Document details

Procedia Computer Science
Volume 59, 2015, Pages 540-549
1st International Conference on Computer Science and Computational Intelligence, ICCSCI 2015;
Bina Nusantara UniversityAnggrek CampusJakarta; Indonesia; 24 August 2015 through 26 August 2015

Testing Division Rings and Fields Using a Computer Program

(Conference Paper) (Open Access)

Aditya, R., Zulfikar, M.T., Manik, N.I.

Department of Mathematics and Statistics, Bina Nusantara University, KH Syahdan Street 9, Palmerah, West-Jakarta, 11480, Indonesia

Abstract

View references (4)

Algebra is a branch of mathematics which is studying algebraic structures of sets with respect to some operations on them. Each structure has its own axioms and properties. In some cases, to test what kind of structure is a given set and operations is difficult to be done manually. To help this, an application program to test algebraic structures is developed. In this article, we focus on two structures: division ring and fields , and an application program to test them is created by using Java, an open-source based programming language. This application program provides testing of various input of finite set such as integers, matrices and alphabets. By this application, testing of algebraic structures can be done faster than the manual one and its results are accurate. In this article, we focus on testing division rings and fields , together with some examples. © 2015 The Authors. Published by Elsevier B.V.

SciVal Topic Prominence

Topic: Structural Operational Semantics | Labeled Transition System | Process Algebra

Prominence percentile: 39.792

Author keywords

Abstract Algebra | Computational Algebra | Division Rings | Fields

Indexed keywords

Engineering controlled terms: Abstracting | Algebra | Application programs | Artificial intelligence | Computer software | Integer programming | Java programming language | Open source software

Engineering uncontrolled terms: Abstract algebra | Algebraic structures | Computational algebra | Fields | Finite set | Open sources

Engineering main heading: Software testing

Related documents

A validity argument for an undergraduate mathematics concept inventory
Melhuish, K. , Hicks, M.D.
(2019) Quantitative Measures of Mathematical Knowledge: Researching Instruments and Perspectives

Three conceptual replication studies in group theory
Melhuish, K.
(2018) Journal for Research in Mathematics Education

Study on ring: G (x)-nil-clean and strongly g (x)-clean, but not strongly g (x)-nil-clean
Sulandra, I.M. , Mulyasari, I.
(2020) AIP Conference Proceedings

ISSN: 18770509
Source Type: Conference Proceeding
Original language: English

DOI: 10.1016/j.procs.2015.07.537
Document Type: Conference Paper
Volume Editors: Budiharto W.
Sponsors:
Publisher: Elsevier

References (4)

1 Gilbert, L., Gilbert, J.
(2009) *Elements of Modern Algebra*. Cited 15 times.
7th ed, Belmont: Brooks/Cole

2 Fraleigh, J.B.
(2003) *A First Course in Abstract Algebra*. Cited 330 times.
7 th ed. Boston: Addison-Wesley

3 Manik, N.I., Fortunatadewi, F., Tasman, D.
Testing of Rings and Fields Using a Computer Program
Journal of Software, 9 (5), pp. 1140-1150.

4 Pressman, R.S.
(2010) *Software Engineering: A Practitioners Approach*. Cited 4766 times.
New York: McGraw

Procedia Computer Science

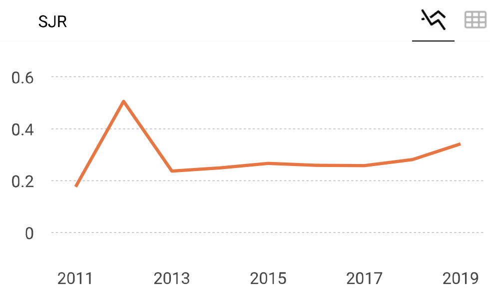
COUNTRY	SUBJECT AREA AND CATEGORY	PUBLISHER	H-INDEX
Netherlands Universities and research institutions in Netherlands	Computer Science Computer Science (miscellaneous)	Elsevier BV	59

