A PACKAGE OF PROGRAMS FOR DETERMINATION OF SOME CLASSES OF SUBGROUPOIDS

Gheorghe Ivan and Mihai Ivan

West University of Timişoara, Seminarul de Algebră, 4, Bd. V. Pârvan, 300223, Timişoara, Romania
(e-mail: ivan @ math.uvt.ro)

Abstract. In this paper we give three programs on computer for finding the subgroupoids, wide subgroupoids and normal subgroupoids of a finite groupoid. Applying these programs for groups, we can determine all subgroups and normal subgroups of a given finite group.

2000 Mathematics Subject Classification: 20L13, 68W20.
Key words and phrases: groupoid, disjoint unions of groupoids, subgroupoid, wide subgroupoid, normal subgroupoid.

1. INTRODUCTION

The concept of groupoid has introduced by H. Brandt [Math. Ann. 96, 360 -366 (1926; JFM 52.0110.09)]. A groupoid is an algebraic structure determined by a partially composition law and a nonempty set of units. In the language of categories, a groupoid is a small category in which all morphisms are invertible. For more details and references about groupoids the reader can be consult the papers ([1]-[3], [6]-[8]).

The plan of this paper is as follows. In the first section we give some preliminary concepts concerning groupoids. In the second section we present an algorithm for finding the subgroupoids of a groupoid. This algorithm is implemented on computer and we obtain the program $BGroidAP^2$. This program has published in [4; Zbl 1109.20310]. In the Section 3 and Section 4 we give the programs $BGroidAP^3$ and $BGroidAP^4$ for to determine the wide subgroupoids resp. the normal subgroupoids of a finite groupoid. We illustrate the utilization of these programs on some finite groupoids.

The programs exposed in this paper represent an essential tool for the study of finite groupoids.

2. CLASSES OF SUBGROUPOIDS

Let $(G, G_0)$ be a pair of nonempty sets with $G_0 \subseteq G$, endowed with the surjections $\alpha, \beta : G \to G_0$, called the source and the target map, respectively, a (partial) composition law $\mu : G(2) \to G, (x, y) \to \mu(x, y)$, where $G(2) = \{(x, y) \in G \times G | \beta(x) = \alpha(y)\}$ and an injection $\iota : G \to G, x \to \iota(x)$ called
the inversion map. We write sometimes $x \cdot y$ or $xy$ for $\mu(x, y)$ and $x^{-1}$ for $\iota(x)$. The elements of $G(2)$ are called composable pairs of $G$.

**DEFINITION 2.1.** ([4]) (i) The 5-tuple $(G, \alpha, \beta, \mu; G_0)$ is a semigroupoid, if the composition law is associative, i.e. $(xy)z = x(yz)$, for all $x, y, z \in G$ such that the products $(xy)z$ and $x(yz)$ are defined.

(ii) A monoidoid is a semigroupoid $(G, \alpha, \beta, \mu; G_0)$ such that the identities property holds, i.e. for each $x \in G$ we have $(\alpha(x), x), (x, \beta(x)) \in G(2)$ and $\alpha(x)x = x\beta(x) = x$.

(iii) The 6-tuple $(G, \alpha, \beta, \mu, \iota; G_0)$ is a groupoid or a $G_0$-groupoid, if $(G, \alpha, \beta, \mu, \iota; G_0)$ is a monoidoid such that the inverses property holds, i.e. for each $x \in G$ we have $(x^{-1}, x), (x, x^{-1}) \in G(2)$ and $x^{-1}x = \beta(x), xx^{-1} = \alpha(x)$.

□

**Remark 2.1.** The definition of the groupoid is equivalent as the one used in the paper ([2]).

The element $\alpha(x)$ [resp. $\beta(x)$] denoted sometimes by $u_l(x)$ [resp. $u_r(x)$] is the left unit [resp. right unit] of $x \in G$. The set $G_0$ is called the unit set of $G$. A $G_0$-groupoid $G$ will be denoted by $(G, \alpha, \beta; G_0)$ or $(G; G_0)$. The maps $\alpha, \beta, \mu$ and $\iota$ are called the structure functions of $G$.

If $(G, \alpha, \beta; G_0)$ is a groupoid, then the following properties hold (see [3]):

1. $\alpha(u) = \beta(u) = u, u \cdot u = u$ and $\iota(u) = u$ for all $u \in G_0$;
2. $\alpha(xy) = \alpha(x)$ and $\beta(xy) = \beta(y), \forall (x, y) \in G(2)$;
3. $\alpha(x^{-1}) = \beta(x), \beta(x^{-1}) = \alpha(x)$ for all $x \in G$;
4. $G(u) = \{x \in G|\alpha(x) = \beta(x) = u\}$ is a group under the restriction of $\mu$ to $G(u)$, called the isotropy group at $u$ of $G$.

□

**Example 2.1.** (i) A group $G$ having $e$ as unity, is just a $\{e\}$- groupoid in the following way: the maps $\alpha, \beta : G \to G_0$ and $\iota : G \to G$ are given by $\alpha(x) = \beta(x) = e, \iota(x) = x^{-1}$ for all $x \in G$; for all $x, y \in G$ the element $x \cdot y$ is the product of elements $x$ and $y$ in the group $G$. Conversely, every groupoid $G$ with one unit is a group.

(ii) The null groupoid over a set. Any nonempty set $X$ may be regarded as a groupoid on itself with the groupoid structure $G = G_0 = X, \alpha = \beta = \iota = Id_X; x, y \in X$ are composable iff $x = y$ and we define $x \cdot x = x$.

□

**Example 2.2.** (i) The groupoid $\mathcal{T}_{inj}(S, X)$. For a nonempty set $X$ denote by $\mathcal{T}_{inj}(S, X) = \{f : S \to X|\forall S, \emptyset \neq S \subseteq X, f\text{ is injective}\}$. For $f \in \mathcal{T}_{inj}(S, X)$, let $D(f)$ be the domain of $f$ and let $R(f) = f(D(f))$. For $G = \mathcal{T}_{inj}(S, X)$, let $G(2) = \{(f, g) \in G \times G|R(f) = D(g)\}$ and for $(f, g) \in G(2)$ define $\mu(f, g) = g \circ f$. If $Id_S$ denotes the identity map on $S$, then $G_0 = \{Id_S|\emptyset \neq S \subseteq X\}$ is the set of units of $G$. The maps $\alpha, \beta : G \to G_0$ and $\iota : G \to G$ are defined by $\alpha(f) = Id_{D(f)}, \beta(f) = Id_{R(f)}$ and $\iota(f) = f^{-1}$. Thus $\mathcal{T}_{inj}(S, X)$ is a groupoid, called the groupoid of injective functions from the nonempty sets $S$ of $X$ into $X$.

