfel $k_{x,x} = k_{y,y}$. Dacă $k_{x,x}=0$, atunci $k_{x,y} = - k_{y,x}$ (pentru că $k_{x,y} + k_{y,x} = k_{x,x} = 0$). Dacă $k_{x,x}\neq 0$, atunci $k_{x,y} = k_{x,x}- k_{y,x}$ (mod $k_{x,x}$) (pentru că $k_{x,y} + k_{y,x} = k_{x,x}$ (mod $k_{x,x}$)).

Dacă pentru fiecare $x\in X$ cu proprietatea că $k_{x,x} \neq 0$ şi fiecare y aparţinând clasei de echivalenţă a lui x (i.e. $(x,y)\in R$) înlocuim $k_{x,y}$ prin $k_{x,y}$ mod $k_{x,x}$, atunci noua familie satisface de asemenea condiţiile 1-2. În cele ce urmează considerăm că dacă $k_{x,x} \neq 0$ şi $x\neq y$, atunci

$$k_{x,y}\in \{0,1,\ldots,k_{x,x}-1\}$$

pentru orice y pentru orice $(x,y)\in R$. Deoarece $k_{x,y} = k_{x,x}- k_{y,x}$ (mod $k_{x,x}$), rezultă că dacă $k_{y,x} = 0$, atunci $k_{x,y}=0$. Dacă $k_{y,x} \neq 0$ şi $x\neq y$, atunci $k_{x,y}= k_{x,x}- k_{y,x}$.

Nu este dificil de demonstrat că

$$G=\{(x,k_{x,y}+tk_{x,x},y): (x,y)\in R, t\in Z\}$$

este un subgrupoid al lui $X\times\mathbf{Z}\times X$ cu $G^{(0)} = X$.

Reciproc, orice subgrupoid G al lui $X\times\mathbf{Z}\times X$ cu $G^{(0)} = X$ poate fi obţinut în acest fel. Demonstraţia este similară cu cea a

2. SUBGROUPOIDS OF THE TRIVIAL GROUPOID

Let X be a set and **Z** be the group of integer numbers. Under the operations

$$(x, n, y)(y, m, z) = (x, n+m, y)$$
$$(x, n, y)^{-1} = (y, -n, x)$$

$X\times\mathbf{Z}\times X$ is a groupoid whose unit space

$$\{(x,0,x): x\in X\}$$

can be identified with X.

Let R be the graph of an equivalence relation on X, and $\{k_{x,y}\}_{(x,y)\in R}$ be a family of integer numbers satisfying the following conditions

1. $k_{x,x}\geq 0$ for all $x\in X$.
2. If $k_{x,x} \neq 0$, then $k_{x,y} + k_{y,z} = k_{x,z}$ (mod $k_{x,x}$), else $k_{x,y} + k_{y,z} = k_{x,z}$.

Then $k_{x,x} = k_{y,y}$ for all $(x,y)\in R$. Indeed, if $k_{x,x} = 0$, then $k_{x,y} + k_{y,y} = k_{x,y}$. Hence $k_{y,y} = 0$. Conversely, if $k_{y,y} = 0$, since $k_{y,x} + k_{x,x} = k_{y,x}$, it follows that $k_{x,x} = 0$. If $k_{x,x} \neq 0$, then $k_{x,y} + k_{y,y} = k_{x,y}$ (mod $k_{x,x}$) and consequently, $k_{y,y} = 0$ (mod $k_{x,x}$). Similarly, $k_{x,x} = 0$ (mod $k_{y,y}$). Thus $k_{x,x} = k_{y,y}$. If $k_{x,x}=0$, then $k_{x,y} = - k_{y,x}$ (because $k_{x,y} + k_{y,x} = k_{x,x} = 0$). If $k_{x,x}\neq 0$, then $k_{x,y} = k_{x,x}- k_{y,x}$ (mod $k_{x,x}$) (because $k_{x,y} + k_{y,x} = k_{x,x}$ (mod $k_{x,x}$)).

If for each $x\in X$ with the property that $k_{x,x} \neq 0$ and each y belonging to the equivalence class of x (i.e. $(x,y)\in R$) we replace $k_{x,y}$ by $k_{x,y}$ mod $k_{x,x}$, then the new family still satisfies the conditions 1-2. Subsequently, we consider that if $k_{x,x} \neq 0$ and $x\neq y$, then

$$k_{x,y}\in \{0,1,\ldots,k_{x,x}-1\}$$

for all y such that $(x,y)\in R$. Since $k_{x,y} = k_{x,x}- k_{y,x}$ (mod $k_{x,x}$), it follows that if $k_{y,x} = 0$, then $k_{x,y}=0$. If $k_{y,x} \neq 0$ and $x\neq y$, then $k_{x,y}= k_{x,x}- k_{y,x}$.

It is not difficult to prove that

$$G =\{(x,k_{x,y}+tk_{x,x},y): (x,y)\in R, t\in Z\}$$

is a subgroupoid of $X\times\mathbf{Z}\times X$ with $G^{(0)} = X$.