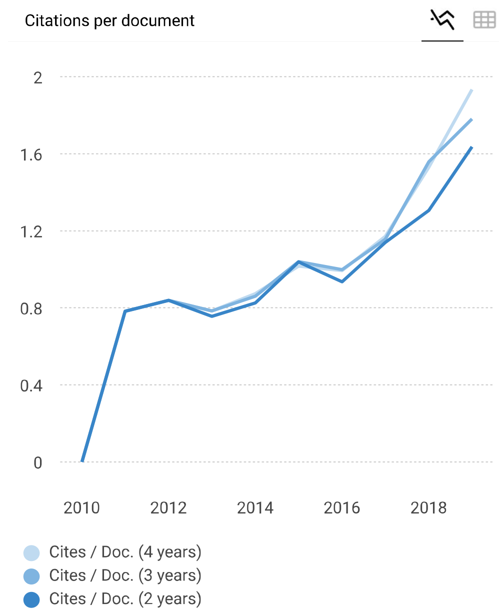
PUBLICATION TYPE	ISSN	COVERAGE	INFORMATION
Conferences and Proceedings	18770509	2010-2019	Homepage How to publish in this journal

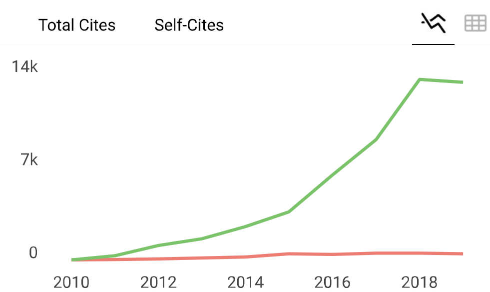
SCOPE

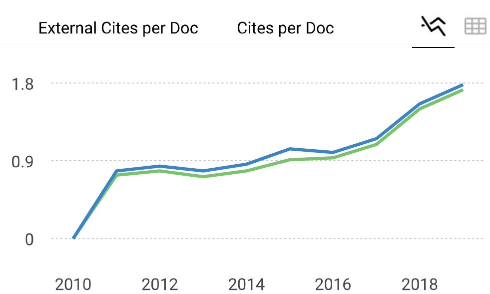
Launched in 2009, Procedia Computer Science is an electronic product focusing entirely on publishing high quality conference proceedings. Procedia Computer Science enables fast dissemination so conference delegates can publish their papers in a dedicated online issue on ScienceDirect, which is then made freely available worldwide.

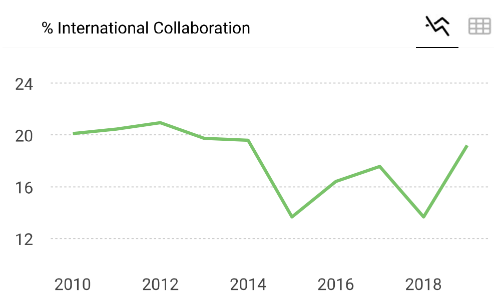
Join the conversation about this journal

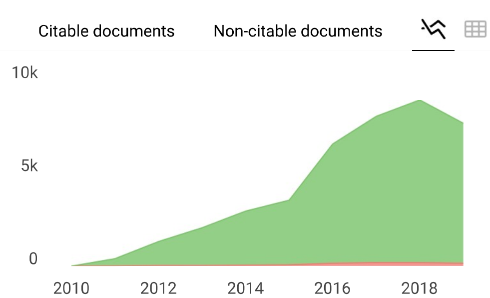


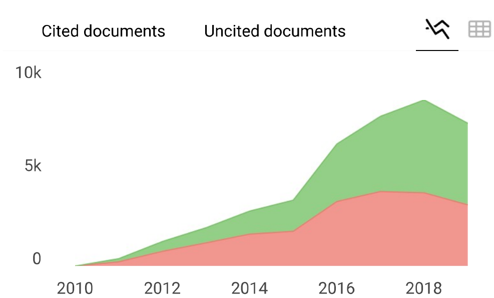


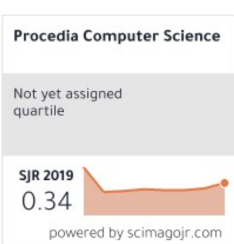












← Show this widget in your own website

Just copy the code below and paste within your html code:

