International Journal of Pure and Applied Mathematics
Volume 119 No. 1 2018, 225-234
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
doi: 10.12732/ijpam.v119i1.18

ijpam.eu

GENERATING SQL QUERIES FOR FILTERING NEAR-RINGS ON FINITE CYCLIC GROUPS

Maria Malinova[1], Angel Golev[2 §], Asen Rahnev[3]
[1,2,3]Faculty of Mathematics and Informatics, "Paisii Hilendarski"
University of Plovdiv
Plovdiv, BULGARIA

Abstract: A formal language to describe subsets from near-rings on finite cyclic groups has been created. A software module for filtering and viewing near-rings has been implemented. The module is a part of the system for generating and researching near-rings. With its help, it is much easier to group near-rings and make hypotheses about their structure.

AMS Subject Classification: 16Y30
Key Words: near-rings, generating near-rings, formal query, generating SQL query

1. Introduction

An algebraic system $(G, +, *)$ is a *(left) near-ring* on $(G, +)$, if $(G, +)$ is a group, $(G, *)$ is a semigroup and $a * (b + c) = a * b + a * c$ for $a, b, c \in G$. The left distributive law yields $x * 0 = 0$ for $x \in G$. A near-ring $(G, +, *)$ is called *zero-symmetric*, if $0 * x = 0$ holds for $x \in G$.

J. R. Clay initiated the study of near-rings, whose additive groups are finite cyclic in 1964 [1]. Some sufficient conditions for the construction of near-rings on any finite cyclic groups were obtained. In 1968 all near-rings on cyclic groups of order up to 7 were computed [2]. Later all near-rings on cyclic groups of order 8 [4], of order up to 12 [6], of order up to 13 [7], of order up to 15 [9] as well as of order up to 24 [12] and up to 29 [14] were computed.

Received: January 12, 2018
Revised: June 2, 2018
Published: June 20, 2018

§Correspondence author

We will assume G coincides with the set $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$, $2 \le n < \infty$ since every cyclic group of order n is isomorphic to the group of the remainders of modulo n. We will denote the functions mapping $\mathbb{Z}_n$ into itself by π, and the addition and the multiplication modulo n we will denote by $+$ and $\cdot$ respectively. The equality $c = a \cdot b$ will be equivalent to the congruence $ab \equiv c \pmod{n}$.

It is known that there exists a bijective correspondence between the left distributive binary operations $*$ defined on $\mathbb{Z}_n$ and the n^n functions π mapping $\mathbb{Z}_n$ into itself. If $r * 1 = b$ defines the function $\pi(r) = b$, then according to [1, Theorem II], the binary operation $*$ is left distributive exactly when, for any $x, y \in \mathbb{Z}_n$, the equality

$$\pi(x) \cdot \pi(y) = \pi(x \cdot \pi(y)) \tag{1}$$

holds.

According to the above result, obtaining of the near-rings on $\mathbb{Z}_n$ is equivalent to obtaining the functions π such that equation (1) holds.

2. Researching the Structure of Already Generated Near-Rings

We represent a near-ring on a cyclic group of order n with a list of the values of function $\pi(r)$, $r = 0, \ldots, n-1$. In order to make assumptions (hypotheses) about the structure of the near-rings, we work with numbered sequences of generated near-rings described as above [11].

Here are some near-rings on $\mathbb{Z}_9$:

(172) 0, 0, 1, 0, 0, 7, 0, 0, 4
(173) 0, 0, 1, 0, 0, 7, 3, 0, 4
(231) 0, 0, 1, 6, 0, 7, 0, 0, 4
(318) 0, 0, 4, 0, 0, 1, 0, 0, 7
(319) 0, 0, 4, 0, 0, 1, 3, 0, 7
(320) 0, 0, 4, 6, 0, 1, 0, 0, 7

The addition in a near-ring is the same as the addition in $\mathbb{Z}_n$. The table for multiplication in a near-ring is obtained by using the rule for multiplication in a near-ring: $p * q = \pi(p) \cdot q$.

