A STUDY OF CATEGORICAL MODEL OF MULTISETS

BY

ISAH, AHMED IBRAHIM

DEPARTMENT OF MATHEMATICS

FACULTY OF SCIENCE

AHMADU BELLO UNIVERSITY

ZARIA, NIGERIA

NOVEMBER, 2012

A STUDY OF CATEGORICAL MODEL OF MULTISETS

BY

ISAH, AHMED IBRAHIM

B. Sc (A.B.U. 1998).

M.Sc/SCIE/03051/2008-2009

A THESIS SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES, AHMADU BELLO UNIVERSITY, ZARIA, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER DEGREE IN MATHEMATICS

DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE

AHMADU BELLO UNIVERSITY,

ZARIA, NIGERIA

NOVEMBER, 2012

DEDICATION

This thesis is dedicated to my late parents.

DECLARATION

I hereby declare that the work in this thesis titled "*A Study of Categorical Model of Multisets"* was performed by me in the Department of Mathematics, under the supervision of Prof. D. Singh and Dr. A. M. Ibrahim. The information derived from the literature has been duly acknowledged in the text and a list of references provided. No part of this work has been presented for another degree or diploma at any institution.

---------- ----------

Ahmed Ibrahim Isah Date

CERTIFICATION

This thesis entitled: A STUDY OF CATEGORICAL MODEL OF MULTISETS by Ahmed Ibrahim ISAH meets the regulations governing the award of the degree of Master of Science of Ahmadu Bello University, and is approved for its contribution to knowledge and literary presentation.

---------- ----------

Professor D. Singh Date

Major Supervisor

---------- ----------

Dr. A. M. Ibrahim Date

Minor Supervisor

---------- ----------

Professor S. U. Momoh Date

External Examiner

---------- ----------

Dr. B. Sani Date

Head of Department

---------- ----------

Professor A. A. Joshua Date

Dean, School of Postgraduate Studies

ACKNOWLEDGMENT

I first thank Almighty Allah, the greatest source of knowledge who out of his mercy granted me the energy, strength, wisdom and power to undertake and complete this work successfully. My special regards and appreciation go to my supervisors Prof. D. Singh and Dr. A. M. Ibrahim for their guidance and patience in giving valuable advice and constructive suggestions throughout the period of the research. I wish to express my appreciation to all members of the Department of mathematics, A. B. U. Zaria for extending their support and encouragement. In particular, I wish to extend my appreciation to Dr. B. Sani the Head of Department, Prof B. K. Jha, Dr. J. Singh, Dr. A. A. Tijjani, Dr. A. Mohammed, Mal. I. A. Fulatan and Mal. A. Alkali.

I wish to extend my appreciation to my relatives, friends and colleagues who in one way or the other contributed to the success of the work.

I want to use this opportunity to condole the Department and the entire university community on the death of Dr. U. M. Makarfi and our colleague Ibrahim Bashir Enaji who passed away within the period, may the Almighty Allah grant them an eternal rest, Amen.

ABSTRACT

We present a crisp and critical survey of the development of the theory of categories which formed the content of our paper titled, *A Survey of the Development of the Theory of Categories*, published at *International Journal of Innovation Management and Technology*, vol.3, No. 2, (2012) pp. 156-159. It is observed that the theory originally arose in mathematics out of the need of formalism to describe the passage from one type of mathematical structure to another, and later became an autonomous field of research that has now occupied a central position in most of the branches of mathematics, some areas of theoretical computer science and mathematical physics. We emphasized that, basically the theory of categories studies structures in terms of the mappings between them. This as well applies to categories themselves taken as structures with functors as the mappings between them. Furthermore, the mappings in categories can be applied to functors to form mappings between the functors called *Natural transformations.*

We show that multisets determine a category denoted **Mul.** Fundamental notions of categories of sets are defined in multiset context. Finally, we develop and prove various propositions in **Mul.**