<a href="https://www.scim

Metrics based on Scopus® data as of April 2020

H **Hadi Farahani** 4 months ago

ScienceDirect

Procedia Computer Science

Open access

3.0

CiteScore

Guide for authors

Menu

Search in this journal

International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

Edited by Widodo Budiharto

Volume 59,

Pages 1-584 (2015)

Download full issue

Previous vol/issue

Next vol/issue

Receive an update when the latest issues in this journal are published

Sign in to set up alerts

FEEDBACK

Procedia Computer Science

Open access

Search in this journal

About the journal

Aims and scope Editorial board Abstracting and indexing

Procedia Computer Science

Open access

Procedia Computer Science

Open access

Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 59 (2015) 1
Procedia Computer Science

International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

Preface

The development of computer science, computational intelligence, information and technology knowledge grows rapidly nowadays. The researchers, engineers and scientists in this fields need a media to share their knowledge and idea and to expand the networking. The 1st International Conference on Computer Science and Computational Intelligence (ICCSCI 2015) is an international forum for researchers, engineers and scientists to present their knowledge of technological advances and research in the fields of Computer Science, and Computational Intelligence information and technology.
Two related tracks will be held, covering both applications and current research work in the area of Computer Science, Bio-Informatics, Computer Games, Computer Vision, Information Engineering, Robotics and Automation, Internet of Things and many other things closely related to Computer Science. In the area of Computational Intelligence, Fuzzy System, Machine Learning, Natural Language Processing, Neural Networks, Pattern Recognition and many other things closely related to Computational Intelligence are covered.

We would like to thank to all participants, keynote speakers, reviewers and committee for the contributions to the conference program and proceeding. We would like to express our appreciation to the reviewers for the valuable review and suggestion, so that we can maintain the quality of this proceeding very well. We would like to thank Elsevier for supporting publication of this conference proceeding.

Guest Editors:
Dr. Widodo Budiharto
Email: wbudiharto@binus.edu
Lili Ayu Wulandhari, PhD
Email: lwulandhari@binus.edu

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)
doi:10.1016/j.procs.2015.07.573

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science

Procedia Computer Science 59 (2015) iii–v

Contents

doi:10.1016/S1877-0509(15)02221-8

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 59 (2015) 540 – 549

Procedia Computer Science

International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

Testing Division Rings and Fields Using a Computer Program

Ricky Aditya[1], Muhammad Taufiq Zulfikar, Ngarap Imanuel Manik[1]

Department of Mathematics and Statistics, Bina Nusantara University, KH Syahdan Street 9, Palmerah, West Jakarta 11480
[b]Second affiliation, Address, City and Postcode, Country

Abstract

Abstract algebra is a branch of mathematics which is studying algebraic structures of sets with respect to some operations on them. Each structure has its own axioms and properties. In some cases, to test what kind of structure is a given set and operations is difficult to be done manually. To help this, an application program to test algebraic structures is developed. In this article, we focus on two structures : division ring and fields, and an application program to test them is created by using Java, an open-source based programming language. This application program provides testing of various input of finite set such as integers, matrices and alphabets. By this application, testing of algebraic structures can be done faster than the manual one and its results are accurate. In this article, we focus on testing division rings and fields, together with some examples.

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

Keywords:Abstract Algebra; Division Rings; Fields; Computational Algebra

1. Introduction

Abstract algebra, sometimes also called as modern algebra, is a branch of mathematics that is studying algebraic structures, such as groups, rings and fields[1,2]. Algebraic structures generally refers to a set with one or more finitary operations defined on it. An algebraic structure is characterized by properties of its operations.

There are many algebraic structures that are studied in abstract algebra, but now we will focus on ring theory. In ring theory, there are ring, commutative ring, integral domain, division ring, field, etc[1,2]. Each structure has its own definition, axioms and characterizations, and there are also relations among them. A set with respect to some operations might also be more than one structures.

Previously, an application program to test rings and fields has been developed[3]. The application is able to test ring, commutative ring, division ring, and field. In that application, we input the set by inputing its elements, and

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)
doi:10.1016/j.procs.2015.07.537

also define the operations on that set. After that, the program will analyze the structure of the inputed set. As an output, it is shown what properties hold in the set, and what kind of algebraic structures is the set.

