Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Ricky Aditya
Assignment title:	Periksa similarity
Submission title:	Testing Division Rings and Fields Using a Computer Program
File name:	Testing_Division_Rings_and_Fields_Using_a_Computer_Progr...
File size:	791.54K
Page count:	10
Word count:	2,992
Character count:	15,156
Submission date:	01-Jul-2022 10:27AM (UTC+0700)
Submission ID:	1865274229

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

Testing Division Rings and Fields Using a Computer Program

by Aditya Ricky

Submission date: 01-Jul-2022 10:27AM (UTC+0700)
Submission ID: 1865274229
File name: Testing_Division_Rings_and_Fields_Using_a_Computer_Program.pdf (791.54K)
Word count: 2992
Character count: 15156

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:

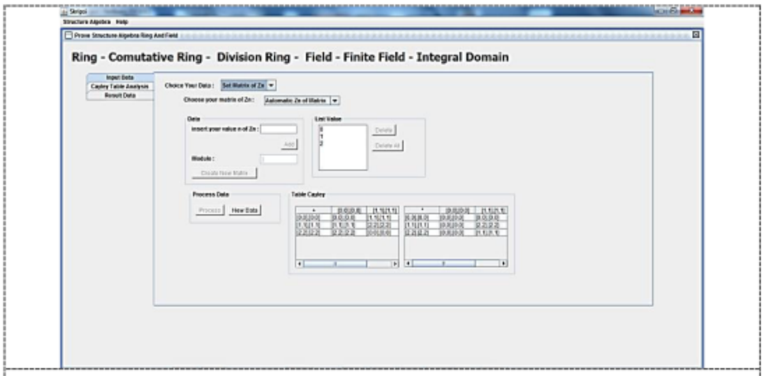

Figure 1. Input set of matrices by user.

Then we execute the Cayley table analysis:

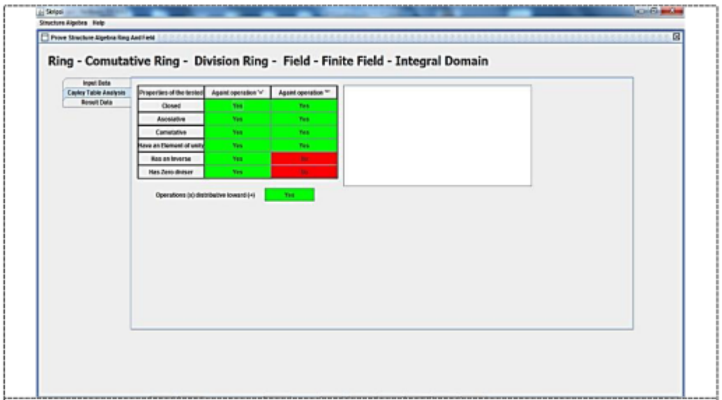

Figure 2. Cayley table analysis of set of matrices.

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.

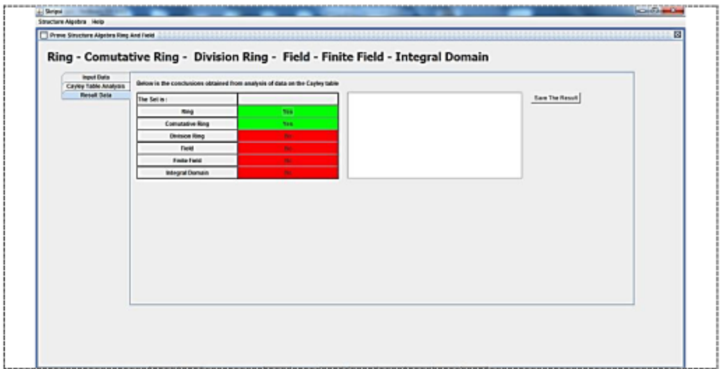

Figure 3. Conclusion of set of matrices.

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	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:

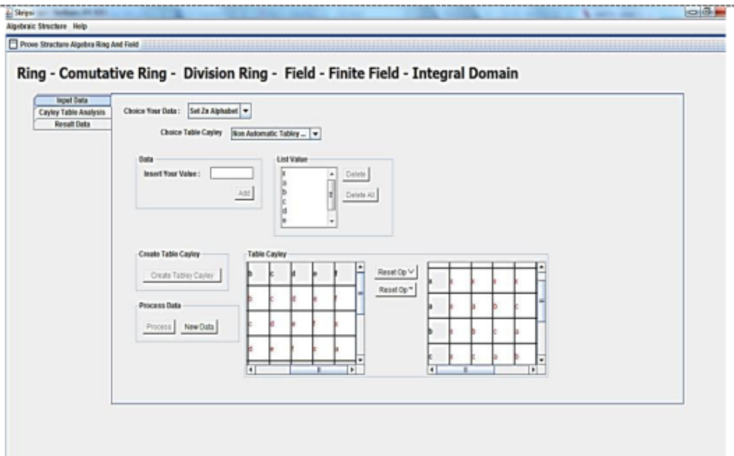

Figure 4. Input set of alphabet with designed Cayley table.