In particular, if $X = \{1, 2, \ldots, n\}$, the groupoid of the injective functions defined on the subsets of $\{1, 2, \ldots, n\}$ is called the symmetric groupoid of degree $n$ and is denoted by $S_n$; for several properties of $S_n$, see [5].
(ii) Let $Ox$ be a system of cartesian coordinates in a plane. We consider the subsets $Ox = \{(x, 0) \in \mathbb{R}^2 | (\forall) x \in \mathbb{R}\}$ and $Oy = \{(0, y) \in \mathbb{R}^2 | (\forall) y \in \mathbb{R}\}$ of $X = \mathbb{R}^2$. Let $G = \{f_1 = 1d_{Ox}, f_2 = 1d_{Oy}, f_3 = \sigma_{Ox}, f_4 = \sigma_{Oy}\} \subset \mathcal{F}_{\text{inj}}(S, \mathbb{R}^2)$ where $f_3 : Ox \to Oy, f_4 : Oy \to Ox$ are defined by $f_3(x, 0) = (0, x)$ and $f_4(0, y) = (y, 0)$ ($\sigma_{Ox}$ resp. $\sigma_{Oy}$ is called the saltus function defined on $x$-axis resp. $y$-axis).

For the composable pairs of $G$, the map $\mu$ is defined by: $\mu(f_1, f_1) = f_1; \mu(f_1, f_3) = f_3; \mu(f_2, f_2) = f_2; \mu(f_2, f_4) = f_4; \mu(f_3, f_2) = f_3; \mu(f_3, f_4) = f_1; \mu(f_4, f_1) = f_4; \mu(f_4, f_3) = f_2$.

The unit set of $G$ is $G_0 = \{f_1 = 1d_{Ox}, f_2 = 1d_{Oy}\}$ and $\alpha, \beta : G \to G_0$, $\iota : G \to G$ are given by $\alpha(f_j) = \beta(f_j) = \iota(f_j) = f_j$ for $j = 1, 2; \alpha(f_3) = \beta(f_3) = f_1; \alpha(f_4) = \beta(f_3) = f_2; \iota(f_3) = f_4$ and $\iota(f_4) = f_3$. It is easy to verify that $(G; G_0)$ is a groupoid, denoted by $\mathcal{F}_{(4;2)}(\mathbb{R}^2)$ and called the groupoid of saltus functions defined on the axes of coordinates in a plane.

Example 2.3. If $\{G_i | i \in I\}$ is a disjoint family of groupoids, let $G = \bigcup_{i \in I} G_i, G(2) = \bigcup_{i \in I} G_i(2)$ and $G_0 = \bigcup_{i \in I} G_{i,0}$, where $G_{i,0}$ is the unit set of $G_i$. Here, $x, y \in G$ may be composed iff they lie in the same groupoid $G_i$ and they are composable in $G_i$. This groupoid is denoted by $\coprod_{i \in I} G_i$ and is called the disjoint union of groupoids $G_{i, i} \in I$. In particular, the disjoint union of groups $G_{i, i} \in I$ is a groupoid.

A finite $G_0$-groupoid $G$ such that $|G| = n$ and $|G_0| = m$ is called $(n; m)$-groupoid or finite groupoid of type $(n; m)$. We will sometimes denote a finite groupoid of type $(n; m)$ by $G_{(n; m)}$.

Example 2.4. (i) Each finite groupoid of type $(n; 1)$ is a group.

(ii) Each finite groupoid of type $(n; n)$ is a null groupoid.

(iii) The groupoid $\mathcal{F}_{(4;2)}(\mathbb{R}^2)$ is a $(4; 2)$-groupoid.

DEFINITION 2.2. (i) Let $(G, \alpha, \beta; G_0)$ be a groupoid. A pair $(H; H_0)$ of nonempty sets such that $H \subseteq G$ and $H_0 \subseteq G_0$ is a subgroupoid of $G$, if the following conditions hold: (1) $\alpha(H) = \beta(H) = H_0$; (2) for all $x, y \in H$ such that $xy$ is defined, we have $xy \in H$ and (3) for all $x \in H$, we have $x^{-1} \in H$.

(ii) A subgroupoid $(H; H_0)$ of a $G_0$-groupoid $G$ with property that $H_0 = G_0$ is called wide subgroupoid of $G$.

(iii) A wide subgroupoid $(H; G_0)$ of a groupoid $(G; G_0)$ is called a normal subgroupoid of $G$, if for all $x \in G$ and $h \in H$ such that the product $xhx^{-1}$ is defined, we have $xhx^{-1} \in H$.

The intersection of any collection of subgroupoids of a groupoid is itself a subgroupoid of that groupoid. If $G$ is a $G_0$-groupoid and $X$ is a nonempty subset of $G$, then the intersection of all subgroupoids of $G$ which contain $X$ is a subgroupoid, denoted by $< X >$ and called the generated subgroupoid of $G$ by $X$.

Example 2.5. (i) In a group $G$, every subgroupoid (in fact, subgroup) is a wide subgroupoid and conversely.

(ii) If $G$ is a $G_0$-groupoid, then $G_0$ is a normal subgroupoid of $G$, called the null subgroupoid of $G$. 

3
(iii) Let be the Klein 4-group $K_4 = \{(1), \sigma = (12)(34), \tau = (13)(24), \sigma \circ \tau = (14)(23)\} \subset S_4$ (it is a subgroup of the symmetric group $S_4$ of degree 4). We have $\sigma^2 = \tau^2 = (1)$ and $\tau \circ \sigma = \sigma \circ \tau$. We consider the disjoint union $G = K_4 \coprod F_{(4;2)}$ of the group $K_4$ with the groupoid $F_{(4;2)}(\mathbf{R}^2)$ given in Example 1.2 (ii). We have that $G = \{(1), \sigma, \tau, \sigma \circ \tau, f_1 = \text{Id}_{Ox}, f_2 = \text{Id}_{Oy}, f_3 = \sigma_{Ox}, f_4 = \sigma_{Oy}\}$ is a $G_0$-groupoid of type $(8;3)$, where $G_0 = \{(1), f_1, f_2\}$. It is easy to verify that $K_4$ and $F_{(4;2)}(\mathbf{R}^2)$ are subgroupoids of $G$. Also, $N_1 = K_4 \coprod \{f_1, f_2\}$ and $N_2 = \{(1)\} \coprod F_{(4;2)}(\mathbf{R}^2)$ are wide subgroupoids of $G$. Moreover, $N_1$ and $N_2$ are normal subgroupoids of $G$. \hfill \Box

3. ALGORITHM FOR DETERMINATION OF SUBGROUPOIDS. THE $BGroidAP2$ PROGRAM

We consider a given finite universal algebra $(G, \alpha, \beta, \mu, \iota; G_0)$ such that $|G| = n$ and $|G_0| = m$ with $1 \leq m \leq n$. We denote the elements of $G$ by $a_1, a_2, \ldots, a_m, a_{m+1}, \ldots, a_n$ such that $G_0 = \{a_1, a_2, \ldots, a_m\}$.