Based on that application, in this paper, the authors delevop it by adding more algebraic structures to test and more kinds of input set. In the developed application, the input is added by matrix and alphabet input. For alphabet input, user can also input the Cayley table manually. In the previous work, the input of the set is limited to integers modulo n and the operation must be defined explicitly in a formula. By this addition, some various examples of algebraic structures can be tested. We can also obtain various results of testing in this application program and the difference among some algebraic structures can be shown clearer.

2. Basic Concepts in Abstract Algebra : Division Rings and Fields

In this section we will discuss some basic concepts and theories about abstract algebra. Specifically, we discuss about division ring and field. We must discuss about ring first, because division ring and field are derived class of ring. These concepts have been given in[1,2].

2.1 Rings

A ring is an algebraic structure of a set that consisting two binary operations: addition and multiplication, in which the set is an Abelian group with respect to the addition, a semigroup with respect to the multiplication and the multiplication is distributive to the addition. In other words, a set R with two binary operations: addition ($+$) and multiplication ($\cdot$) is a **ring** if these following conditions hold for any $a,b,c \in R$:

a. $a+b \in R$ and $a \cdot b \in R$ (closed property of addition and multiplication)
b. $(a+b)+c = a+(b+c)$ and $(a \cdot b)\cdot c = a\cdot(b \cdot c)$ (associative property of addition and multiplication)
c. There exists $0 \in R$ such that $0+a = a+0 = a$, for any $a \in R$ (existence of the identity element of addition).
d. There exists $(-a) \in R$ such that $a+(-a) = (-a)+a = 0$ (existence of additional inverse)
e. $a+b = b+a$ (commutative property of addition)
f. $a\cdot(b+c) = a\cdot b + a \cdot c$ and $(a+b)\cdot c = a\cdot c + b\cdot c$ (distributive property)

Some examples of rings are the set of integers (**Z**) with respect to usual addition and multiplication, the set of all $n \times n$ real matrices with respect to matrix addition and multiplication, and the set of integers modulo n (Z_n) with respect to addition and multiplication modulo n. By adding some conditions, some special classes of rings can be defined. This will be discussed in the next sub-sections.

2.2. Division Rings and Fields

A ring is an Abelian (commutative) group with respect to addition, because it satisfies all axioms of addition group and its addition operation is commutative. A ring is not always a multiplicative group. A ring is just a semigroup with respect to multiplication, i.e. it only satisfies closed and associative property. Multiplication in a ring is not always commutative, as multiplication in ring of all $n \times n$ real matrices is not commutative. Some rings might have identity element of multiplication (also called by unity), usually denoted by 1, such that $1\cdot a = a \cdot 1 = a$, for any $a \in R$, but some others might not. As an example **Z** has 1 as its multiplication identity, but the ring of even integers does not have any multiplicative identity. From this fact, two new classes of rings are defined:

a. A ring is said to be a **commutative ring** if its multiplication is commutative, i.e. $a \cdot b = b \cdot a$, for any $a,b \in R$.
b. A ring is said to be a **ring with unity** if it has identity element of multiplication.

Examples of commutative rings and rings with unity are **Z** and $\mathbf{Z_n}$. The set of all $n \times n$ real matrices is a ring with unity (the identity matrix), but not a commutative ring.

Another axiom of multiplicative group that does not always hold in a ring is that every element has multiplicative inverse. This is because even the multiplicative unity of a ring does not always exists. However, in some ring with unity, we can find that any nonzero element has multiplicative inverse, as in the ring of all rational numbers **Q** and ring of all real number **R**, and in some others not, as in the ring of all integer **Z**. In other side, existence of multiplicative inverse has no correlation with the commutativity of the multiplication. Therefore we can define two more classes of rings:

a. A ring is said to be a **division ring** if it is a ring with unity (1) and any its nonzero element has multiplicative inverse, i.e. if $a \in R$ and $a \neq 0$, then it exists $a^{-1} \in R$ such that $a \cdot a^{-1} = a^{-1} \cdot a = 1$.

b. A ring is said to be a **field** if it is a division ring and a commutative ring.

Examples of fields (and also be examples of division rings) is the ring of all rational numbers **Z** and ring of all real numbers □ with respect to usual addition and multiplication. The set of all integers **Z** is not a division ring, and so not a field. It is a little bit difficult to find an example of a division ring that is not a field, but it will be showed later.