Conversely, any subgroupoid G of $X\times\mathbf{Z}\times X$ with $G^{(0)} = X$ can be obtain in this way. The proof is similar with the proof Proposition 2.5 [3]. Indeed, let R be the principal groupoid associated to G. For each propoziţiei 2.5 [3]. Într-adevăr, fie R grupoidul principal asociat lui G. Pentru fiecare $x\in X$, grupul de izotropie în x asociat grupoidului G este

$$G_x^x = \{(x, k, x) \in X \times \mathbf{Z} \times X : (x,k,x)\in G\}.$$

Astfel $H_x^x = \{k: (x, k, x) \in G_x^x\}$ este un subgroup of $\mathbf{Z}$, şi de aceea există un întreg $k_{x,x}\geq 0$ astfel încât $H_x^x = k_{x,x}\mathbf{Z}$ ($k_{x,x} = 0 \iff H_x^x = \{0\}$, şi în caz contrar $k_{x,x}$ este cel mai mic număr întreg pozitiv $k\in H_x^x$). Aşadar $G_x^x = \{(x, k_{x,x}t, x): t\in \mathbf{Z}\}$. Deoarece pentru orice grupoid G

$$\gamma \to \gamma_0\gamma \; [: G_{d(\gamma_0)}^{d(\gamma_0)} \to G_{d(\gamma_0)}^{r(\gamma_0)}]$$

$$\gamma \to \gamma\gamma_0 \; [: G_{r(\gamma_0)}^{r(\gamma_0)} \to G_{d(\gamma_0)}^{r(\gamma_0)}]$$

sunt bijective, rezultă că $k_{x,x} = k_{y,y}$ pentru orice $(x,y)\in R$, iar dacă $k_{x,x}= 0$, atunci G_y^x (respectiv, G_x^y) conţine doar un element $(x, k_{x,y}, y)$ (respectiv, $(y,k_{y,x},x)$). Deoarece $G_y^x = \left(G_y^x\right)^{-1}$, $k_{y,x} = - k_{x,y}$. Dacă $k_{x,x} \neq 0$, atunci dacă $k_{x,y}$ este cel mai mic întreg nenegativ k cu proprietatea că $(x,k,y)\in G_y^x$, şi dacă (x,n,y) este un alt element din G_y^x, atunci $(x,k_{x,y},y)\,(x,n,y)^{-1}\in G_x^x$. Prin urmare există $t\in \mathbf{Z}$ astfel încât $k_{x,y} - n = tk_{x,x}$.

3. GRUPOIZI ASOCIAŢI SISTEMELOR DINAMICE DISCRETE

Fie X o mulţime şi $\varphi: X \to X$ o funcţie. Notăm prin $G(X, \varphi)$ mulţimea:

$$G(X, \varphi) = \{(x,k,y)\in X\times Z\times X: \text{există } n\in \mathbf{N} \text{ astfel încât } n+k\geq 0 \text{ şi } \varphi^{n+k}(x) = \varphi^n(y)\},$$

unde $\mathbf{Z}$ este grupul numerelor întregi şi $\mathbf{N}$ mulţimea numerelor naturale. Atunci $G(X, \varphi)$ este un subgrupoid al lui $X\times\mathbf{Z}\times X$ cu acelaşi spaţiu al unităţilor [4]. Notăm cu R_φ relaţia de echivalenţă: $(x,y)\in R_\varphi \iff$ există $n\in \mathbf{N}$ astfel încât $\varphi^n(x) = \varphi^n(y)$ (sau echivalent, $(x,0,y)\in G(X,\varphi)$)

Demonstrăm că dacă $\varphi_1, \varphi_2: X \to X$ sunt

$x\in X$, the isotropy group at x associated to the groupoid G is

$$G_x^x = \{(x, k, x) \in X \times \mathbf{Z} \times X : (x,k,x)\in G\}.$$

Thus

$$H_x^x = \{k: (x, k, x) \in G_x^x\}$$

is a subgroup of $\mathbf{Z}$, and therefore there is an integer $k_{x,x}\geq 0$ such that $H_x^x = k_{x,x}\mathbf{Z}$ ($k_{x,x} = 0$ iff $H_x^x = \{0\}$ and $k_{x,x}$ is the smallest positive integer $k\in H_x^x$ otherwise). Hence

$$G_x^x = \{(x, k_{x,x}t, x): t\in \mathbf{Z}\}.$$

Since for any groupoid G

$$\gamma \to \gamma_0\gamma \; [: G_{d(\gamma_0)}^{d(\gamma_0)} \to G_{d(\gamma_0)}^{r(\gamma_0)}]$$

$$\gamma \to \gamma\gamma_0 \; [: G_{r(\gamma_0)}^{r(\gamma_0)} \to G_{d(\gamma_0)}^{r(\gamma_0)}]$$

are bijections, it follows that $k_{x,x} = k_{y,y}$ for all $(x,y)\in R$ and if $k_{x,x}= 0$, then G_y^x (respectively, G_x^y) contains only one element $(x, k_{x,y}, y)$ (respectively, $(y,k_{y,x},x)$). Since $G_y^x = \left(G_y^x\right)^{-1}$, $k_{y,x} = - k_{x,y}$. If $k_{x,x} \neq 0$, then if $k_{x,y}$ is the smallest nonnegative integer k with the property that $(x,k,y)\in G_y^x$, and if (x,n,y) is another element of G_y^x, then

$$(x,k_{x,y},y)\,(x,n,y)^{-1}\in G_x^x.$$

Thus there is $t\in Z$ such that $k_{x,y} - n = tk_{x,x}$.