We give an algorithm for decide if the universal algebra $(G, \alpha, \beta, \mu, \iota; G_0)$ is a $G_0$-groupoid and for determine the subgroupoids of $G$. This algorithm is constituted by the following stages.

Stage I. We introduce the initial data: $n = |G|, m = |G_0|$; the functions $\alpha, \beta, \iota$ and $\mu$ given by its tables of structure.

Stage II. Test if the universal algebra $(G, \alpha, \beta, \mu, \iota; G_0)$ is a groupoid.

For this, the following steps are executed:

step 1. $(G, \alpha, \beta, \mu, \iota; G_0)$ is a structure well-defined, i.e. $\alpha, \beta$ are surjections, $\iota$ is injective and $\mu$ is defined on $G_2$ with values in $G$;

step 2. $(G, \alpha, \beta, \mu; G_0)$ is a semigroupoid;

step 3. the semigroupoid $(G, \alpha, \beta, \mu; G_0)$ is a monoidoid;

step 4. the monoidoid $(G, \alpha, \beta, \mu; G_0)$ is a groupoid.

step 5. If the above steps are satisfied, make the tables of the structure functions $\alpha, \beta, \iota$ and $\mu$ and write the message “$G$ is a groupoid”.

Stage III. Determine the subgroupoids of $G$. The following steps must be executed:

step 1. Write all nonempty subsets $X$ of $G$;

step 2. Determine the subgroupoid $< X >$ of $G$ generated by $X$;

step 3. Sort by cardinal all subgroupoids determined in the step 2;

step 4. List the subgroupoids produced in the above step;

step 5. For each subgroupoid make its subgroupoid table.

Let us we present the correspondence between the initial data and input data:
\[ G = \{ a_1, a_2, \ldots, a_m, a_{m+1}, \ldots, a_n \} \iff \{ 1, 2, \ldots, m, m+1, \ldots, n \} \]

The absence of an element from the arrow "\( j \)" and the column "\( k \)" of the table of \( \mu \) indicates the fact that the pair \((a_j, a_k)\) in \( G \times G \) is not composable. The element \( a_j k = \mu(a_j, a_k) \) is represented by 0 in the table of input data, if the product \( a_j \cdot a_k \) is not defined.

**Example 3.1.** Let the groupoid \( G = K_4 \coprod \mathcal{F}_{(4;2)}(\mathbb{R}^2) \), see Example 2.5 (iii). We have \( G = \{ a_1 = (1), a_2 = f_1, a_3 = f_2, a_4 = \sigma, a_5 = \tau, a_6 = \sigma \circ \tau, a_7 = f_3, a_8 = f_4 \} \) and the correspondence between the initial data and input data are the following:

\[
\begin{align*}
G &= \{ a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8 \} \iff \{ 1, 2, 3, 4, 5, 6, 7, 8 \} \\
|G| &= 8 \iff 8 \\
|G_0| &= 3 \iff 3
\end{align*}
\]

| \( \mu \) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| a_1 | a_2 | a_3 | a_4 | a_5 | a_6 | a_7 | a_8 |
| a_2 | a_3 | a_4 | a_5 | a_6 | a_7 | a_8 |
| a_3 | a_4 | a_5 | a_6 | a_7 | a_8 |
| a_4 | a_5 | a_6 | a_7 | a_8 |
| a_5 | a_6 | a_7 | a_8 |
| a_6 | a_7 | a_8 |
| a_7 | a_8 |
| a_8 | a_1 | a_2 | a_3 | a_4 | a_5 | a_6 | a_7 | a_8 |

\[
\begin{align*}
\begin{array}{cccccccc}
\alpha(a_k) & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\
\beta(a_k) & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\
\gamma(a_k) & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{cccccccc}
\mu \circ \sigma & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\
\mu \circ \tau & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{cccccccc}
\alpha(a_k) & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\
\beta(a_k) & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\
\gamma(a_k) & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{cccccccc}
\mu \circ \sigma & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\
\mu \circ \tau & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\
\end{array}
\end{align*}
\]
The implementation of the above algorithm on computer is realized in the program \textit{BGroidAP2}, which is composed from two modules denoted by \textit{unit21.dfm} and \textit{unit21.pas}. The module \textit{unit21.pas} is consists from the principal program followed of procedures and functions.

The principal program of the module \textit{unit21.pas} is constituted from the following lignes.

| Lignes | The module \textit{unit21.pas} |
|--------|--------------------------------|
| 001    | unit Unit1;                   |
| 002    | interface                     |
| 003    | uses                          |
| 004    | Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, Grids, DBGrids, ShellAPI, Db, DBTables, StdCtrls, Menus, ExtCtrls, ComCtrls, ToolWin, Spin; |
| 005    | const                         |
| 006    | nmax = 200;                   |
| 007    | type                          |
| 008    | TSubSet = Set of Byte;        |
| 009    | TForm1 = class(TForm)         |
| 010    | MainMenu1: TMainMenu;         |
| 011    | File1: TMenuItem;             |
| 012    | OpenFile1: TMenuItem;         |
| 013    | SaveFile1: TMenuItem;         |
| 014    | GroupBox1: TGroupBox;         |
| 015    | StringGrid1: TStringGrid;     |
| 016    | StringGrid2: TStringGrid;     |
| 017    | GroupBox2: TGroupBox;         |
| 018    | StringGrid3: TStringGrid;     |
| 019    | StringGrid4: TStringGrid;     |
| 020    | OpenDialog1: TOpenDialog;     |
| 021    | SaveDialog1: TSaveDialog;     |
| 022    | Splitter1: TSplitter;         |
| 023    | Splitter2: TSplitter;         |
| 024    | ToolBar1: TToolBar;           |
| 025    | ToolBar2: TToolBar;           |
| 026    | Splitter4: TSplitter;         |
| 027    | ToolButton1: TToolButton;     |
| 028    | ToolButton2: TToolButton;     |
| 029    | New1: TMenuItem;              |
| 030    | ToolBar3: TToolBar;           |
| 031    | ToolButton4: TToolButton;     |
| 032    | ToolButton5: TToolButton;     |
| 033    | Label2: TLabel;               |
The module \textit{unit21.pas}