3. Methods

In designing the application program, the Waterfall method model is used[3], with the stages as follows[4]:

A. Design of Screen Display

There are four screen displays made on the stage is designing the application program of algebraic structure testing. The draft of the screen display design is as follows.

B. Design of Prologue / Opening Screen Display

This is what users see when the program running. The prologue display contains program title, user's identity, supervising lecturer's identity, and Jbutton. Jbutton is useful to close the prologue display and open the main display.

C. Design of Algebraic Structure Testing Display

The display provides users to perform ring, commutative ring, integral domain, division ring and field testing. On the screen, there are three main sub-tabs: Data Input, Analysis of Cayley Table and Results Analysis. Data Input sub-tab is used to input the elements of tested set and fill the Cayley table. Analysis of Cayley Table sub-tab allows the user to see the testing result of the Cayley table. Result Analysis sub-tab shows conclusions of Cayley table testing results, that is what kind of structures is the tested set.

D. Module Design (Pseudocode)

In its development, the application program was built by forming the program modules. The are many modules in this application program as we need to check many axioms. Some of them are shown in this paper. Actually we use more modules.

Module ClosedOperation (for checking closed property)

```
Begin
        For any cell, repeat
        Begin
                If there is element of a cell that is not element of the set
                Begin
                        flag = false
                End
                Otherwise
                Begin
                        flag= true
                End
        End
End
```

Module AssociativeOperation (for checking associative property)

```
Begin
        Count=0
        From i=1, to i=number of elements, repeat
        Begin
                From j=1 to j=number of elements, repeat
                Begin
                        From k=1 to k=number of elements, repeat
                        Begin
                                temp = j-th row, k-th column element
                                location = column position temp
                                left = i-th row, location-th column element
                                temp = i-th row, j-th column element
                                location = row position temp
                                right = location-th row, k-th column element
                                If left=right
                                Begin
                                        Count=count+1
                                End
                        End
                  End
        End
        If count = number of elements of the set
        Begin
                flag=true
        End
        Otherwise
        Begin
                    flag=false
        End
        End
```

Modul CommutativeOperation (for checking commutative property)

```
Begin
        Count=0
        From i=1, to i=number of elements, repeat
        Begin
                From j=1, to j=number of elements, repeat
                Begin
                        Left = i-th row, j-th column element
                        Right = j-th row, i-th column element
                        If left=right
                        Begin
                                Count = count+1
                        End
                End
        End
        If count = number of elements of the set
        Begin
                flag=true
        End
        Otherwise
        Begin
```

```
                flag =false
        End
End
```

Module UnityOperation (for checking existence of unity)

```
Begin
        flag =false
        From i=1, to i=number of elements, repeat
        Begin
                If row ke-i = row header dan column ke-i = column header
                Begin
                        unity = i-th element of the set
                        flag = true
                End
        End
End
```

Modul InverseOperation (for checking inverse existence property)

```
Begin
        If UnityOperation = true
        Begin
                Count=0
                From i=1, to i=number of elements, repeat
                Begin
                        From j=1, to j=number of elements, repeat
                        Begin
                                If i-th row, j-th column element = unity
                                Begin
                                        Inverse of i-th element = j-th element
                                        Count=count+1
                                End
                        End
                End
                If count=number of elements of the set
                Begin
                        flag=true
                End
        End
End
```

Module CayleyAnalysis (for Cayley table analysis)

```
Begin
        If ClosedOperation = true, AssociativeOperation = true, CommutativeOperation = true, UnityOperation =
true, InverseOperation = true dan Distributive=true with respect to (+)
        Begin
                Conclusion = RING
                If ClosedOperation = true, AssociativeOperation = true, CommutativeOperation = true with
                respect to (*)
                Conclusion = COMMUTATIVE RING
        Otherwise,
        Begin
                Conclusion = Not COMMUTATIVE RING
        End
```

```
        If UnityOperation = true, InverseOperation = true with respect to (*)
        Begin
                Conclusion=DIVISION RING
        End
        Otherwise,
        Begin
                Conclusion = Not DIVISION RING
        End
        If UnityOperation = true, InverseOperation = true , CommutativeOperation = true with respect to (*)
        Begin
                Conclusion = FIELD
        End
        Otherwise,
        Begin
                Conclusion = Not FIELD
        End
        End
        Otherwise,
        Begin
                Conclusion = Not RING
        End
End
```

Based on that design method, an application program to test algebraic structures has been developed. We will see the result of this program in the next section.