3. GROUPOIDS ASSOCIATED TO DISCRETE DYNAMICAL SYSTEMS

Let X be a set and $\varphi: X \to X$ be a function. Let us denote be $G(X, \varphi)$ the set:

$$G(X, \varphi) = \{(x,k,y)\in X\times Z\times X: \text{there is } n\in N \text{ such that } n+k\geq 0 \text{ and } \varphi^{n+k}(x) = \varphi^n(y)\},$$

where $\mathbf{Z}$ is the group of integers and $\mathbf{N}$ is the set of nonnegative integers. Then $G(X, \varphi)$ is a subgroupoid of $X\times\mathbf{Z}\times X$ having the same unit space [4]. Let us denote by R_φ the graph of the following equivalence relation: $(x,y)\in R_\varphi$ iff there is $n\in N$ such that $\varphi^n(x) = \varphi^n(y)$ (or equivalently, iff $(x,0, y)\in G(X,\varphi)$)

Let us prove that if $\varphi_1, \varphi_2: X \to X$ are două funcţii pe X, atunci G(X, φ_2) $\subset$ G(X, φ_1) dacă şi numai dacă pentru fiecare $x\in X$, există $n(x)\in \mathbf{N}$ astfel încât $\varphi_1^{n(x)+1}(x) = \varphi_1^{n(x)}(\varphi_2(x))$. Deoarece $(x,1,\varphi_2(x))\in$ G(X, φ_2) $\subset$ G(X, φ_1), rezultă că există $n(x)\in \mathbf{N}$ astfel încât $\varphi_1^{n(x)+1}(x) = \varphi_1^{n(x)}(\varphi_2(x))$. Reciproc dacă există $n(x)\in N$ astfel încât $\varphi_1^{n(x)+1}(x) = \varphi_1^{n(x)}(\varphi_2(x))$, atunci pentru orice $k\in \mathbf{N}$, $\varphi_1^{n(k,x)+k}(x) = \varphi_1^{n(k,x)}(\varphi_2^k(x))$, unde

$$n(k,x)= n(x)+ n(\varphi_2(x))+ ...+ n(\varphi_2^{k-1}(x)).$$

Dacă $(x,k,y)\in$ G(X, φ_2), atunci există $m\in \mathbf{N}$ astfel încât $m+k\geq 0$ şi $\varphi_2^{m+k}(x) = \varphi_2^m(y)$. Atunci $\varphi_1^{n(m+k,x)+m+k}(x) = \varphi_1^{n(m+k,x)}(\varphi_2^{m+k}(x))$ şi $\varphi_1^{n(m,y)+m}(y) = \varphi_1^{n(m,y)}(\varphi_2^m(y))$. În consecinţă, dacă

$$p=\max\{n(m+k,x), n(m,y)\},$$

atunci $\varphi_1^{p+m+k}(x)=\varphi_1^{p+m}(y)$. Aşadar (x,k,y) este în G(X, φ_1).

Dacă $R_{\varphi_1}\subset R_{\varphi_2}$ şi pentru fiecare $x\in X$, există $n(x)\in N$ astfel încât $\varphi_1^{n(x)+1}(x) = \varphi_1^{n(x)}(\varphi_2(x))$, atunci G(X, φ_2) = G(X, φ_1). Într-adevăr, dacă $(x,k,y)\in$ G(X, φ_1), atunci există $m\in \mathbf{N}$ astfel încât $m+k\geq 0$ şi $\varphi_1^{m+k}(x) = \varphi_1^m(y)$. Atunci $\varphi_1^{m+k+n(x)+n(y)}(x) = \varphi_1^{m+n(x)+n(y)}(y)$. Prin urmare

$$\varphi_1^{n(x)+n(y)}(\varphi_2^{m+k}(x))= \varphi_1^{n(x)+n(y)}(\varphi_2^m(y)),$$

şi de aceea $(\varphi_2^{m+k}(x),\varphi_2^m(y))\in R_{\varphi_1}\subset R_{\varphi_2}$. Astfel există $p\in \mathbf{N}$ astfel încât $\varphi_2^{p+m+k}(x)=\varphi_2^{p+m}(y)$. Deci (x,k,y) este în G(X, φ_2).

Fie G un subgrupoid al lui $X\times\mathbf{Z}\times X$ cu $G^{(0)}$=X şi fie $R_G = \{(x,y): (x,0,y)\in G\}$. Atunci există a funcţie $\varphi:X\to X$ astfel încât G = G(X,φ) dacă şi numai dacă pentru fiecare $x\in X$ există $y_x\in X$ astfel încât $(x, 1, y_x) \in G$. Într-adevăr, dacă $x'\in X$ satisface $(x,x')\in R_G$, atunci $(x',1,y_x)\in G$. Dacă definim $\varphi(x')=y_x$ pentru orice x' în clasa de echivalenţă a lui x

two functions on X, then G(X, φ_2) $\subset$ G(X, φ_1) if and only if for each $x\in X$, există $n(x)\in \mathbf{N}$ such that $\varphi_1^{n(x)+1}(x) = \varphi_1^{n(x)}(\varphi_2(x))$. Since $(x,1,\varphi_2(x))\in$ G(X, φ_2) $\subset$ G(X, φ_1), it follows that there is $n(x)\in \mathbf{N}$ such that $\varphi_1^{n(x)+1}(x) = \varphi_1^{n(x)}(\varphi_2(x))$. Conversely, if there is $n(x)\in N$ such that $\varphi_1^{n(x)+1}(x) = \varphi_1^{n(x)}(\varphi_2(x))$, then for all $k\in \mathbf{N}$,

$$\varphi_1^{n(k,x)+k}(x) = \varphi_1^{n(k,x)}(\varphi_2^k(x))$$

where $n(k,x)= n(x)+ n(\varphi_2(x))+ ...+ n(\varphi_2^{k-1}(x))$

If $(x,k,y)\in$ G(X, φ_2), then there is $m\in \mathbf{N}$ such that $m+k\geq 0$ and $\varphi_2^{m+k}(x) = \varphi_2^m(y)$. Then $\varphi_1^{n(m+k,x)+m+k}(x) = \varphi_1^{n(m+k,x)}(\varphi_2^{m+k}(x))$ and $\varphi_1^{n(m,y)+m}(y) = \varphi_1^{n(m,y)}(\varphi_2^m(y))$. Consequently, if

$$p=\max\{n(m+k,x), n(m,y)\},$$

then $\varphi_1^{p+m+k}(x)=\varphi_1^{p+m}(y)$. Hence (x,k,y) is in G(X, φ_1).