Then we execute the Cayley table analysis:

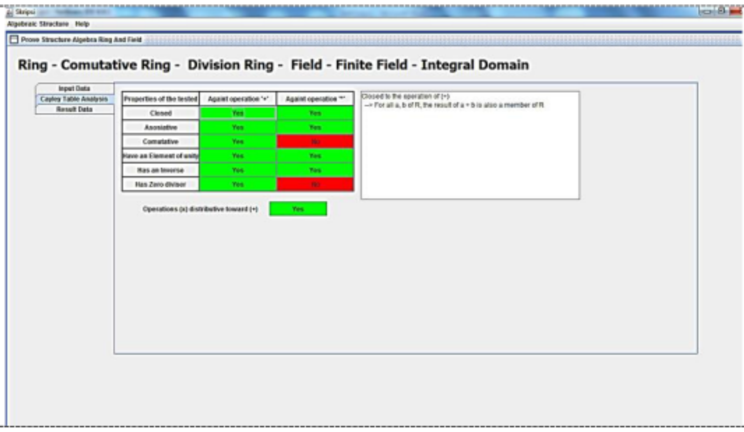

Figure 5. Cayley table analysis of alphabet with designed Cayley table.

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.

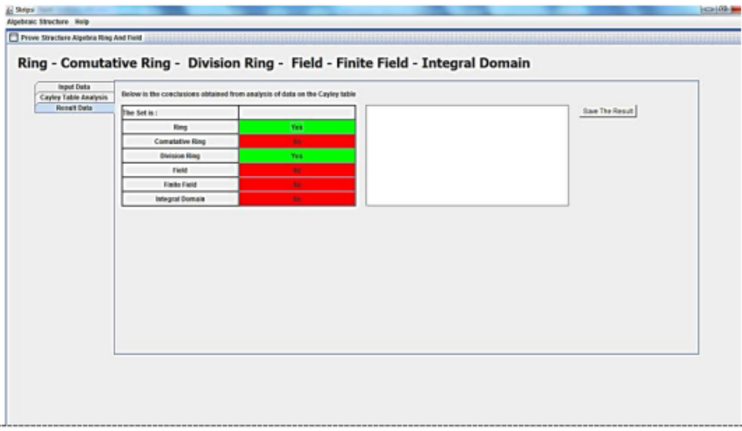

Figure 6. Conclusion of set of alphabet with designed Cayley table.

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.

Testing Division Rings and Fields Using a Computer Program

ORIGINALITY REPORT

17%	9%	14%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Ivy Tech Community College Central Office Student Paper	3%
2	dinarischita1207.blogspot.com Internet Source	2%
3	WEI-TING KARY CHIEN, WAY KUO. "OPTIMAL BURN-IN SIMULATION ON HIGHLY INTEGRATED CIRCUIT SYSTEMS", IIE Transactions, 1992 Publication	1%
4	A.R. AMIR-MOÉZ, A.L. FASS. "SOME ALGEBRAIC STRUCTURES", Elsevier BV, 1962 Publication	1%
5	Rida T. Farouki. "Chapter 2 Preamble", Springer Science and Business Media LLC, 2008 Publication	1%
6	Rosen, . "Algebraic Structures", Discrete Mathematics and Its Applications, 1999. Publication	1%
7	Yulianto, Budi, Eileen Heriyanni, Lusiana Citra Dewi, and Timothy Yudi Adinugroho.	1%

	"Architecture and Implementation of Instant Messaging in Educational Institution", Procedia Computer Science, 2015. Publication	
8	Yiming (Kevin) Rong, Samuel H. Huang, Zhikun Hou. "Fixture Modeling and Analysis", Elsevier BV, 2005 Publication	1%
9	www.freepatentsonline.com Internet Source	1%
10	www.sr2jr.com Internet Source	1%
11	Submitted to University of Bradford Student Paper	1%
12	www.m-hikari.com Internet Source	1%
13	ar.scribd.com Internet Source	1%
14	Submitted to University of Dehli Student Paper	<1%
15	math.uchicago.edu Internet Source	<1%
16	www.cse.wustl.edu Internet Source	<1%
17	Petros Elia. "", IEEE Transactions on Information Theory, 11/2007 Publication	<1%

18	Todd K. Moon. "Error Correction Coding", Wiley, 2005 Publication	<1%
19	bu.edu.sa Internet Source	<1%
20	rewards.theday.com Internet Source	<1%
21	digitalcommons.unl.edu Internet Source	<1%

Exclude quotes On

Exclude bibliography On

Exclude matches < 5 words