4. Some Testing Results

In this section, we will see some examples of algebraic structure testings using a computer program described in previous section. The program is a development from previous work[3] so that its basic design and specifications are same. The first step in running this program is inputing the set that will be tested together with defining some operations, in this case addition and multiplication, on it. After that, by clicking Jbutton "Analysis", we will obtain the analysis of Cayley table of our inputed set and the program will show what operation properties are satisfied in our set. Then in sub-tab "Result Analysis", it is shown what kind of structures is our set.

We will see those processes in two examples: a set of ternary matrices:

$$\left\{\begin{bmatrix}0 & 0\\0 & 0\end{bmatrix},\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix},\begin{bmatrix}2 & 2\\2 & 2\end{bmatrix}\right\}$$

with respect to matrix addition and multiplication modulo 3, and a set of alphabet $\{x,a,b,c,d,e,f\}$ with designed Cayley table.

4.1 Testing set of matrices

First the user input the set $\left\{\begin{bmatrix}0 & 0\\0 & 0\end{bmatrix},\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix},\begin{bmatrix}2 & 2\\2 & 2\end{bmatrix}\right\}$ and define matrix addition and multiplication modulo 3:

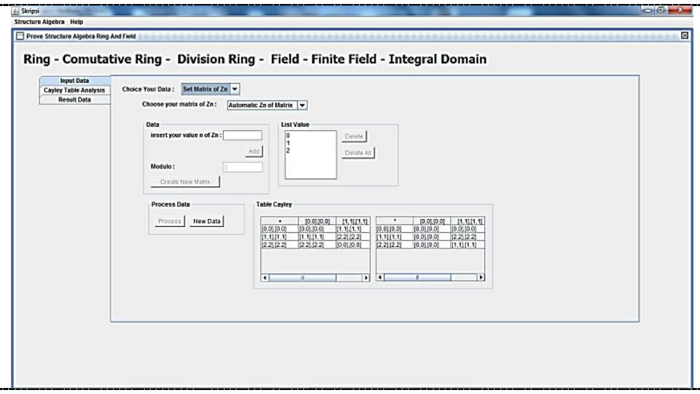


Figure 1. Input set of matrices by user.

Then we execute the Cayley table analysis:

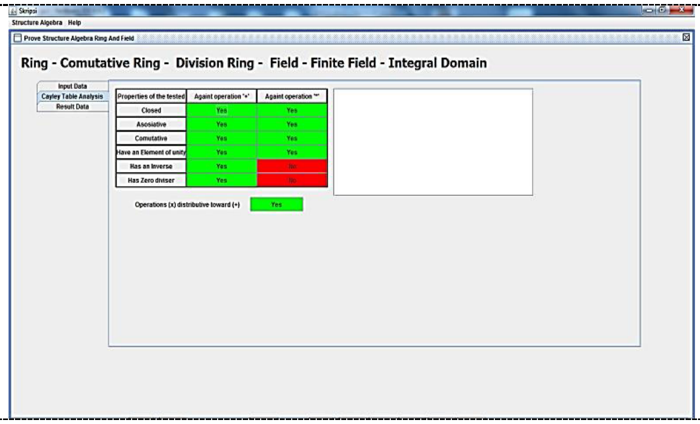


Figure 2. Cayley table analysis of set of matrices.

As the results, the program shows that that set of matrices is a ring and a commutative ring, but not a division ring and not a field.