| Lignes | The module \textit{unit21.pas} |
|--------|--------------------------------|
| 034    | SpinEdit1: TSpinEdit;          |
| 035    | ToolButton6: TToolButton;      |
| 036    | Label3: TLabel;                |
| 037    | SpinEdit2: TSpinEdit;          |
| 038    | Savesubgroupoid1: TMenuItem;  |
| 039    | StatusBar1: TStatusBar;        |
| 040    | StatusBar2: TStatusBar;        |
| 041    | ToolButton3: TToolButton;      |
| 042    | ToolButton11: TToolButton;     |
| 043    | procedure FormShow(Sender: TObject);  |
| 044    | procedure Button1Click(Sender: TObject);  |
| 045    | procedure Button2Click(Sender: TObject);  |
| 046    | procedure StringGrid1SetEditText(Sender: TObject; ACol, ARow: Integer; const Value: String);  |
| 047    | procedure StringGrid2SetEditText(Sender: TObject; ACol, ARow: Integer; const Value: String);  |
| 048    | procedure OpenFile1Click(Sender: TObject);  |
| 049    | procedure SaveFile1Click(Sender: TObject);  |
| 050    | procedure StringGrid3SelectCell(Sender: TObject; ACol, ARow: Integer; var CanSelect: Boolean);  |
| 051    | procedure New1Click(Sender: TObject);       |
| 052    | procedure ToolButton4Click(Sender: TObject);  |
| 053    | procedure Savesubgroupoid1Click(Sender: TObject);  |
| 054    | procedure ToolButton3Click(Sender: TObject);  |
| 055    | private          |
| 056    |   ForcedStop : Boolean;  |
| 057    |   err_message : String;  |
| 058    |     subgr : array[1..10000] of TSubSet;  |
| 059    |     units, SelectedSub : TSubSet;  |
| 060    |     m, n, nsub : Integer;  |
| 061    |     h : array[0..nmax, 0..nmax] of Byte;  |
| 062    |     u_left, u_right, inv : array[0..nmax] of Integer;  |
| 063    | procedure WMDropFiles(var Msg: TWMDropFiles);  message WM_DROPFILES;  |
| 064    | procedure PerformFileOpen(const FileName1 : string);  |
| 065    | procedure PerformFileSave(const FileName1 : string);  |
| 066    | procedure PerformSaveSubgroupoid(t : TSubSet; const FileName1 : string);  |
| 067    | procedure MakeUnitsTable;  |
| 068    | procedure MakeGroupoidTable;  |
| 069    | procedure MakeSubgroupoidTable(t : TSubSet);  |
| 070    | function ToStr(x : Integer) : String;  |
The module \texttt{unit21.pas} contains the following functions and procedures:

- \texttt{function SubsetToString(t : TSubSet) : String;}
- \texttt{function Cardinal(t : TSubSet) : Byte;}
- \texttt{procedure Cover(var t : TSubSet);}
- \texttt{function AlreadyFound(t : TSubSet) : Boolean;}
- \texttt{procedure AddSubgroupoid(t : TSubSet);}
- \texttt{procedure GenerateSubgroupoids(t : TSubSet; r : Byte);}
- \texttt{procedure SortByCardinal;}
- \texttt{procedure ListSubgroupoids;}
- \texttt{function IsStructure : Boolean;}
- \texttt{function IsSemigroupoid : Boolean;}
- \texttt{function IsMonoidoid : Boolean;}
- \texttt{function IsGroupoid : Boolean;}

The procedures and functions marked by the symbol "\texttt{*}" can be found in [4] or in the preprint \texttt{arXiv:math/0602604v1 [math GR]}. The other procedures and functions contained in the module \texttt{unit21.pas} are presented in the follows.

```pascal
var
i, j : Byte;
begin
DragAcceptFiles(Handle, True);
StringGrid1.EditorMode := True;
n := 0;
m := 0;
for i := 0 to nmax do
  for j := 0 to nmax do
    h[i,j]:= 0;
for i := 0 to nmax do begin
  u_left[i] := 0;
  u_right[i] := 0;
  inv[i] := 0;
  end;
  nsub := 0;
  SelectedSub := [ ]
  end;
end.
```

The procedures and functions marked by the symbol "\texttt{*}" can be find in [4] or in the preprint \texttt{arXiv:math/0602604v1 [math GR]}. The other procedures and functions contained in the module \texttt{unit21.pas} are presented in the follows.
procedure TForm1.Cover;
var
  i, j : Byte;
  modif : Boolean;
begin
repeat
  modif := false;
  for i := 1 to n do if i in t then begin
    if not (inv[i] in t) then begin
      modif := true;
      t := t + [inv[i]]
    end;
  end;
  for j := 1 to n do if j in t then
    if u_right[i] = u_left[j] then
      if not (h[i, j] in t) then begin
        modif := true;
        t := t + [h[i, j]]
      end
  end
until not modif
end;

function TForm1.Cardinal;
var
  i, nr : Byte;
begin
  nr := 0;
  for i := 1 to n do
    if i in t then
      nr := nr + 1;
  Cardinal := nr;
end;

function TForm1.SubsetToString;
var
  s : String;
  i : Byte;
begin
  s := '{';
  for i := 1 to n do
    if i in t then
      s := s + '{a' + tostr(i) + ', '; delete(s, length(s)-1, 2);
  s := s + '}';
procedure TForm1.AddSubgroupoid;
begin
  nsub := nsub + 1;
  subgr[nsub] := t;
  if nsub = 3000 then
    ForcedStop := true;
end;

function TForm1.AlreadyFound;
var
  i : Integer;
begin
  AlreadyFound := False;
  for i := 1 to nsub do
    if t = subgr[i] then
      AlreadyFound := True;
    if t = [] then
      AlreadyFound := True
end;

procedure TForm1.GenerateSubgroupoids;
var
  i : Byte;
begin
  Cover(t);
  if not AlreadyFound(t) then
    AddSubgroupoid(t);
  for i := r to n do
    if not (i in t) then
      if not ForcedStop then
        GenerateSubgroupoids(t + [i], i);
end;

procedure TForm1.SortByCardinal;
var
  i, j : Integer;
  aux : TSubSet;
begin
  for i := 1 to nsub - 1 do
    for j := i + 1 to nsub do
      if Cardinal(subgr[i]) > Cardinal(subgr[j]) then begin
        aux := subgr[i];
        subgr[i] := subgr[j];
        subgr[j] := aux;
      end;
  end;
end;
subgr[j] := aux
end
end;

