International Journal of Basic Science and Technology
March, Volume 8, Number 2, Page 59 - 63

ISSN 2488-8648
http://www.ijbst.fuotuoke.edu.ng/ 59

Congruence Relationship of Monoids in Machines

Emenone, C.E.

Department of Mathematics,University of Delta Agbor Delta State, Nigeria

Article Information

Article # 08009
Received: 12th April 2022
1st revision: 16th April 2022.
2nd revision: 17th April 2022.
Acceptance: 20th April 2022
Available on line:21st April 2022

Key Words
Monoid, Algebraic Structure, Binary Operation, Computer Arithmetic, Homomorphic Theory

Abstract

A monoid M = (M, . , e) is an algebraic structure which under a defined binary operation possesses the property of associativity and has identity element. i.e. for all x,y,z in M, the operation x.(y.z) = (x.y).z and x.e = e.x = e in the operation (x,y) →(x.y). If M is a monoid and R in M an equivalence relation on M satisfying the property $_xR_u$ and $_yR_v$ implies $_{x.y}R_{u.v}$, then R is a congruence relation. The usefulness of function to monoids transformation in connection to word construction in computer arithmetic is investigated. Machine approaches derived from homomorphic theory are used to demonstrate the linked concepts of simulation and realisation. Some theorems about algebraic structures and machines are given and demonstrated with appropriate illustrations.

*Corresponding Author**;** Emenone, C.E.; **cemenonye@gmail.com**

Introduction

The systematic study of algebraic structures could be made easier when a few algebraic systems are singled out to give insight into the overall algebraic theory and its usefulness in a real-life situation. This work has picked only monoids and some of their attendant properties. Monoid is one of the algebraic structures useful in word formation in computer machines.

In his mathematical machine theory, Anzt and Huckle (2019) stated that; if f_1 and f_2 are two abstract machines with the input monoid of f_1 free and there exists an injective monoid homomorphism J: M→ M_{f_1}, a surjective monoid homomorphism. E : M → M_{f_2}, and a function h : A_2 → A_1 such that it commutes. Then there exists a monoid homomorphism H : f_1 → f_2 a morphim of abstract machines.

Prather (1976) states that computer arithmetic are inherently finite, so there cannot be an exact correspondence (isomorphism) between this arithmetic and the actual number of a system they attempt to imitate. He posited that for many purposes imitation or simulation is quite acceptable because the correspondence of the actual numbers and their computer representations have been designed to preserve certain important algebraic features. According to Balachandran and Mungesu (2007), the characterization of homomorphic images using congruence relation on the domain in monoids and related concepts of simulation and realization are aptly illustrated through the machine design techniques deriving from homomorphism theory.

The monoids have great application in finite state machines, formal languages e.t.c. Kohazi (2007). Krohn-Rhodes in Wikipedia (2021) states that in theoretical computer science, the study of monoids is fundamental for automata theory.

Achary and Mahapatra (2010) in theoretical computers demonstrated the basis for the application of the homomorphism theory by making a connection to the language recognition capabilities of machines. They stated that if A and B are automata, then A is said to have the capability of B if there is a redesignation of I_A and a monoid homomorphism. Furthermore, Reingold (1974) in his homomorphism theory for automata showed that if $\emptyset: A \to B$ is a surjective automata homomorphism, then the kernel R is an automata congruence on A and B ≈ A/R

The significance of machines in the life of man cannot be over-emphasized hence some work has been done concerning the process of word formation in machines. Arbib (1969) emphasized the appropriateness of the use of categorical algebra notations and methods in the framework of automata theory. He showed that the proofs of basic results linking the notion of machines, normal forms of a machine, division of machines, and monoids are reduced to trivial exercises in commutative diagrams.

Group theory is one aspect of algebra studied in Mathematics and Sciences because of its wide range of applications. Monoids are specifically chosen because they are useful infinite machines and formal languages. The importance of homomorphism for algebra is applicable in the theory of monoids. The characterization of homomorphic images through congruence relations is very useful in monoids and the related concepts of simulation and realization are

ISSN 2488-8648
http://www.ijbst.fuotuoke.edu.ng/

shown through machine techniques deriving from homomorphic theory.