If $R_{\varphi_1}\subset R_{\varphi_2}$ and for each $x\in X$, there is $n(x)\in N$ such that $\varphi_1^{n(x)+1}(x) = \varphi_1^{n(x)}(\varphi_2(x))$, then G(X, φ_2) = G(X, φ_1). Indeed, if $(x,k,y)\in$ G(X, φ_1), then there is $m\in \mathbf{N}$ such that $m+k\geq 0$ and $\varphi_1^{m+k}(x) = \varphi_1^m(y)$. Then $\varphi_1^{m+k+n(x)+n(y)}(x) = \varphi_1^{m+n(x)+n(y)}(y)$. Hence

$$\varphi_1^{n(x)+n(y)}(\varphi_2^{m+k}(x))= \varphi_1^{n(x)+n(y)}(\varphi_2^m(y)),$$

and therefore $(\varphi_2^{m+k}(x),\varphi_2^m(y))\in R_{\varphi_1}\subset R_{\varphi_2}$. Thus there is $p\in \mathbf{N}$ such that $\varphi_2^{p+m+k}(x)=\varphi_2^{p+m}(y)$. Hence (x,k,y) is in G(X, φ_2).

Let G be a subgroupoid of $X\times\mathbf{Z}\times X$ with $G^{(0)}$=X and let $R_G = \{(x,y): (x,0,y)\in G\}$.Then there is a function $\varphi:X\to X$ such that G = G(X,φ) iff for each $x\in X$ there is $y_x\in X$ such that $(x, 1, y_x) \in G$. Indeed, if $x'\in X$ satisfy $(x,x')\in R_G$, then $(x',1,y_x)\in G$. If we define $\varphi(x')=y_x$ for all x' in the class of x with respect to R_G, then G(X,φ)=G. Conversely, if G=G(X,φ), then $(x,1,\varphi(x))\in G$.

relativ la R_G, atunci $G(X,\varphi)=G$. Reciproc, dacă $G=G(X,\varphi)$, atunci $(x,1,\varphi(x))\in G$.

4. SUBGRUPOIZI AI GRUPOIZILOR TRIVIALI PE MULŢIMI FINITE

Să presupunem că $X = \{x_1, x_2, \ldots, x_n\}$. În secţiunea 2 am arătat că orice subgrupoid G al lui $X \times \mathbf{Z} \times X$ cu $G^{(0)} = X$ este dat de o familie de numere întregi $\{k_{x,y}\}_{(x,y)\in R}$ ce satisface anumite proprietăţi. Să observăm că în cazul X finit, este suficient să cunoaştem lista orbitelor $o_1,\ldots, o_p$ lui G (sau echivalent grupoidul principal R asociat lui G), numerele k_{x_q,x_q}, q=1..p, $x_q\in o_q$ şi $k_{x_i,x_{j_i}}$, i=1..n, unde j_i este cel mai mare număr întreg ce satisface j<i şi $(x_i,x_j)\in R$ (mai precis, x_j precede x_i în orbita lui $[x_i]$ bine ordonată relativ la relaţia $x_j < x_m \Leftrightarrow j < m$). Pentru fiecare $x_i\in X$, notăm $p(x_i) = x_{j_i}$ (punctul care precede x_i în orbita lui $[x_i]$). Arătăm că orbitele şi setul de numere de deasupra (n valori) sunt suficiente pentru a determina G. Într-adevăr, dacă $x \in [x_q]$, atunci $k_{x,x}=k_{x_q,x_q}$. Dacă $(x,y)\in R$ şi x > y, atunci există m astfel încât $y=p^m(x)$ şi astfel $k_{x,y} = (k_{x,p(x)} + k_{p(x),p(p(x))} +\ldots+k_{p^{m-1}(x),p^m(x)})$ mod $k_{x,x}$ dacă $k_{x,x}\neq 0$ şi $k_{x,y} = k_{x,p(x)} + k_{p(x),p(p(x))} + \ldots +k_{p^{m-1}(x),p^m(x)}$, în caz contrar. Dacă $(x,y)\in R$ şi x < y, atunci există m astfel încât $x = p^m(y)$ şi astfel $k_{x,y} = (k_{p^m(y),p^{m-1}(y)}+\ldots+ k_{p(p(y)),p(y)} + k_{p(y),y})$ mod $k_{x,x} = ((k_{x,x} - k_{p^{m-1}(x),p^m(x)}) + \ldots +(k_{x,x} - k_{p(y),p(p(y))}) +(k_{x,x} - k_{y,p(y)}))$ mod $k_{x,x}$, dacă $k_{x,x} \neq 0$, şi $k_{x,y} = - k_{p(y),p(p(y))} - \ldots - k_{p(y),p(p(y))} - k_{y,p(y)}$, în caz contrar.