procedure TForm1.ListSubgroupoids;
var
  i : Integer;
begin
  StringGrid3.RowCount := nsub;
  for i := 1 to nsub do
    StringGrid3.Cells[0, i - 1] := SubsetToString(subgr[i])
end;

procedure TForm1.PerformSaveSubgroupoid;
var
  f : TextFile;
  i, j, ordin, nunits : Byte;
  sir, ind : array[1..nmax] of Byte;
begin
  ordin := 0;
  for i := 1 to n do
    if i in SelectedSub then begin
      ordin := ordin + 1;
      sir[ordin] := i;
      ind[i] := ordin
    end;
  nunits := cardinal(SelectedSub * units);
  AssignFile(f, FileName1);
  rewrite(f);
  writeln(f, ordin);
  writeln(f, nunits);
  writeln(f);
  for i := 1 to ordin do
    write(f, ind[u2left[sir[i]]], ' ');
  writeln(f);
  for i := 1 to ordin do
    write(f, ind[u2right[sir[i]]], ' ');
  writeln(f);
  for i := 1 to ordin do
    write(f, ind[inv[sir[i]]], ' ');
  writeln(f);
  for i := 1 to ordin do begin
    for j := 1 to ordin do
      write(f, ind[h[sir[i], sir[j]]], ' ');
  writeln(f);
end;
writeln(f);
end;
CloseFile(f)
end;

procedure TForm1.MakeSubgroupoidTable;
var
  ordin, i, j : Byte;
  sir : array[1..nmax] of Byte;
begin
  ordin := 0;
  for i := 1 to n do
    if i in t then begin
      ordin := ordin + 1;
      sir[ordin] := i
    end;
  StringGrid4.RowCount := ordin + 1;
  StringGrid4.ColCount := ordin + 1;
  for i := 1 to ordin do begin
    StringGrid4.Cells[0, i] := tostr(sir[i]);
    StringGrid4.Cells[i, 0] := tostr(sir[i])
  end;
  for i := 1 to ordin do
    for j := 1 to ordin do
      if h[sir[i], sir[j]] <> 0 then
        StringGrid4.Cells[j, i] := tostr(h[sir[i], sir[j]])
      else
        StringGrid4.Cells[j, i] := ”
  end;
procedure TForm1.Savesubgroupoid1Click(Sender: TObject);
begin
  if SelectedSub <> [ ] then
    if SaveDialog1.Execute then
      PerformSaveSubgroupoid(SelectedSub, SaveDialog1.FileName)
  else
    else Application.MessageBox(‘No subgroupoid selected.’ , ”, mb_OK)
end;

procedure TForm1.StringGrid3SelectCell(Sender: TObject; ACol, ARow: Integer; var CanSelect: Boolean);
begin
  MakeSubgroupoidTable(subgr[ARow + 1]);
  SelectedSub := subgr[ARow + 1]
end:
procedure TForm1.Button1Click(Sender: TObject);
begin
  nsub := 0;
  ForcedStop := false;
  GenerateSubgroupoids([], 1);
  SortByCardinal;
  ListSubgroupoids;
  StatusBar2.SimpleText := tostr(nsub) + ' subgroupoid(s) found.'
end;

We illustrate the utilization of the program BGroidAP2 in the following cases.

Example 3.2. Determination of subgroupoids of a groupoid of type \((8;2)\).

We consider the subset \(G_{(8;2)} = \{g_j | j = 1, 8\}\) of the symmetric groupoid \(S_3\) (see [5]), where:

- \(g_1 = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}\), \(g_2 = \begin{pmatrix} 1 & 3 \\ 1 & 3 \end{pmatrix}\), \(g_3 = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}\),
- \(g_4 = \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix}\), \(g_5 = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix}\), \(g_6 = \begin{pmatrix} 1 & 3 \\ 1 & 2 \end{pmatrix}\), \(g_7 = \begin{pmatrix} 1 & 3 \\ 2 & 1 \end{pmatrix}\), \(g_8 = \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix}\).

We denote the restrictions of the structure functions \(\alpha, \beta, \iota\) and the composition law defined on the groupoid \(S_3\) to \(G_{(8;2)}\) by the same symbols. Using the correspondence

\[G_{(8;2)} = \{g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8\} \leftrightarrow \{1, 2, 3, 4, 5, 6, 7, 8\}\]

the input data are given by the following tables:

\[
\begin{array}{cccccccc}
1 & 0 & 3 & 4 & 5 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 6 & 7 & 8 \\
2 & & & & & & & \\
3 & 0 & 1 & 5 & 4 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & 1 & 3 & 5 & \\
5 & 0 & 0 & 0 & 3 & 1 & 4 & \\
6 & 0 & 7 & 2 & 8 & 0 & 0 & 0 \\
7 & 0 & 6 & 8 & 2 & 0 & 0 & 0 \\
8 & 0 & 0 & 0 & 0 & 7 & 6 & 2
\end{array}
\]

Execute the program BGroidAP2 for \(G_{(8;2)}\) and the window program of obtained results is presented in the Figure 1.

Therefore, \(G_{(8;2)}\) is a groupoid with unit set \(G_{(8;2).0} = \{g_1, g_2\}\) and it has 11 subgroupoids (see, Fig. 1). Using the initial data and input data for this groupoid, the correspondence between output data and final data is the following:
Figure 1: The subgroupoids of a \((8; 2)\) groupoid

| Output data | \(\leftrightarrow\) Final data |
|-------------|--------------------------------|
| \(\{1\}\), \(\{2\}\) | \(H^1_{(1;1)} = \{g_1\}\) |
| \(\{1, 2\}\) | \(H^2_{(1;1)} = \{g_2\}\) |
| \(\{1, 3\}\), \(\{2, 8\}\) | \(H^3_{(2;2)} = \{g_1, g_2\}\) |
| \(\{1, 2, 3\}\), \(\{1, 2, 8\}\) | \(H^4_{(2;1)} = \{g_1, g_3\}\) |
| \(\{1, 2, 3, 8\}\) | \(H^5_{(3;2)} = \{g_1, g_2, g_3\}\) |
| \(\{1, 2, 4, 6\}\) | \(H^6_{(4;2)} = \{g_1, g_2, g_3, g_8\}\) |
| \(\{1, 2, 5, 7\}\) | \(H^7_{(4;2)} = \{g_1, g_2, g_4, g_6\}\) |
| \(\{1, 2, 3, 4, 5, 6, 7, 8\}\) | \(H^8_{(8;2)} = \{g_1, g_2, g_5, g_7\}\) |
| \(\{1, 2, 3, 4, 5, 6, 7, 8\}\) | \(H^9_{(8;2)} = \{g_1, g_2, g_3, g_8\}\) |

\(\square\)
4. DETERMINATION OF WIDE SUBGROUPOIDS. THE BGroidAP3 PROGRAM

We give an algorithm for decide if the universal algebra \((G, \alpha, \beta, \mu, \nu; G_0)\) is a \(G_0\)-groupoid and for determine the all wide subgroupoids of \(G\). This algorithm is constituted by the following stages.