This work has picked only monoids and its attendant properties to illustrate the importance of algebraic structure to machines. The paper relates monoids and their congruence to word formation in machines and gives insight into how the compiler translates the high-level language to machine code. This eventually affords a little knowledge of how the machines function

Definitions

2.1 An algebraic structure $\mathbf{A_C = (A_C ; \mathfrak{T} ; P) = (A_1, A_2, \ldots\ldots\ \mathfrak{T} ; P)}$ is a collection of algebraic sets, functions and propositions written as

$\mathfrak{T} = \mathbf{U}\ \mathfrak{T}_a$ where each $\mathbf{F} \in \mathfrak{T}_a$ takes its values in the set A_a where A_a is the algebras of the structure (Sarachino, 1992)

2.2 A monoid M = (M, . , e) is an algebraic structure which under a defined binary operation possesses the property of associativity and has identity element. i.e. for all x,y,z in M, the operation x.(y.z) = (x.y).z and x.e=e.x=e in the operation (Audu *et al*., 2001).

(x,y) →(x.y). if M is a monoid and R in M an equivalence relation on M satisfying the property $_xR_u$ and $_yR_v$ implies $_{x.y}R_{u.v}$,then R is a congruence relation.

2.3 There are interactions between algebraic structures that are useful to science and technology. There exists certain relationship between pairs of element of a given set. If for each pair (x, y) of elements of a given set A the statement x ~ y has a meaning then ~ is a relation in A. Furthermore, if the relation ~ on the set A possesses the properties of:

(i) x ~ x for all x ∈ A;

$T_a \in T$ and $T_B \in T \Rightarrow T_{a\,o}\ T_B \in T$

Then T is called a transformation monoid

T = (T, $_0$, I_s) when composition is taken as monoid operation.

i.e.

$(T_{ao}, T_B)\ (s) = (T_B\,(s))$

is taken as the product of $T_{a0}\ T_B$ in the monoid T.

Obviously, $T_{a0}\,(T_B, {}_0, T_Y) = (T_{ao}, T_B)_O\ T_a$ and

$T_{a\,0}\ I_s = I_{s\,0}\ T_a = T_a$

shows that T is indeed a monoid.

2.4 Now consider the following monoids:

Let ∂_I be a finite set of symbols denoted by $\sum = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ and called an alphabet.

A finite sequence of symbols from ∑ written one after another without anyintervening punctuation is called a word x on ∑

Thus $x = \sigma_{I\,1}\ \sigma_{i\,2} \ldots \sigma_{I\,k}$

If the length of x is denoted by /x/ = K, then the null word x on ∑ (a sequence of length zero) is admitted in defining.

$$\sum * = \{x: x\ is\ a\ word\ in\ the\ alphabet\ \sum\}$$

i.e. $\sum * = \{x: x\ is\ a\ word\ on\ the\ alphabet\ \sum\}$ is the set of all words. generable from the given alphabet. (Humphreys, 1997)

Consider y on ∑ where $y = \sigma_{j\,1}, \sigma_{j\,2\ldots,}\sigma_{j\,m}$ and subsequently a juxtaposition (concatenation of words i.e. $x.y = \sigma_{i\,1}\ \sigma_{i\,2}\ldots\sigma i_k . \sigma_{j\,1}, \sigma_{j\,2\ldots,}\sigma_{jm}$)

here, it is obvious that

$x\,.\,e\ =\ e\,.\,x = x$ for every word x.

Then ∑∗ is the free monoid generated by the alphabet ∑

The following are examples of monoid generated juxtaposition:

Let $\sum = \{a, b, c, d\}$, then ∑∗ could be

abba dabba = abbadabba where (a, b, c, d) is in ∑∗

Now let ~ be a relation on a set A. then if;

(i) x ~ x ∈ A for x ∈ A

(ii) x ~ y implies y ~ x, for x, y ∈ A and

(iii) x ~ y and y ~ z implies x ~ z for all x, y, z ∈ A

then the relation is called equivalence relation i.e. a relation is called equivalence relation if its reflexive; symmetric and transitive.

A monoid is therefore defined as follows: Let M = (M, ., e) and consists of a set M such that x.y = y.x is in M and x(y.z) = (x.y)z is in M for x, y, z in M with a distinguished element e and binary operation (x, y) → x.y.

If for all x, y, z ∈ M, and $x.\,(y.z) = (x.\,y).\,z$

i.e. associativity and there is e ∈ M such that

x.e = e. x = x i.e. identity (Arigbadu, 1992)

2.5 Now let M = (M,. ,e) for any monoid and let R be an equivalent relation on the M that satisfies the additional property $_xR_u$ and $_yR_v$ implies $_{x-y}R_{u-v}$, then R is called a congruence relation M.