For example, for the near-ring on $\mathbb{Z}_6$

(63) $0, 5, 4, 0, 2, 1,$

the tables for addition and multiplication are as follows:

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

*	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	5	4	3	2	1
2	0	4	2	0	4	2
3	0	0	0	0	0	0
4	0	2	4	0	2	4
5	0	1	2	3	4	5

For this near-ring the right distributive law fails:

$$(1+2) * 1 = \pi(3)\cdot 1 = 0 \cdot 1 = 0 \neq 3 = 5+4 = \pi(1)\cdot 1 + \pi(2)\cdot 1 = 1 * 1 + 2 * 1\,.$$

So far, a text editor has been used for our research on generated near-rings and make various assumptions before being strictly proven. But the number of near-rings for big n increases extremely fast, and this kind of work has been quite slow.

For more efficient research, all the generated near-rings were stored in a database. For each n, $3 \leq n \leq 25$, a table containing near-rings above $\mathbb{Z}_n$ is created. Especially for near-rings over $\mathbb{Z}_{25}$, only 17 886 931 near-rings are saved in the table. A huge group of near-rings above $\mathbb{Z}_{25}$ that are described with a theorem [12, Theorem 9], are not included in the table. The number of these near-rings is $5^{20} = 95\,367\,431\,640\,625$ and practically they are not generated.

The fields of each table are as follows:

- 'N' – a consecutive number of near-ring;
- n numerical fields ('E1', 'E2', ..., 'En') for each value of the function π;
- 'NOTE' – a field for text description and comments;
- 'THEOREM' – an index which points to a table containing the names of already proven and new theorems;
- 'STATUS' – boolean field to mark a near-ring.

We mark these near-rings, which are described with theorems or for which a hypothesis has been made. Several major groups of described near-rings are marked in the database in advance, such as those with all values of π: 0 or 1; those with values 0, 1 or $n-1$; and a few more fully described groups of near-rings.

	Zero-symmetric	Nonzero-symmetric	Total number
$\mathbb{Z}_3$	6	1	7
$\mathbb{Z}_4$	16	1	17
$\mathbb{Z}_5$	28	1	29
$\mathbb{Z}_6$	65	33	98
$\mathbb{Z}_7$	111	1	112
$\mathbb{Z}_8$	349	1	350
$\mathbb{Z}_9$	1 169	1	1 170
$\mathbb{Z}_{10}$	807	393	1 200
$\mathbb{Z}_{11}$	1 311	1	1 312
$\mathbb{Z}_{12}$	4 467	1 055	5 522
$\mathbb{Z}_{13}$	5 263	1	5 264
$\mathbb{Z}_{14}$	10 505	5 256	15 761
$\mathbb{Z}_{15}$	21 783	6 215	27 998
$\mathbb{Z}_{16}$	16 834 653	1	16 834 654
$\mathbb{Z}_{17}$	72 816	1	72 817
$\mathbb{Z}_{18}$	15 032 215	610 684	15 642 899
$\mathbb{Z}_{19}$	286 380	1	286 381
$\mathbb{Z}_{20}$	876 919	109 847	986 766
$\mathbb{Z}_{21}$	1 164 023	304 834	1 468 857
$\mathbb{Z}_{22}$	2 225 545	1 111 088	3 336 633
$\mathbb{Z}_{23}$	4 371 615	1	4 371 616
$\mathbb{Z}_{24}$	15 821 973	2 619 758	18 441 731
$\mathbb{Z}_{25}$	95 367 449 527 555	1	95 367 449 527 556
$\mathbb{Z}_{26}$	34 749 177	17 400 576	52 149 753
$\mathbb{Z}_{27}$	286 174 087 734	1	286 174 087 735
$\mathbb{Z}_{28}$	207 919 830	19 570 310	227 490 140
$\mathbb{Z}_{29}$	273 300 895	1	273 300 896

Table 1: Number of near-rings on $\mathbb{Z}_n$, $3 \leq n \leq 29$ [14]

The main purpose of the work is to make a tool for filtering the near-rings according to the values of the function π, and thus to facilitate their investigation.

3. A Module for Visualization and Filtering of Near-Rings

Within the software system for research of near-rings [17], we implemented a module which performs requests for filtering and visualizing of near-rings. The module features two different options for describing the desired subset of near-rings.