Dacă există a funcţie $\varphi: X \to X$ astfel încât $G = G(X,\varphi)$, atunci $k_{x,x} \neq 0$ pentru orice x (Deoarece X este finită, rezultă $\varphi^m(x)$ poate lua doar un număr finit de valori distincte când m variază. Astfel putem alege $k_{x,x}>0$ să fie cel mai mic număr întreg pozitiv k astfel încât există $n\in\mathbf{N}$ cu proprietatea că$\varphi^{k+n}(x) =$

4. SUBGROUPOIDS OF TRIVIAL GROUPOIDS ON FINITE SETS

Let us assume that $X = \{x_1, x_2, \ldots, x_n\}$. In Section 2 we have noticed that every subgroupoid G of $X \times \mathbf{Z} \times X$ with $G^{(0)} = X$ is given by a family of integer numbers $\{k_{x,y}\}_{(x,y)\in R}$ satisfying certain properties. Let us notice that in the finite case it is enough to know the list of orbits $o_1,\ldots, o_p$ of G (or equivalently the principal groupoid R associated to G), the numbers k_{x_q,x_q}, q=1..p, $x_q\in o_q$ and $k_{x_i,x_{j_i}}$, i=1..n, where j_i is the larger integer satisfying j<i and $(x_i,x_j)\in R$ (more precisely, x_j precedes x_i in the orbit of $[x_i]$ endowed with the well-order relation $x_j < x_m$ iff j < m). For each $x_i\in X$, let us denote $p(x_i) = x_{j_i}$ (the point that precedes x_i in the orbit of $[x_i]$). Let us show that the orbits and the above set of numbers (n numbers) are enough to determine G. Indeed, if $x \in [x_q]$, then $k_{x,x}=k_{x_q,x_q}$. If $(x,y)\in R$ and x > y, then există m such that $y=p^m(x)$ and thus $k_{x,y} = (k_{x,p(x)} + k_{p(x),p(p(x))} +\ldots+k_{p^{m-1}(x),p^m(x)})$ mod $k_{x,x}$ if $k_{x,x}\neq 0$ and $k_{x,y} = k_{x,p(x)} + k_{p(x),p(p(x))} + \ldots +k_{p^{m-1}(x),p^m(x)}$ otherwise.

If $(x,y)\in R$ and x < y, then există m such that $x = p^m(y)$ and thus $k_{x,y} = (k_{p^m(y),p^{m-1}(y)}+\ldots+ k_{p(p(y)),p(y)} + k_{p(y),y})$ mod $k_{x,x} = ((k_{x,x} - k_{p^{m-1}(x),p^m(x)}) + \ldots +(k_{x,x} - k_{p(y),p(p(y))}) +(k_{x,x} - k_{y,p(y)}))$ mod $k_{x,x}$, if $k_{x,x} \neq 0$, and $k_{x,y} = - k_{p(y),p(p(y))} - \ldots - k_{p(y),p(p(y))} - k_{y,p(y)}$ otherwise.

If there is a function $\varphi: X \to X$ such that $G = G(X,\varphi)$, then $k_{x,x} \neq 0$ for all x (Since X is finite, it follows that $\varphi^m(x)$ can only take a finite number of distinct values, when m varies. Thus we can choose $k_{x,x}>0$ to be the smallest positive number k such that there is $n\in\mathbf{N}$ with the property that $\varphi^{k+n}(x) = \varphi^n(x)$).

For implementation in Maple we use a list L with n pair of elements:

$\varphi^n(x)$).

Pentru implementare în Maple utilizăm o listă L cu n perechi of elemente:

L[i][1] = eticheta orbit ce conţine x_i;

L[i][2] = k_{x_i,x_i} dacă x_i este primul element din orbita [x_i], şi $k_{x_i,p(x_i)}$ în caz contrar.

Etichetele orbitelor sunt din {1, 2, …, n_{orbits}} (n_{orbits} = număr orbite) şi sunt alese astfel încât dacă $i < j$, x_i (respectiv, x_j) este primul element al orbitei o_q (respectiv o_p), atunci $q \le p$.

Procedura Maple grupoid_data(phi) construieşte lista L pentru grupoidul G(X,φ) asociat unei funcţii φ specificată ca o listă de valori [y_1, y_2, …, y_n] cu semnificaţia $y_i = \varphi(x_i)$ pentru orice i=1..n. Utilizăm pentru exemplificare următoarea funcţie:

L[i][1] = the label of orbit that contains x_i;

L[i][2] = k_{x_i,x_i} if x_i is the first element of the orbit [x_i], and $k_{x_i,p(x_i)}$ otherwise.

The labels of the orbits are elements of the set {1, 2, …, n_{orbits}} (n_{orbits} = number of orbits) and are chosen such that if $i < j$, x_i (respectively, x_j) is the first element of the orbit o_q (respectively o_p), then $q \le p$.

The Maple procedure groupoid_data(phi) construct the list L for the groupoid G(X,φ) associate with a function φ specified as a list of values [y_1, y_2, …, y_n] meaning that $y_i = \varphi(x_i)$ for all i=1..n. We identify the elements with their indexes in the list. We use the following function:

```
>phi:=[15,14,13,12,2,11,13,8,7,6,6,5,3,5,1];
          φ:= [15, 14, 13, 12, 2, 11, 13, 8, 7, 6, 6, 5, 3, 5, 1]
  1->15, 2->14, 3->13, 4->12, 5->2, 6->11, 7->13, 8->8, 9->7, 10->6, 11-
>6,
```

Procedurile cycle_detection şi equiv_detection utilizate în groupoid_data au fost definite în [1] şi se bazează pe algoritmul lui Brent [2] pentru rezolvarea aşa numitei probleme „cycle detection problem" pentru φ.