Let A be any set. A binary operation on A is a mapping of AXA into a set B containing A as a subset which maps each element (x, y) in AXA into an element x,y of B i.e. A rule of combination of any two elements of A to produce another object usually but not necessarily in A is a binary operation. (Aziken, 1996)

Transformation of Monoids

Monoids could be constructed from classes through algebra transformation. Functions or mapping can also be transformed from a set into itself. In computer science, a computer transforms programs from high level language into machine language. Codes transform numbers or alphanumeric data into binary sequences. If S is any set and T= (T_a) is a collection of mapping T_a : S → S which contains the identity

transformation I_s and is closed with respect to composition of mapping i.e.
aa: da = aada
cab.e = cab
Also if $\sum = \{0,1\}$, $then$ $\sum* = \{e, 0{,}00{,}01{,}10{,}000{,}001 \ldots.\}$

01101 = 01101
10101= 10101
01e =01

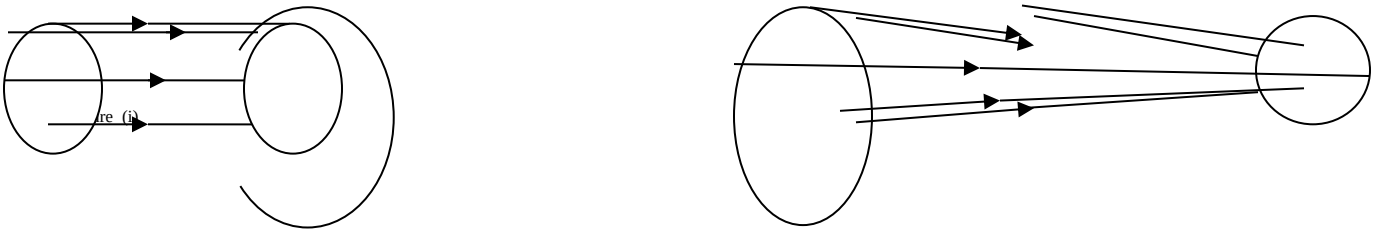
Homomorphism of Monoids

There exist a great deal of interaction between algebraic structures, and the monoid is not left out. The relationship between monoids is better understood through the way they operate on others. Yehoshafat (1967) defines homomorphism is a mapping from one algebraic system to another, which preserves structure. It is defined as follows:

Let (M,∗) and (N.A) be two monoids. A homomorphism from M to N is a mapping $\Phi : M \rightarrow N$ Such that for any x, y ∈ M, $\Phi(x * y) = \Phi(x) * \Phi(y)$.

The homomorphism theory isolates features of algebras and reveals precise conditions under which one algebra can simulate another. If $\Phi : M \rightarrow N$ is a homomorphism as defined above and Φ is 1 – 1(injective) then Φ is called monomorphic. If Φ is ontoΦ, Φ is called epimorphic. If Φ is both 1 – 1 and onto, Φ is an isomorphism from M to N.

The diagram below visualizes the effect of the above concept i.e.



In fig (i) the homomorphismΦ: M → N is injective. We can also say that N realizes M
Fig (ii) Φ: M → N is surjective (onto) N is a homomorphic image of N.

Illustrations

The following illustrations are made in this work:

(i) consider the monoid $\sum = (A, +, 0) and\ B = (B, +, 0)\ and\ the\ mapping\ \Phi : A \rightarrow B\ where$

$$\Phi(n) = \begin{cases} 0\ if\ n\ is\ even \\ 1\ if\ n\ is\ odd \end{cases}$$

Surely, $\Phi(n + m) = \Phi(n) + \Phi(m).\ mode\ 2$
This could be verified by considering the four cases

n	m
even	even
even	odd
odd	odd

If n and m are both even, then n + m are even i.e.
Φ(n + m) = 0 + 0 = Φ(n) + Φ(m)
If n and m are both odd, then n + m is even i.e.
Φ(n + m) = 1 + 1 = $\Phi(n) + \Phi(m) = 0$
Here B the homomorphic image of ∑ and B only implement the arithmetic rules.