The first way to describe the filter is by setting the values of the function π for a specific argument using buttons and drop-down menus to set the value of the argument, relation and the value of the function π for this argument. The rule is added to the filter.

Based on the selected filter, an SQL query is generated and executed for the corresponding table.

For example, for $n = 6$ and added filters for $\pi(0) = 0$ and $\pi(3) = 3$ it will generate the following SQL query:

```
Select * from NR6 where E0 = 0 AND E3 = 3;
```

There are two check boxes 'Only without Notes' and 'Only without Status' on the screen. When selected, the condition will be added the corresponding condition in the where clause of the SQL query.

The values of the arguments can vary from 0 to $n - 1$. The relation can be $<$, $>$, $=$, $! =$ and $\in$. The values of the function π can be values from 0 to $n - 1$, sets or ranges of integers.

At any time, we can mark the selected near-rings and add a name to the theorem or hypothesis or write a brief comment. The changes are saved in the database with an appropriate request.

For example, to set the status field of a particular near-ring which is uniquely identified by the field N (for example, 101), the following code is generated

```
Update NR8 Set STATUS = 1 where N = 101
```

This feature of the module is not convenient for large n, especially when we must set values of the π function for almost all arguments. We need to set the big number of values from drop-down menus, regardless of the ability to duplicate the fields. It is also inconvenient in cases to select a subset of near-rings as follows: we want the values of π for k of chosen arguments to be exactly k values but in any order, for example: the values of $\pi(a)$, $\pi(b)$, $\pi(c)$ can be equal to a permutation of three X, Y and Z distinct values.

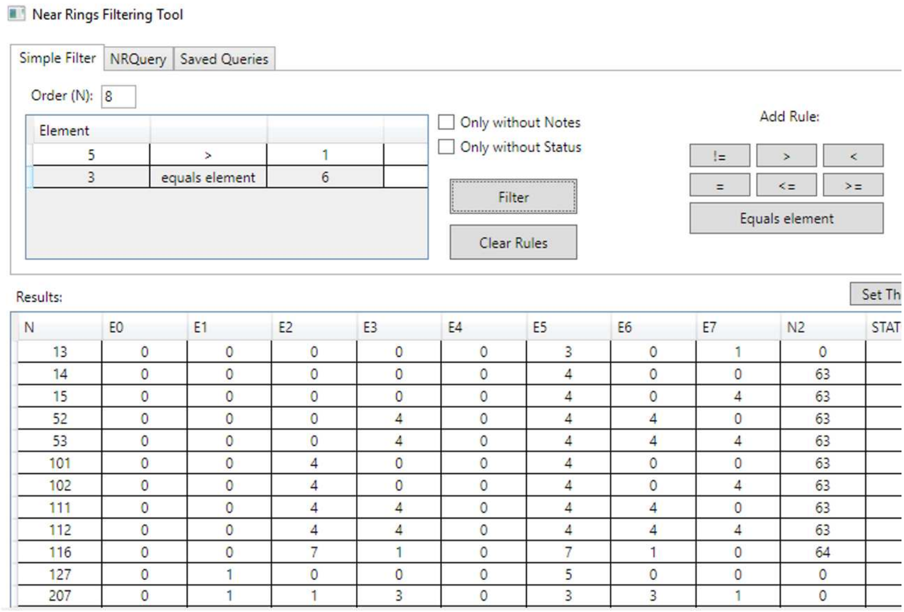

Figure 1: The module for filtering near-rings

4. Formal Queries for Filtering of Near-Rings

A tool with a formal way of description of a subset of near-rings is also implemented in the module, which will be discussed below. Using a formal description a specific subset of near-rings can be described in a short and quick way.

For example, for $n = 4$ by the formal query "`P(0)=0 and P(i)=(0,1)`", we will select zero-symmetric near-rings with value of the function for its first argument $= 0$ and values 0 or 1 for all other arguments.

The following SQL query will be generated:

```
select * from NR4
where
E0 = 0
and E1 in (0, 1)
and E2 in (0, 1)
and E3 in (0, 1);
```

If we want to describe the near-ring $\pi(1, 1, \ldots, 1)$ it is enough to write "`P(i)=1`".