The procedures cycle_detection and equiv_detection used in groupoid_data(phi) were definite in [1] and are based on Brent algorithm [2] to solve the cycle detection problem for given φ.

```
groupoid_data:=proc(phi)
local marko, kx,n, Lxy, x, y, orbit, test;
  n:=nops(phi); orbit:=0;marko:=[seq([-1,-1],i=1..n)];
  kx:=[seq([-1,-1],i=1..n)];
  for x from 1 to n do
     kx[x]:=cycle_detection(phi,x);test:=0; y:=x-1;
     while y>=1 do
       if(kx[y][1]=kx[x][1]) then
         Lxy:=equiv_detection(phi,x,y,kx[x][1]+max(kx[x][2],kx[y][2]));
         if(Lxy<>NULL) then
            marko[x][1]:=marko[y][1];marko[x][2]:=Lxy[1]; test:=1; y:=0
         end if
       end if;
       y:=y-1
     end do;
     if test=0 then
        orbit:=orbit+1;marko[x][1]:=orbit; marko[x][2]:=kx[x][1]
     end if
  end do;
  RETURN(marko)
end proc;
```

Să presupunem că subgrupoidul G al lui X×**Z**×X este implementat ca o listă gd de n perechi de elemente. Pentru a determina primul element al fiecărei orbite, predecesorul şi sucesorul (dacă există) al fiecărui element putem utiliza procedura orbit_first_predecessor_successor(gd): F[i] este primul element din orbita i, Lp[i] predecesorul (dacă există) al lui i in the orbit of [i] iar Lp[i] = i, în caz contrar, Ls[i] este succesorul (dacă există) al lui i în orbita [i] şi Ls[i] = -1, în caz contrar.

Let us assume that a subgroupoid G of X×**Z**×X is encoded by a list gd of n pair of elements. To determine the first element of each orbit, the predecessor and successor (if exists) of each element we can use the procedure orbit_first_predecessor_successor(gd): F[i] is first element in the orbit i, Lp[i] is the predecessor (if exists) of i in the orbit of [i] and Lp[i] = i, othewise, Ls[i] is the successor (if exists) of i in the orbit of [i] and Ls[i] = -1, otherwise.

```
orbit_first_predecessor_successor:=proc(gd)
local n,j,p,i,F,Lp,Ls;
   n:=nops(gd); Lp:=[seq(-1,i=1..n)]; Ls:=[seq(-1,i=1..n)];
   p:=0; F:=[];
   for i from 1 to n do
      if gd[i][1]>p then
           p:=gd[i][1]; Lp[i]:=i;F:=[op(F),i]
      else
           j:=i-1;
           while gd[j][1]<>gd[i][1] do j:=j-1 end do;
           Lp[i]:=j;
           if Ls[j]<0 then Ls[j]:=i end if
      end if
   end do;
   RETURN([F,Lp,Ls])
end proc;
```

Procedura compute_kxy(gd,x,y) returnează $k_{x,y}\in \mathbf{Z}$ pentru un grupoid G reprezentatde gd.

The procedure compute_kxy(gd,x,y) return $k_{x,y}\in \mathbf{Z}$ for a groupoid G encoded by gd.

```
compute_kxy:=proc(gd,x,y)
  local kx,kxy,z,FPS;
  FPS:=orbit_first_predecessor_successor(gd); kx:=gd[FPS[1][gd[x][1]]][2];
  if x=y then RETURN(kx) end if;
  if gd[x][1]<>gd[y][1] then RETURN(NULL);
     else
     if x>y then
        kxy:=gd[x][2]; z:=FPS[2][x];
        while z>y do
           kxy:=irem(kxy+gd[z][2],kx); z:=FPS[2][z]
        end do;
     else
           kxy:=gd[y][2];z:=FPS[2][y];
           while z>x do
              kxy:=irem(kxy+gd[z][2],kx); z:=FPS[2][z];
           end do;
        kxy:= kx-kxy
     end if
  end if;
  RETURN(kxy)
end proc;
```

Rezultatele lor pentru funcţia φ fixată anterior sunt:

Their result for the function φ priory fixed are:

```
> groupoid_data(phi);
[[1, 2], [2, 3], [3, 2], [2, 0], [2, 1], [4, 2], [3, 0], [5, 1], [3, 1],
[4, 1], [4, 0], [2, 1], [3, 0], [2, 0], [1, 1]]
```

```
> orbit_first_predecessor_successor(groupoid_data(phi));
[[1, 2, 3, 6, 8], [1, 2, 3, 2, 4, 6, 3, 8, 7, 6, 10, 5, 9, 12, 1],
[15, 4, 7, 5, 12, 10, 9, -1, 13, 11, -1, 14, -1, -1, -1]]
```

```
> compute_kxy(groupoid_data(phi),5,14);
                                        2
```

Procedura det_y_x(gd,x) returnează lista elementelor y∈X astfel încât (x,1,y)∈G, unde G este grupoidul reprezentat de gd.

The procedure det_y_x(gd,x) returns the list of elements y∈X such that (x,1,y)∈G, where G is the groupoid whose data are stored in gd.

```
det_y_x:=proc(gd,x)
local y,kxy,kx,Ly,test,FPS;
   FPS:=orbit_first_predecessor_successor(gd);
   kx:=gd[FPS[1][gd[x][1]]][2]; Ly:=[];
   if kx=1 then
        for y from 1 to nops(gd) do
          if gd[y][1]=gd[x][1] then Ly:=[op(Ly),y] end if
        end do;
   else
        y:=x; kxy:=0;
        while FPS[2][y]<>y do
          kxy:=irem(kxy+gd[y][2],kx);
          if kxy=1 then Ly:=[op(Ly),FPS[2][y]] end if;
          y:=FPS[2][y];
        end do;
        y:=FPS[3][x];
        kxy:=0;
        while y>0 do
          kxy:=irem(kxy+kx-gd[y][2],kx);
          if kxy=1 then Ly:=[op(Ly),y] end if;
          y:=FPS[3][y];
        end do;
   end if;
   RETURN(Ly)
end proc;
```

```
> det_y_x(groupoid_data(phi),12);
                                       [5]
```

Dacă există un element x∈X astfel încât det_y_x(gd, x) întoarce [], atunci grupoidul G reprezentat de gd nu provine dintr-un sistem dinamic discret.