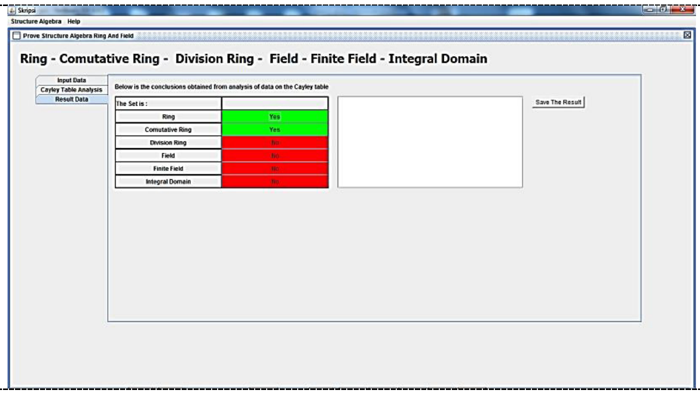


Figure 3. Conclusion of set of matrices.

4.2 Testing set of alphabet with designed Cayley table

In this case, we see the testing of a set of alphabets $\{x,a,b,c,d,e,f\}$ with designed Cayley table. Cayley tables of addition and multiplication are given below:

+	x	a	b	c	d	e	f
x	x	a	b	c	d	e	f
a	a	b	c	d	e	f	x
b	b	c	d	e	f	x	a
C	c	d	e	f	x	a	b
d	d	e	f	x	a	b	c
E	e	f	x	a	b	c	d
f	f	x	a	b	c	d	e

+	x	a	b	c	d	E	F
x	x	x	x	x	x	X	x
a	x	a	b	c	d	E	f
b	x	b	c	a	e	F	d
c	x	c	a	b	f	D	e
d	x	d	f	e	a	C	b
e	x	e	d	f	b	A	c
f	x	f	e	d	c	B	a

The user input the set of alphabet and the designed Cayley table:

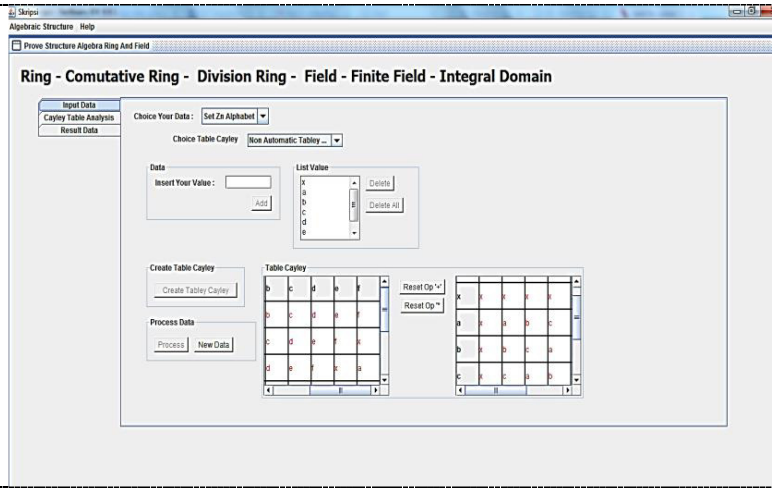


Figure 4. Input set of alphabet with designed Cayley table.

Then we execute the Cayley table analysis:

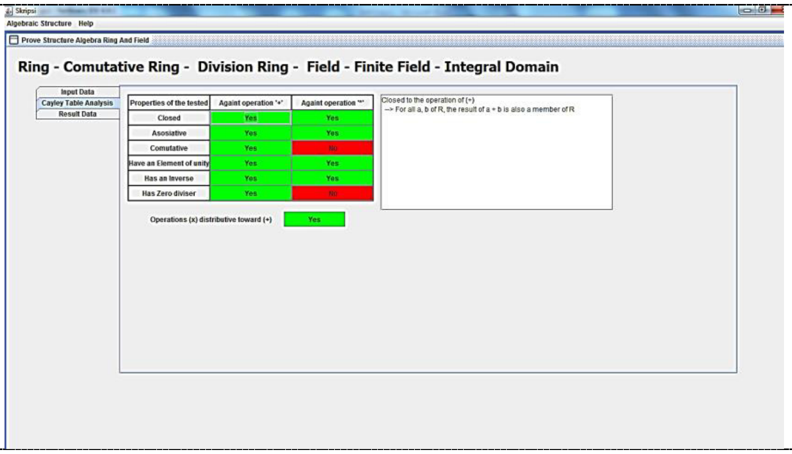


Figure 5. Cayley table analysis of alphabet with designed Cayley table.

As the results, the program shows that our inputed set is a ring and a division ring, but not a commutative ring and so not a field.