Figure 2: The editor for the formal query

Description of the formal language

To define an argument of π we use P (<argument>) = <value> or P (<list of arguments or ranges>) = <value>, where the argument can be a numerical constant or a special identifier. For example: P(0)=0, P(1,3,5)=1.

To determine the argument, we can use the odd and even identifiers, which are associated with all odd and even arguments. We can use constants and simple arithmetic expressions associated with n, for example P (n/2) = n/2. These constants are calculated in advance for the corresponding n.

Finally, if we want to define the values of all other arguments, we can use the identifier i: P(0)=0 and P(n/2)=1 and P(i)=(0,3).

To set the values of the function for the corresponding argument we use constants or lists and ranges enclosed in small brackets, for example: P(0)!=0 and P(i)=(0,1,3).

When we have only and operators in the expression, they can be replaced also by ",": P(0)=0, P(N/2)=0, P(odd)=(0,1).

There is another special option for expressions of the following kind P(2,5,8) = [1,4,7], which we use to denote that the values of $\pi(2)$, $\pi(5)$, $\pi(8)$ can be equal to a permutation of 1, 4 and 7.

For $n = 9$ from the query P(0)=0, P(2,5,8)=[1,4,7], P(i)=0 the following SQL query will be generated:

```
select * from NR9
where
E0 = 0
and 2 in (1,4,7) and 5 in (1,4,7) and 8 in (1,4,7)
and (2 + 5 + 8 = (1 + 4 + 7))
and E1 = 0 and E3 = 0 and E6 = 0 and E7 = 0;
```

The result from the query will be:

(172) 0, 0, 1, 0, 0, 7, 0, 0, 4

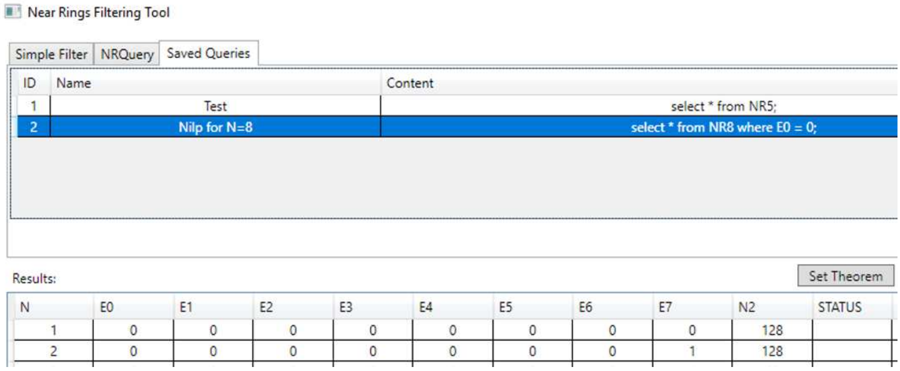

Figure 3: The tool for saving queries

(318) 0, 0, 4, 0, 0, 1, 0, 0, 7
(402) 0, 0, 7, 0, 0, 4, 0, 0, 1

If we want to write a formal query for near-rings on $\mathbb{Z}_9$ corresponding to [12, Theorem 9] (0, 3 and 6 are the nilpotents of second degree on $\mathbb{Z}_9$), it will look like this: `P(0,3,6)=0, P(i)=(0,3,6)`. If we have a identifier `nilpotents` which is set with the list of all nilpotents of second degree on $\mathbb{Z}_n$, the formal request will look like this: `P(nilpotents)=0, P(i)=(nilpotents)`. The generated SQL query will be:

```
select * from NR9
where
0 = 0 and 3 = 0 and 6 = 0
and 1 in (0,3,6) and 2 in (0,3,6)
and 4 in (0,3,6) and 5 in (0,3,6)
and 7 in (0,3,6) and E8 in (0,3,6)
```

In this case, the identifier `i` will be associated with all arguments except for 0, 3 and 6. The number of near-rings matching this query is $3^6 = 729$, including the zero near-ring.

The program module has the possibility to record a subset into a file or into a new table in the database. Generated SQL queries can be written to the database in a special query table. The program makes it possible to load these requests and execute them.