If there is an element x∈X such that det_y_x(gd, x) returns [], then the groupoid G encoded by gd does not arise from a discete dynamical system.

Mai eficient, procedura check_dynamical(gd) returnează 1 dacă grupoidul reprezentat prin gd provine dintr-un sistem dinamic discret şi 0, în caz contrar.

More efficiently, the procedure check_dynamical(gd) returns 1 if the groupoid encoded by gd arises from a discrete dynamical system and 0, otherwise.

```
check_dynamical:=proc(gd)
local x,y,kxy,kx,test,FPS;
   FPS:=orbit_first_predecessor_successor(gd);
   for x from 1 to nops(gd) do
      kx:=gd[FPS[1][gd[x][1]]][2]; test:=0;
      if kx<>1 then
         y:=x; kxy:=0;
         while test=0 do
           kxy:=irem(kxy+gd[y][2],kx);
           if kxy=1 then
             test:=3;
           else
             if FPS[2][y]=y then test:=1
             else y:=FPS[2][y]
             end if
           end if
         end do;
         if (test<3) şi (FPS[3][y]<=0) then test:=2;
         else y:=FPS[3][y]; kxy:=0
         end if;
         while test<2 do
           kxy:=irem(kxy+kx-gd[y][2],kx);
           if kxy=1 then
             test:=3;
           else
             if FPS[3][y]<=0 then test:=2 else y:=FPS[3][y]end if
           end if
         end do;
      else test:=3;
      end if;
      if test<>3 then RETURN(0) end if;
   end do;
   RETURN(1);
end proc;
```

Procedura hash_table_subgrupoids(gd) afişează toate funcţiile φ pe X care dau naştere unui subgrupoid G(X, φ) of G (grupoidul reprezentat prin gd).

The procedure hash_table_subgroupoids(gd) prints all the functions φ on X that gives rise to a subgroupoid G(X, φ) of G (the groupoid encoded by gd).

```
hash_table_subgroupoids:=proc(gd)
local n,i,j, L, num;
   n:=nops(gd); L:=[];
   for i from 1 to n do L:=[op(L),det_y_x(gd,i)] end do;
   print(seq(cat(convert(i, name),` --> `, seq(cat(convert(L[i][j],
              name),` | `), j = 1..nops(L[i])-1),convert(L[i][nops(L[i])],
                                                        name)), i=1..n)) ;
   num:=1; for i from 1 to n do num:=num*nops(L[i]) end do;
   print(cat(`Number of functions: `, convert(num,name), `.`))
end proc;
```

```
> hash_table_subgroupoids(groupoid_data(phi));
1 -> 15, 2 -> 12 | 14, 3 -> 9 | 13, 4 -> 12 | 14, 5 -> 4 | 2,
6 -> 10 | 11, 7 -> 9 | 13, 8 -> 8, 9 -> 7 | 3, 10 -> 6, 11 -> 6,
12 -> 5, 13 -> 7 | 3, 14 -> 5, 15 -> 1
                        Number of functions: 256.
```

Procedura orbits_rel(gd) determină lista orbitelor (claselor) relativ la relaţia: x~y <=> (x,0,y)∈G (G grupoidul reprezentat de gd).

The procedure orbits_rel(gd) returns the list of orbits (classes) with respect to the relation: x~y iff (x,0,y)∈G (G the groupoid whose data are stored in gd).

```
orbits_rel:=proc(gd)
local x,y,kxy,kx,Lx,test,L,visit,FPS;
   FPS:=orbit_first_predecessor_successor(gd);
   L:=[];
   visit:=[seq(1,i=1..nops(gd)+1)];
   x:=1;
   while x<=nops(gd) do
      kx:=gd[FPS[1][gd[x][1]]][2];
      Lx:=[x];
      y:=FPS[3][x];
      kxy:=0;
      while y>0 do
         kxy:=irem(kxy+kx-gd[y][2],kx);
         if kxy=0 then Lx:=[op(Lx),y]; visit[y]:=0; end if;
         y:=FPS[3][y];
      end do;
      L:=[op(L),Lx];
      x:=x+1;
      while visit[x]=0 do x:=x+1 end do;
   end do;
   RETURN(L)
end proc;
```

```
> orbits_rel(groupoid_data(phi));
[[1], [2, 4], [3, 7], [5], [6], [8], [9, 13], [10, 11], [12,14],[15]]
```

Dacă grupoidul G reprezentat gd provine dintr-un sistem dinamic discret, atunci procedura hash_table_variant(gd) afişează toate funcţiile φ:X→X cu proprietatea că G(X,φ)=G şi care în plus sunt constante pe clasele de echivalenţă relativ la relaţia: x~y <=> (x,0,y)∈G.