**Stage I.** We introduce the initial data, see Section 2.

**Stage II.** Test if the universal algebra \((G, \alpha, \beta, \mu, \nu; G_0)\) is a groupoid. This stage is composed by five steps; see, Section 2.

**Stage III.** Determine the all wide subgroupoids of \(G\). The following steps must be executed:

- **step 1.** Write all nonempty subsets \(X\) of \(G\) with property that \(G_0 \subseteq X\);
- **step 2.** Determine the subgroupoid \(\langle X \rangle\) of \(G\) generated by \(X\);
- **step 3.** Sort by cardinal all wide subgroupoids determined in the step 2;
- **step 4.** List the wide subgroupoids produced in the above step;
- **step 5.** For each wide subgroupoids make its subgroupoid table.

This algorithm is implemented on computer and we obtain the program \(BGroidAP3\), which is composed from two modules denoted by \(unit31.dfm\) and \(unit31.pas\).

The principal program of the module \(unit31.pas\) consists from the following lignes.

| Lignes  | The module unit31.pas                        |
|---------|---------------------------------------------|
| 001 - 027 | the lignes 001 - 027 of the module unit21.pas; |
| 028 - 036 | the lignes 029 - 037 of the module unit21.pas; |
| 037     | ToolButton7: TToolButton;                   |
| 038 - 043 | the lignes 038 - 043 of the module unit21.pas; |
| 044 - 051 | the lignes 045 - 052 of the module unit21.pas; |
| 052     | procedure Button7Click(Sender: TObject);    |
| 053 - 107 | the lignes 053 - 107 of the module unit21.pas; |
| 108                     | end.                                        |

The new procedure of the module \(unit31.pas\) is presented in the follows.

```pascal
procedure TForm1.ToolButton7Click(Sender: TObject);
begin
  nsub := 0;
  ForcedStop := false;
  GenerateSubgroupoids(units, m + 1);
  SortByCardinal;
  ListSubgroupoids;
  StatusBar2.SimpleText := tostr(nsub) + ' subgroupoid(s) found.'
end;
```

We illustrate the utilization of the program \(BGroidAP3\) in the following cases.
Example 4.1. Determination of wide subgroupoids of a disjoint union of two groupoids. Let the groupoid $G = K_4 \coprod F_{(4;2)}(R^2)$ and use the inputs data presented in the Example 3.1.

Execute the program $BGroidAP3$ for $G$ and the window program of obtained results is presented in the Figure 2.

Figure 2: The wide subgroupoids of a $(8;3)$—groupoid

Therefore, this groupoid has 10 wide subgroupoids (see, Fig. 2). Using the correspondence between output data and initial data, its wide subgroupoids are the following: $W_{(3;3)}^1 = G_0$, $W_{(4;3)}^2 = \{(1), Id_{Ox}, Id_{Oy}, \sigma\}$, $W_{(4;3)}^3 = \{(1), Id_{Ox}, Id_{Oy}, \tau\}$, $W_{(4;3)}^4 = \{(1), Id_{Ox}, Id_{Oy}, \sigma \circ \tau\}$, $W_{(5;3)}^5 = \{(1), Id_{Ox}, Id_{Oy}, \sigma_{Ox}, \sigma_{Oy}\}$, $W_{(6;3)}^6 = \{(1), Id_{Ox}, Id_{Oy}, \sigma, \tau, \sigma \circ \tau\}$, $W_{(6;3)}^7 = \{(1), Id_{Ox}, Id_{Oy}, \sigma_{Ox}, \sigma_{Oy}\}$, $W_{(6;3)}^8 = \{(1), Id_{Ox}, Id_{Oy}, \tau, \sigma_{Ox}, \sigma_{Oy}\}$, $W_{(6;3)}^9 = \{(1), Id_{Ox}, Id_{Oy}, \tau, \sigma_{Ox}, \sigma_{Oy}\}$, $W_{(8;3)}^{10} = K_4 \coprod F_{(4;2)}(R^2)$. 

16
Applying program BGroidAP2, we obtain that $G$ has 29 subgroupoids.

Example 4.2. Determination of wide subgroupoids of a groupoid of type $(8; 2)$. Let the groupoid $G_{(8; 2)} = \{g_j | j = 1, 8\}$ given in Example 3.2. Use the inputs data presented in the Example 2.2 and execute the program BGroidAP3. We find that this groupoid has 7 wide subgroupoids, namely:

1. $W_{(2;2)}^1 = G_{(8;2)}$,
2. $W_{(3;2)}^2 = \{g_1, g_2, g_3\}$,
3. $W_{(3;2)}^3 = \{g_1, g_2, g_8\}$,
4. $W_{(4;2)}^4 = \{g_1, g_2, g_3, g_8\}$,
5. $W_{(4;2)}^5 = \{g_1, g_2, g_4, g_6\}$,
6. $W_{(4;2)}^6 = \{g_1, g_2, g_5, g_7\}$,
7. $W_{(8;2)}^7 = G_{(8;2)}$.

5. DETERMINATION OF NORMAL SUBGROUPOIDS. THE BGroidAP4 PROGRAM

We give an algorithm for decide if the universal algebra $(G, \alpha, \beta, \mu, \iota; G_0)$ is a $G_0$-groupoid and for determine the normal subgroupoids of $G$. This algorithm is constituted by the following stages.

Stage I. We introduce the initial data, see Section 2.

Stage II. Test if the universal algebra $(G, \alpha, \beta, \mu, \iota; G_0)$ is a groupoid. This stage is composed by five steps; see, Section 2.

Stage III. Determine the normal subgroupoids of $G$. The following steps must be executed:

- **step 1.** Write all nonempty subsets $X$ of $G$ with property that $G_0 \subseteq X$;
- **step 2.** Determine the normal subgroupoid $<X>$ of $G$ generated by $X$;
- **step 3.** Sort by cardinal all normal subgroupoids determined in the step 2;
- **step 4.** List the normal subgroupoids produced in the above step;
- **step 5.** For each normal subgroupoid make its subgroupoid table.