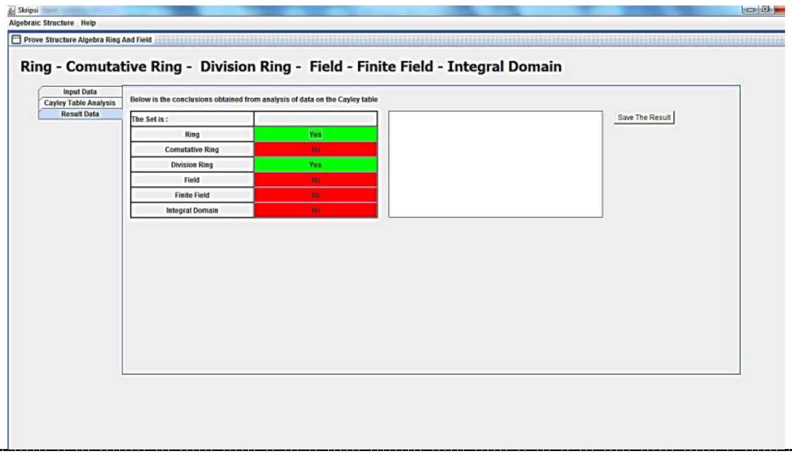


Figure 6. Conclusion of set of alphabet with designed Cayley table.

Why we need to test that set? Because if we test **Z**, we will have that its addition and multiplication are always commutative. Thus any tested set will result in commutative ring. To give an example of a set that is a ring but not commutative, we need to give different kind of input. Therefore, we need to consider matrix multiplication, that is not commutative. However, to create a non-commutative division ring, it is difficult if we use set of matrices because not all nonzero matrix has multiplicative inverse. So, to provide a non-commutative division ring, we need to define the addition and multiplication by defining their Cayley tables manually.

5. Conclusion

From previous section, we have seen that the application program is worked properly for some various examples. It gave same results as manual testing based on modern algebra concepts. In checking associative and commutative properties, it spend less time than manual testing, especially for testing set with many elements. Comparing with previous program, the developed program has some advantages such as:

1. There are some new algebraic structure added, such as division ring.

2. Input of the application program are added by matrix and alphabet input, so it has more variation of set to be tested. The Cayley table can also be inputed manually.
3. By more various input set, we can test some non-commutative structures, which are not able to be tested if the input is limited to integer modulo n.

However the application program also has disadvantages such as:

1. The input set is still limited to finite set. There are some structures of infinite set with well-defined operations and this program cannot test them. For example, the set of integers **Z** with usual addition and multiplication. This program still cannot test it.
2. For alphabet input with designed Cayley table which has many elements, the user will spend more time to input the Cayley table and it can be ineffective.
3. Some algorithms that are used in this program are brute-force type. Therefore, for testing set with too many elements, it might spend many time for running the application.

This application program can still be developed by adding more algebraic structures to test and more variation of input set. So far, this application program has worked well and its results are accurate.

References

1. Gilbert L and Gilbert J. *Elements of Modern Algebra*, 7th ed, Belmont: Brooks/Cole, 2009
2. Fraleigh JB. *A First Course in Abstract Algebra*, 7 th ed. Boston: Addison-Wesley, 2003.
3. Manik NI, Fortunatadewi F, and Tasman D. Testing of Rings and Fields Using a Computer Program. Journal of Software Vol.9 No.5:1140-1150.
4. Pressman RS. Software Engineering: A Practitioners Approach. New York: McGraw, 2010.

The 1st International Conference on Computer Science and Computational Intelligence 2015

Certificate of Oral Presentation

Presented to

Ricky. A

(Binus University,Indonesia)

Paper ID : ICCSCI93

Paper Title : Testing Division Rings and Fields Using a Computer Program

For successfully presenting his/her paper in International Conference on Computer Science and Computational Intelligence 2015, during August 24-26, 2015, in Indonesia.

ICCSCI
1st International Conference on Computer Science and Computational Intelligence

Dr. Widodo Budiharto, S.Si., M.Kom.
General Chair

Fredy Purnomo, S.Kom., M. Kom.
Dean of School of Computer Science

BINUS UNIVERSITY