If the groupoid G whose data are stored in gd arises from a dynamical system, then the procedure hash_table_variant(gd) prints all functions φ:X→X with the property that G(X,φ)=G and that are constant on the equivalence classes of the relation x~y iff (x,0,y)∈G.

```
hash_table_variant:=proc(gd)
local n,i,j, L, num;
 n:=nops(gd); L:=det_functions(gd);
 print(seq(cat(convert(i, name),` --> `, `y`, convert(L[1][i],
                                  name)),  i=1..n)) ;print(`where, `);
 print(seq(cat(` y`,convert(i,name),` = `,
           seq(cat(convert(L[2][i][j], name), `|`),j=1..nops(L[2][i])-1),
               convert(L[2][i][nops(L[2][i])],name)),i=1..nops(L[2])));
 num:=1; for i from 1 to nops(L[2]) do num:=num*nops(L[2][i]) end do;
 print(cat(`Number of functions: `, convert(num,name), `.`))
end proc;
```

```
> hash_table_variant(groupoid_data(phi));
1 -> y1, 2 -> y2, 3 -> y3, 4 -> y2, 5 -> y4, 6 -> y5, 7 -> y3, 8 -> y6, 9
-> y7, 10 -> y8, 11 -> y8, 12 -> y9, 13 -> y7, 14 -> y9, 15 -> y10
where,
y1 = 15, y2 = 12|14, y3 = 9|13, y4 = 4|2, y5 = 10|11, y6 = 8, y7 = 7|3,
y8 = 6,  y9 = 5,  y10 = 1
                       Number of functions: 32.
```

Dacă A este lista elementelor unei submulţimi$\{a_1,a_2,\ldots,a_m\}\subset X$, gd este lista ce reprezintă un subgrupoid $G\subset X\times\mathbf{Z}\times X$ atunci procedura groupoid_reduction(gd,A) determină lista gda ce stochează datele grupoidului

$G|A=\{(x,k,y)\text{: } (x,ky)\in G \text{ and } x,y\in A\}$ (contracţia lui G la A) privit ca subgrupoid al lui $A\times\mathbf{Z}\times A$.

If A is the list of elements of a subset $\{a_1,a_2,\ldots,a_m\}\subset X$, gd is the list that stores the data of a subgroupoid $G\subset X\times\mathbf{Z}\times X$, then the procedure groupoid_reduction(gd,A) returns the list gda that stores the data of the groupoid

$G|A=\{(x,k,y)\text{: } (x,ky)\in G \text{ and } x,y\in A\}$ (the reduction of G to A) seen as a subgroupoid of $A\times\mathbf{Z}\times A$.

```
groupoid_reduction:=proc(gd,A)
local gda,i,j,a,orbit,FPS;
   gda:=[seq([-1,-1],i=1..nops(A))];
   a:=sort(A);
   orbit:=0;
   FPS:=orbit_first_predecessor_successor(gd);
   for i from 1 to nops(a) do
      j:=i-1;
      while j>0 do
           if gd[a[i]][1]=gd[a[j]][1] then
              gda[i][1]:=gda[j][1];
              gda[i][2]:=compute_kxy(gd,a[i],a[j]);j:=-2;
           end if;
           j:=j-1;
      end do;
      if j>-2 then
           orbit:=orbit+1;gda[i][1]:=orbit;
           gda[i][2]:=gd[FPS[1][gd[a[i]][1]]][2]
      end if
   end do;
   RETURN(gda)
end proc;
```

```
> groupoid_reduction(groupoid_data(phi),[3,4,7,13,15]);
```

[[1, 2], [2, 3], [1, 0], [1, 1], [3, 2]]

```
> check_dynamical(groupoid_reduction(groupoid_data(phi),[3,4,7,13,15]));
                                 0
```

```
> groupoid_reduction(groupoid_data(phi),[3,4,5,7,13,14]);
        [[1, 2], [2, 3], [2, 1], [1, 0], [1, 1], [2, 1]]
```

```
> hash_table_variant(groupoid_reduction(groupoid_data(phi),
[3,4,5,7,13,14]));
```

1 --> y1, 2 --> y2, 3 --> y3, 4 --> y1, 5 --> y4, 6 --> y5

where,

y1 = 5, y2 = 6, y3 = 2, y4 = 4|1, y5 = 3

Number of functions: 2.

Ţinând cont că elementele lui A={3,4,5,7,13,14} au fost renumerotate funcţiile φ_A pentru care $G(X,\varphi)|A=G(A,\varphi_A)$ şi care sunt constante pe clasele de echivalenţă ale lui R_φ sunt: `3->13, 4->14, 5->4, 7->13, 13->3|7, 14->5.`

REFERINŢE

[1] Bărbăcioru, I. C. şi Buneci, M. Using Maple to study the groupoid associated to a singly generated dynamical system. I, Annals of the "Constantin Brâncuşi" University of Târgu-Jiu. Egineering Series. No. 3(2010), 308-317.

[2] Brent, R. P. An improved Monte Carlo factorization algorithm, BIT 20 (1980) 176–184.

[3] Buneci M., Groupoids and irreversible discrete dynamical systems I, Fiabilitate and durabilitate (Fiability & durability), No. 1/2012, 350-355.

[4] Renault J., Cuntz-like algebras, in Operator theoretical methods, 371-386, Theta Found., Bucharest, 2000.

Taking into account that the elements of A={3,4,5,7,13,14} were relabeled, the functions φ_A such that $G(X,\varphi)|A=G(A,\varphi_A)$ and that are constant on the equivalence classes of the relation R_φ are `3->13, 4->14, 5->4, 7->13, 13->3|7, 14->5.`

REFERENCES

[1] Bărbăcioru, I. C. and Buneci, M. Using Maple to study the groupoid associated to a singly generated dynamical system. I, Annals of the "Constantin Brâncuşi" University of Târgu-Jiu. Egineering Series. No. 3(2010), 308-317.

[2] Brent, R. P. An improved Monte Carlo factorization algorithm, BIT 20 (1980) 176–184.

[3] Buneci M., Groupoids and irreversible discrete dynamical systems I, Fiabilitate and durabilitate (Fiability & durability), No. 1/2012, 350-355.

[4] Renault J., Cuntz-like algebras, in Operator theoretical, 371-386, Theta Found., Bucharest, 2000.