The program BGroidAP4 is composed from two modules denoted by unit41.dfm and unit41.pas.

The principal program of the module unit41.pas consists from the following lignes.

| Lignes | The module unit41.pas |
|--------|-----------------------|
| 001 - 027 | the lignes 001 - 027 of the module unit21.pas; |
| 028 - 036 | the lignes 029 - 037 of the module unit21.pas; |
| 037 | ToolButton8: TToolButton; |
| 038 - 043 | the lignes 038 - 043 of the module unit21.pas; |
| 044 - 051 | the lignes 045 - 052 of the module unit21.pas; |
| 052 | procedure Button8Click(Sender: TObject); |
| 053 - 075 | the lignes 053 - 075 of the module unit21.pas; |
| 076 | procedure GenerateNormal(t : TSubSet; r : Byte); |
| 077 - 082 | the lignes 077 - 082 of the module unit21.pas; |
| 083 | function IsNormal(t : TSubSet) : Boolean; |
| 084 - 108 | the lignes 083 - 107 of the module unit21.pas; |
| 109 | end. |

The new procedures and the function "IsNormal" of the module unit41.pas
are presented in the follows.

```pascal
procedure TForm1.GenerateNormal;
var
  i : Byte;
begin
  Cover(t);
  if IsNormal(t) then
    if not AlreadyFound(t) then
      AddSubgroupoid(t);
  for i := r to n do
    if not (i in t) then
      if not ForcedStop then
        GenerateNormal(t + [i], i);
end;

function TForm1.IsNormal;
var
  i, j : Byte;
begin
  IsNormal := true;
  for i := 1 to n do if i in t then
    for j := 1 to n do
      if (u_right[j] = u_left[i]) and (u_right[j] = u_right[i]) then
        if not (h[h[j, i], inv[j]] in t) then begin
          IsNormal := false;
          exit;
        end;
  end;
end;

procedure TForm1/toolButton8Click(Sender: TObject);
begin
  nsub := 0;
  ForcedStop := false;
  GenerateNormal(units, m + 1);
  SortByCardinal;
  ListSubgroupoids;
  StatusBar2.SimpleText := tostr(nsub) + ' subgroupoid(s) found.'
end.
```

We illustrate the utilization of the program BGroidAP4 in the following cases.

**Example 5.1.** (i) **Determination of normal subgroupoids of a groupoid of type (9; 3).** We consider the subset $K_{(9;3)} = \{ \varphi_j | j = 1,9 \}$ of the symmetric groupoid $S_3$, where: $\varphi_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \varphi_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \varphi_3 = \begin{pmatrix} 3 \\ 3 \end{pmatrix}, \varphi_4 =$
\[
\begin{pmatrix}
1 \\
2
\end{pmatrix}, \phi_5 = \begin{pmatrix}
1 \\
3
\end{pmatrix}, \phi_6 = \begin{pmatrix}
2 \\
1
\end{pmatrix}, \phi_7 = \begin{pmatrix}
2 \\
3
\end{pmatrix}, \phi_8 = \begin{pmatrix}
3 \\
1
\end{pmatrix}, \phi_9 = \begin{pmatrix}
3 \\
2
\end{pmatrix}.
\]

We denote the restrictions of the structure functions \(\alpha, \beta, \iota\) and the composition law defined on the groupoid \(S_3\) to \(K_{(9;3)}\) by the same symbols. Using the correspondence

\[
K_{(9;3)} = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7, \phi_8, \phi_9\} \longleftrightarrow \{1, 2, 3, 4, 5, 6, 7, 8, 9\}
\]

the input data are given by the following tables:

|   | 1 | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 9 | 0 | 2 | 0 | 0 | 0 | 6 | 7 | 0 | 0 |
| 3 | 0 | 4 | 0 | 0 | 0 | 1 | 5 | 0 | 0 |
|   | 1 | 2 | 3 | 1 | 1 | 2 | 2 | 3 | 3 |
|   | 0 | 0 | 5 | 0 | 0 | 0 | 1 | 4 |   |
|   | 1 | 2 | 3 | 2 | 3 | 1 | 3 | 1 | 2 |
|   | 6 | 0 | 0 | 2 | 7 | 0 | 0 | 0 | 0 |
|   | 1 | 2 | 3 | 6 | 8 | 4 | 9 | 5 | 7 |
|   | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 6 | 2 |
|   | 8 | 0 | 0 | 9 | 3 | 0 | 0 | 0 | 0 |
|   | 0 | 9 | 0 | 0 | 0 | 8 | 3 | 0 | 0 |

Execute the program \(BGroidAP\) for \(K_{(9;3)}\) and the window program of obtained results is presented in the Figure 3.

Therefore, \(K_{(9;3)}\) is a groupoid with unit set \(K_{(9;3),0} = \{\phi_1, \phi_2, \phi_3\}\). This groupoid has 5 normal subgroupoids (see, Fig.3). Using the correspondence between output data and initial data, these normal subgroupoids are the following:

\[
N^1_{(3;3)} = K_{(9;3),0}, N^2_{(5;3)} = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_6\}, N^3_{(5;3)} = \{\phi_1, \phi_2, \phi_3, \phi_5, \phi_8\}, N^4_{(5;3)} = \{\phi_1, \phi_2, \phi_3, \phi_7, \phi_9\}, N^5_{(9;3)} = K_{(9;3)}.
\]

(ii) Applying the program \(BGroidAP2\) for the groupoid \(K_{(9;3)}\) we obtain that it has 14 subgroupoids, namely:

\[
H^1_{(1;1)} = \{\phi_1\}, H^2_{(1;1)} = \{\phi_2\}, H^3_{(1;1)} = \{\phi_3\}, H^4_{(2;2)} = \{\phi_1, \phi_2\}, H^5_{(2;2)} = \{\phi_1, \phi_3\}, H^6_{(3;3)} = \{\phi_1, \phi_2, \phi_4, \phi_6\}, H^7_{(3;3)} = \{\phi_1, \phi_2, \phi_3, \phi_5, \phi_8\}, H^8_{(4;2)} = \{\phi_1, \phi_3, \phi_4, \phi_7\}, H^9_{(4;2)} = \{\phi_1, \phi_3, \phi_5, \phi_9\}, H^{10}_{(4;2)} = \{\phi_1, \phi_2, \phi_3, \phi_7, \phi_9\}, H^{11}_{(5;3)} = N^2_{(5;3)}, H^{12}_{(5;3)} = N^3_{(5;3)}, H^{13}_{(5;3)} = N^4_{(5;3)}, H^{14}_{(9;3)} = K_{(9;3)}.
\]