Generation of SQL query from the formal description

The entered formal query is parsing into its basic elements. The next step is to find the constant identifiers and the ranges. The non-numeric constants are replaced by their predefined values and the expressions are calculated. We use an array to mark the arguments already used. If we have the `odd` or `even` identifiers, they are replaced with $0, 2, \ldots, n-2$ or $1, 3, \ldots, n-1$, excluding arguments already used. Finally, if we have a identifier `i`, condition for all unmarked arguments are generated.

The program is written on C#, and uses the Microsoft SQL Server.

5. Conclusion

A formal language to describe subsets from near-rings on finite cyclic groups has been created. A software module for filtering and viewing near-rings has been implemented. The queries can be constructed with drop-down menus or written in the formal language. The module is a part of the system for generating and researching near-rings. With its help, it is much easier to group near-rings and make hypotheses about their structure.

Acknowledgements

This work was partially supported by the project FP17-FMI-008 of the Scientific Fund of the University of Plovdiv Paisii Hilendarski, Bulgaria.

References

[1] Clay J. R., The near-rings on a finite cycle group, *Amer. Math. Monthly*, **71** (1964), 47–50.

[2] Clay J. R., The near-rings on groups of low order, *Math. Zeitschr.*, **104** (1968), 364–371.

[3] Jacobson R. A., The structure of near-rings on a group of prime order, *Amer. Math. Monthly*, **73** (1966), 59–61.

[4] Pilz G., *Near-rings*, North-Holland, Amst., **23** (1977).

[5] Pilz G., *Near-rings*, North-Holland, Amst., Revised edition, **23** (1983).

[6] Yerby R., H. Heatherly, Near-Ring Newsletter, **7** (1984), 14–22.

[7] Rakhnev A. K., G. A. Daskalov, Construction of near-rings on finite cyclic groups, *Math. and Math. Education*, Sunny Beach, Bulgaria, (1985), 280-288.

[8] Rakhnev A. K., On near-rings, whose additive groups are finite cyclics, *Compt. rend. Acad. bulg. Sci.*, **39** No. 5 (1986), 13–14.

[9] Aichinger E., F. Binder, J. Ecker, R. Eggetsberger, P. Mayr and C. Nöbauer. SONATA: Systems Of Nearrings And Their Applications, Package for the group theory system GAP4. Johanes Kepler University Linz, Austria, (2008). http://www.algebra.uni-linz.ac.at/sonata/

[10] Rahneva O., A. Golev, N. Pavlov, Dynamic Generation of Testing Question in SQL in DeTC, *BAS, Cybernetics and Information Technologies*, **8**, No1 Sofia 2008, 73–81.

[11] Rahnev A., A. Golev, Some New Lower Bounds for the Number of Near-rings on Finite Cyclic Groups, *Int. Journal of Pure and Applied Mathematics*, **59**, No.1 (2010), 59–75.

[12] Rahnev A. K., A. A. Golev, Computing Near-rings on Finite Cyclic Groups, *Compt. rend. Acad. bulg. Sci.*, **63**, book 5 (2010), 645–650.

[13] Golev A., A. Rahnev, Computing Classes of Isomorphic Near-rings on Cyclic Groups of Order up to 23, *Scientific Works, Plovdiv University*, **37**, book 3, Mathematics (2010), 53–66.

[14] Golev A. A., A. K. Rahnev, Computing Near-rings on Finite Cyclic Groups of Order up to 29, *Compt. rend. Acad. bulg. Sci.*, **64**, No. 4, (2011), 461-468.

[15] Golev A., Algorithms for Generating Near-rings on Finite Cyclic Groups, *Proceedings of the Anniversary International Conference REMIA2010, Plovdiv*, 255–262, 10-12 December 2010.

[16] Golev A., A. Rahnev, New results for near-rings on finite cyclic groups, *Proceedings of Annual Workshop Coding Theory and Applications*, 51–54, Gabrovo, December 2011.

[17] Pavlov N., A. Golev, A. Rahnev, Distributed Software system for Testing Near-RIngs Hypotheses and New Constructions for Near-Rings on Finite Cyclic Groups, *Int. Journal of Pure and Applied Mathematics*, **90**, No.3 (2014), 345–356.