(ii) Consider the free monoid N = (N, +, 0), the natural number.
Take X = {1}
The two conditions can be verified i.e.
Here, every n ∈N has a representation as an algebraic expression.
n = 1 + 1 + … +1 (n times)
If M is any monoid, assign Φ(1) in M, then the extension
$\Phi : N \rightarrow M\ is\ given\ by$
$\Phi(n) = \Phi(1) + \Phi(1) + \ldots + \Phi(1) = n\,.\Phi(1)$
If M is denoted additively, M = (M, +, 0), then
Φ(0) = 0

$\Phi(0) = 0 \,.\, \Phi(1) = 0$ *and*
$\Phi(n + m) = (n + m)\ \Phi(1)$
$= n\Phi(1) + m\Phi(1) = \Phi(n) + \Phi(m)$

This has shown monoid homomorphism. This is freely generated monoid homomorphism (Lehma, 1968)

Theorems: The theorems that are related to the word formation and techniques in computer machine operations is discussed in this section, The characterization of homomorphic images by means of congruence relations on the domain of monoids and related concepts of simulation is illustrated through machine design techniques deriving from homomorphic theory

(i). Let M be any monoid and Σ an alphabet.
If $\Phi: \Sigma \to M$ is a mapping, then Φ has a (unique) extension to a monoid homomorphism $\Phi: \Sigma \to M$. Consequently every finitely generated monoid is a homomorphic image of a free monoid.
Proof: Consider an arbitrary mapping $\phi: \Sigma \to M$ relative to an arbitrary monoid $M = (M, ., e)$.
For each word $= \sigma_{I\,L}, \sigma_{I\,2}, \ldots\ldots\ldots \sigma_{I\,K}$ in $\Sigma *$ where $e = \Phi(\in)$
Define $\Phi(x) = \Phi(\sigma_{i1})\ .\Phi(\sigma_{i\,2}) \ldots\ldots \Phi(\sigma_{i\,K})$, where the multiplication takes place in M.
Since $\Phi(x\,.\,y) = \Phi(\sigma_{i\,1}, \sigma_{i\,2}, .\, \sigma_{i\,K}\, \sigma_{j\,1}\sigma_{j\,2, \ldots\ldots}\sigma_{j\,m})$
$= \Phi(\sigma_{i\,1})\,.\,\Phi(\sigma_{i\,2}) \ldots.\, \Phi(\sigma_{i\,K})\,.\,\Phi(\sigma_{j\,1}).\,\Phi(\sigma_{j\,2}) \ldots\,.\,\Phi(\sigma_{j\,m})$
$= \Phi(x)\,.\,\Phi(y)$ _____(*)
(*) is an extension of Φ to monoid homomorphism $\Phi: \Sigma * \to M$
Furthermore, define another monoid homomorphism as $\varphi: \Sigma ** \to M$
With M = (M, ., e).
Here $\varphi(x) = \varphi(\sigma_{i\,1})\,.\,\varphi(\sigma_{i\,2}) \ldots\ldots\ldots \varphi(\sigma_{i\,K})$, And
$.\varphi(x) = \varphi(\sigma_{i1}).\,\varphi(\sigma_{i\,2}) \ldots\ldots\ldots \varphi(\sigma_{i\,K})$,
$.\varphi(x.y) = \varphi(\sigma_{i\,1}\sigma_{i\,2}, \ldots\ldots\ldots \sigma_{i\,K}\ \sigma_{j\,1}\sigma_{j\,2}, \ldots\ldots\ldots \sigma_{j\,m})$
$= \varphi(\sigma_{i\,1})\,.\,\varphi(\sigma_{i\,2}) \ldots\ \varphi(\sigma_{i\,k})\,.\,\varphi(\sigma_{j\,1})\,.\,\varphi(\sigma_{j\,2}) \ldots\varphi(\sigma_{jmi})$
$= \varphi(x)\,.\,\varphi(y)$ _____(**)
(*) and (**) are the same for the definition of x and y. $\varphi = \Phi$ *hence the same extension and so the uniqueness*.
Now let M be generated by a finite subset $\{\mu_1, \mu_2, \ldots, \mu_n\} \subseteq M$
Choose any alphabet Σ *with n symbols and design*
$\Phi(\sigma_{i\,1}) = \mu_i$ where i = 1, 2, …, n
It is clear here that Φ extends to a monoid homomorphism $\Phi: \Sigma * \to M$.
(ii). *Every monoid is isomorphic to a transformation monoid.*
Proof: Let $M = (M, ., e)$ *be any monoid and*
$T_M = \{ T_g: g \in M \}$ *be the collection of transformations*
$T_g: M \to M$ *given by* $T_g(x) = x\,.\,g$ *for* $x \epsilon M$.
This mapping is surjective and in addition
$T_g = T_h \Rightarrow T_g(x) = T_h(x)$ for all $x \in M$
$x\,.\,g = x.h$ *for all* $x \epsilon M$
$g = h$ (taking x = e)
Hence the mapping is also injective. It could be observed that the mapping runs the identity element of m into the identity element of
$T_M: e \to T_e = 1_m$
i.e. $[T_e(x) = x\,.\,e = x = 1_m(x)]$
Similarly, $T_{gh}(x) = x(gh) = (xg)\,h$
$= T_h(T_g(x)) = T_g\,.\,T_n(x)$
Conditions of isomorphism are satisfied. Every monoid is therefore isomorphic to a transformation monoid.
(iii) Let $\Phi: M \to M$ be a monoid monomorphism. Then the kernel of ϕ *is a congruence R on M* is isomorphic to M/R. Conversely if R is a congruence relation on M, then there is a surjective homomorphism $\Phi: M \to M/R$ with kernel R.
Proof Let R denote the kernel of $\phi: M \to M$
Suppose $_xR_u$ and $_yR_v$, by definition of kernek we have