(iii) Also, applying the program \(BGroidAP3\) for \(K_{(9;3)}\), we obtain that it has 5 wide subgroupoids. We observe that, each wide subgroupoid of this groupoid is a normal subgroupoid.

\[\square\]

Example 5.2. (i) Determination of normal subgroupoids of the groupoid \(G_{(8;2)}\). Use the inputs data presented in the Example 3.2 and execute the program \(BGroidAP4\). This groupoid has the following 5 normal subgroupoids:

\[
N^1_{(2;2)} = G_{(8;2),0}, N^2_{(4;2)} = \{g_1, g_2, g_3, g_8\}, N^3_{(4;2)} = \{g_1, g_2, g_4, g_6\}, N^4_{(4;2)} = \{g_1, g_2, g_5, g_7\}, N^5_{(8;2)} = G_{(8;2)}.
\]

(ii) Determination of normal subgroupoids of the groupoid \(K_4 \prod \mathcal{F}_{(4;2)}(R^2)\). Applying the program \(BGroidAP4\) for \(G = K_4 \prod \mathcal{F}_{(4;2)}(R^2)\), we obtain that \(G\) has 10 normal subgroupoids, namely:

\[
N^1_{(3;3)} = G_0, N^2_{(3;3)} = \{(1), Id_{Ox}, Id_{Oy}, \sigma\}, N^3_{(4;3)} = \{(1), Id_{Ox}, Id_{Oy}, \tau\}, N^4_{(4;3)} = \{(1), Id_{Ox}, Id_{Oy}, \sigma \circ \tau\}, N^5_{(5;3)} = \]

19
Figure 3: The normal subgroupoids of a \((9; 3)\)-groupoid

\[\{1\}, \text{Id}_{Ox}, \text{Id}_{Oy}, \sigma_{Ox}, \sigma_{Oy}\}, \ N^6_{(6; 3)} = \{1\}, \text{Id}_{Ox}, \text{Id}_{Oy}, \sigma, \sigma \circ \tau\}, \ N^7_{(6; 3)} = \{1\}, \text{Id}_{Ox}, \text{Id}_{Oy}, \tau, \sigma_{Ox}, \sigma_{Oy}\}, \ N^8_{(6; 3)} = \{1\}, \text{Id}_{Ox}, \text{Id}_{Oy}, \sigma \circ \tau, \sigma_{Ox}, \sigma_{Oy}\}, \ N^9_{(6; 3)} = G. \]

Example 5.3. Determination of subgroups and normal subgroups of a finite group. We consider the dihedral group \(D_5 = \{x_1 = e, x_2 = a, x_3 = a^2, x_4 = a^3, x_5 = a^4, x_6 = b, x_7 = ab, x_8 = a^2b, x_9 = a^3b, x_{10} = a^4b\}\) generated by the elements \(a, b\) with properties \(a^5 = e\) and \(b^2 = e.\)

The inputs data for this group are the following:
Use the above input data and execute the program BGroidAP4. Then $D_5$ has 3 normal subgroups, namely: $N_1 = \{e\}, N_2 = \{e, a, a^2, a^3, a^4\}$ and $N_3 = D_5$.

Applying the program BGroidAP2 or BGroidAP3, we obtain that $D_5$ has 8 subgroups (in fact, subgroupoids and wide subgroupoids with one unit), namely: $H_1 = \{e\}, H_2 = \{e, b\}, H_3 = \{e, ab\}, H_4 = \{e, a^2b\}, H_5 = \{e, a^3b\}, H_6 = \{e, a^4b\}, H_7 = \{e, a, a^2, a^3, a^4\}, H_8 = D_5$.

For more details concerning the programs given in this paper, the reader can be informed at e-mail address: ivan@math.uvt.ro.

REFERENCES

1. R. Brown, From Groups to Groupoids: a brief survey, Bull. London Math. Soc., 19 (1987), 113-134.
2. A. Coste, P. Dazord and A. Weinstein, Groupoides symplectiques, Publ. Dept. Math. Lyon, 2/A (1987), 1-62.
3. Gh. Ivan, Algebraic constructions of Brandt groupoids, Proceedings of the Algebra Symposium, ”Babeş-Bolyai” University, Cluj, (2002), 69-90.
4. Gh. Ivan and G. Stoianov, The program BGroidAP2 for determination of all subgroupoids of a Brandt groupoid, Analele Univ. de Vest, Timişoara, Seria Matematică-Informatică, vol. 42, fasc. 1 (2004), 93-117.
5. M. Ivan, General properties of the symmetric groupoid of a finite set, An. Univ. Craiova, Ser. Mat. Informatică, 30 (2003), no. 2, 109-119.
6. P. J. Higgins, Notes on Categories and Groupoids, Von Nostrand Reinhold, London, 1971.
7. K. Mackenzie, Lie Groupoids and Lie Algebroids in Differential Geometry, London Math. Soc., Lect. Notes Series, 124, Cambridge Univ. Press, 1987.
8. A. Weinstein, Groupoids: Unifying Internal and External Symmetries, Notices Amer. Math. Soc., 43 (1996), 744-752.
REFERENCES

1. R. Brown, From Groups to Groupoids : a brief survey, Bull. London Math. Soc., 19 (1987), 113-134.
2. A. Coste, P. Dazord and A. Weinstein, Groupoides symplectiques, Publ. Dept. Math. Lyon, 2/A (1987), 1-62.
3. Gh. Ivan, Algebraic constructions of Brandt groupoids, Proceedings of the Algebra Symposium, "Babeș-Bolyai" University, Cluj, (2002), 69-90.
4. Gh. Ivan and G. Stoianov, The program BGroidAP2 for determination of all subgroupoids of a Brandt groupoid, Analele Univ. de Vest, Timișoara, Seria Matematică- Informatică, vol. 42, fasc. 1 (2004), 93-117.
5. M. Ivan, General properties of the symmetric groupoid of a finite set, An. Univ. Craiova, Ser. Mat. Informatică, 30 (2003), no. 2, 109-119.
6. P. J. Higgins, Notes on Categories and Groupoids, Von Nostrand Reinhold, London, 1971.
7. K. Mackenzie, Lie Groupoids and Lie Algebroids in Differential Geometry, London Math. Soc., Lect. Notes Series, 124, Cambridge Univ. Press, 1987.
8. A. Weinstein, Groupoids: Unifying Internal and External Symmetries, Notices Amer. Math. Soc., 43 (1996), 744-752.