$$\phi(xy) = \phi(x)\,\phi(y) = \phi(u)\ \phi(v) = \phi(uv)$$
$$= \phi_{xy}R_{uv}$$

Hence the congruency of R on M.

Now $\Phi(x)\phi(y) <=> {}_xR_y <==> [x] = [y]$

The mapping is well defined and injective.

Similarly we can find an element that maps onto a given $[x]\ in$ M/R viz $\phi(x)$

Also e = $\phi(e) \rightarrow [e]$ is the identity element in M/R

It is therefore obvious that the mapping is also surjective.

We need investigate if the mapping preserves product.

Suppose $\phi(y) \rightarrow [y],\ then\ \phi(x).\phi(y) = \phi(xy) \rightarrow [xy] = [x].[y]$

Conversely, if R is congruence on M, then the mapping

$\Upsilon(x) = [x]$ *is easily seen to be a surjective homomorphism*

$\Upsilon: M \rightarrow$ M/R

And since $\Upsilon(x) = \Upsilon(y) \Leftrightarrow [x] = [y] \Leftrightarrow {}_xR_y$

The kernel of Υ is given congruence relation R.

Conclusion

Data processing was greatly aided and simplified by the arrival of computer machines. The ensuing dynamic enhancements just added to the intrigue. The usefulness and contribution of science and technology in general, and mathematics in particular, may be seen in the mechanisms of these devices. Through the interaction of algebraic structures, this research has provided some insight into the machine mechanism. The creation of an algorithm necessitates the collaboration of numerous algebraic systems. Compilers, which translate high-level languages into machine language, make computer machine operation simple. In this process of machine operations, the monoid as an algebraic structure is critical. In machines, the monoid has been demonstrated to be useful in the word production process.

References

Anzt, H. and Huckle, K. (2019). The fundamentals of monoids in machines. *The international Journal of high-performance computing applications*. 33(6), 1069-1078.

Achary, B.P. and Mahapatra, T. (2007). Numerical computation of complex principle value integral.

Arbib, M.A. (1969.) Algebraic theory of machines, languages and semi groups. New York, Academic press.

Arigbadu,B. (1992). Elementary Number theory and Abstract Algebra. London Anfat publishers.

Audu, M.S., Asibong-Ibe, U,J.,Ajala, S.O., Kenku, M.A., Osondu, K.E. and Ajetumobi, M.O.(2001). Lecture notes series number 1. National Mathematical Centre, Abuja Nigeria.

Aziken, G.N. (1996). Numbers and their properties. Lagos, End Time Publishing Home

Balachendra,K .and Mumgesu, A. (2007). Optimal control of singular system via single-term walsh series.153-159. New York, Academic Press.

Goldstain, L.T. (1973). Abstract Algebra,Afirst course. New Jersey. Prentice-Hall Englewood Cliff.

Humphreys, F.J. (1997). A course in Group theory. London, Oxford University Press.

Kohazi, Z. (1970). Switching and Finite Automata theory. New York Mc. Graw- Hill Publishers.

Lehma, D.H. (1968). Machines and Pure Mathematics, pp390

Okonta, P.N. and Emenonye, C.E. (2001). Basic Abstract Algebra. Warri, COEWA Publishers.

Prather, R, (1976). Discrete Mathematical Structure for Computer Science. Boston-Houghton Mifflin Company.

Reingold, B. (1974). Computer Approaches to mathematical problem. New Jersey Prentice-Hall Englewood Cliff.

Saracino, D. (1992). Abstract Algebra- A first course. Illinois. Prospects Heights Publishers.

Yehoshafat, G. (1967) Some properties of the free monoid with applications to automata theory. *Journal of Computer and system Sciences*; 1, 137-154

Wikipedia (2021) History of Computer Science. https://en.m,wikipedia.org.