TABLE OF CONTENTS PAGE

Dedication……iv

Declaration ……v

Certification ……vi

Acknowledgement……vii

Abstract……..……viii

Table of contents……..……..……ix

Chapter one..……..……..1

General Introduction..……1-5

1.1 Background of the study……1

1.2 Statement of the problem……..……4

1.3 Motivation of the research……..……4

1.4 Objective of the research……5

1.5 Organization of the research…….……..5

Chapter two..……6

Literature review...……..6-16

Chapter three..……..……..17

Preliminaries for developing category theory……17-21

3.1 Coretraction and Retraction …….……....17

3.2 Concept of homomorphisms……..……...18

Binary operation…….……18

Homomorphism…..……...……..18

3.3 Commutative diagram.……18

3.4 Algebraic operations on functions……21

Chapter four…..……22

Fundamentals of category theory……..22-61

4.1 Introduction……22

4.2 Definition of category……23

Proposition……25

4.3 Examples of categories..……26

4.4 Types of categories……32

4.5 Constructions on categories……33

4.6 Properties of morphisms……...36

Monomorphism and split monomorphism…..……36

Epimorphism and split epimorphism……38

Bimorphism……39

Isomorphism…….....40

Endomorphism……..43

Idempotent……..43

Right and left equivalent……..45

4.7 Properties of objects……46

Initial and terminal objects…….……47

Proposition…….……47

Product of two objects…….……48

Coproducts……..…….……49

Equalizer…….……..…….50

Coequlizer…….…….……...…….51

4.8 Functors……..…….53

Examples of functors……..…….…….53

Types of functors……..…….54

4.9 Natural transformation..……..……..57

Examples of natural transformation.……..……..62

Chapter five Category of multisets…….……..62-76

5.1 Concept of Multisets……..…….……...62

Definition and representation of multisets…….……63

Multiset functions…….……64

Composition of multiset functions……..…….65

5.2 Category of multisets…….……..66

Monomorphism, epimorphism, biomorphism and isomorphism.……67

Dual of a **Mul…**……..70

Objects in **Mul**…….70

Points of an object…..……..……..71

Product of two objects..…..……..……..72

Dual of product-projections…….…..……..……76

Chapter six…….…….……..…….77

6.1 Summary of our major findings…..……..……..77

6.2 Conclusion and some future directions…….……..79

References……..……..80

CHAPTER ONE

GENERAL INTRODUCTION

1.1 Background of the study

In 1942, Samuel Eilenberg and Saunders MacLane presented a paper introducing Specific *Functors and Natural Transformation* at work, particularly confined to the study of groups. Later in 1945, they presented another paper titled *General Theory of Natural Equivalence* introducing Categories, Functors and Natural Transformations as part of their work in topology, especially algebraic topology. Their goal was to understand natural transformation for which the notion of functors and that of categories were exploited (see [Marquis (2010)], for details).

Category theory can also be seen as a continuation of the work of Emmy Noether, one of MacLane's teacher, on formalizing *abstract processes*. Noether realized that in order to understand a type of mathematical structure, one needs to understand the processes preserving that structure. In order to achieve this understanding, Eilenberg and MacLane in 1945, proposed an axiomatic formalization of the relation between structures and the processes preserving them. This indicates that category theory originally arose in mathematics out of the need of formalism to describe the passage from one type of mathematical structure to another.

Initially, it was not clear whether the theory of category would turn out to be more than just a convenient language which indeed was the case for about ten years. This perspective altogether changed when categories started getting used in homology theory

and homological algebra. In this connection, Maclane and Grothendieck independently described categories in which the collections of morphisms between two fixed objects were shown to have an additional structure. More specifically, it was shown that given two objects X and Y of a category **C**, the set Hom(X,Y) of morphisms form an abelian group (see [Marquis (2010)], for details).

Also for reasons related to the ways homology and cohomology theories were linked, the definition of a category had to satisfy an additional formal property which led to the definition of a category that is now being commonly used.

In the 1960s, Lambek proposed to describe categories as *deductive* systems. He begins with the notion of a graph consisting of two classes *Arrows* and *objects*, and two mappings between them, s: Arrows $\rightarrow$ objects and t: Arrows $\rightarrow$ objects, namely the source and the target mappings. The arrows are usually called the *oriented edge* and the objects called *nodes* or *vertices*.

With these developments, category theory became an autonomous field of research. In fact many branches of modern mathematics could be conveniently described in terms of categories; for example, category of sets, category of relations, category of groups, etc., and most importantly, doing so often reveals deep insights and similarities between seemingly different areas of mathematics (see [Barr and Wells (1999) and [Oosten (2002)]], for details).

Indeed, along with its rapid growth as a branch of mathematics, after the appearance of Lawvere's Ph.D. thesis in universal algebra (1970), it started getting used in computer Science. However, it still remains to be seen whether category theory should be taken as an alternative to set theory.

Summarily, the study of categories can be seen as an attempt to axiomatically comprehend what is characteristically common in various classes of related mathematical structures by way of relating those exploiting structure-preserving functions between them.

In view of the fact that a computer is not good at viewing concrete diagrams, category theory is being extensively used in computer science mainly because it offers a constructive mathematical structure to describe an object.

In addition, we invariably encounter with systems which contain objects with repeated elements or attributes (for example, groups of people, systems of elementary particles, etc., having two or more elements with the same property). We need a (formal) mathematical structure to model this kind of data. In the recent years such mathematical structures have been developed which are in general called *multiset-based structures*. Note that a *multiset* is a collection of objects in which repetition of elements is considered significant. Accordingly, sets are merely special instances of multisets.

Applications of multiset abound, especially in mathematics and computer science, (see [Singh *et al.* (2007) and Blizard (1991)], for details). Typically, multisets have been used in data base theory, membrane computing, etc. For example, in membrane computing, each membrane can be viewed as a collection of objects appearing in multiple copies.

Recently, considering the applications of multisets, on the one hand, and that of categories on the other, categories of multisets are being studied.

1.2 Statement of the problem

As we stated above, category theory is extensively used in Mathematics and Computer Science. However, until recently, almost all the developments that have taken place in the theory of Categories are largely in set theoretic environment. In this thesis, we attempt to investigate the theory of category of multisets.

1.3 Motivation of the research

The motivating factors are basically drawn from various areas of applications of category theory and the pervasive use of multisets. For example, in Lann (2003), functional programming language is described as categories, Cockett (2010) describes categories as graphs with composition, while Singh and Peter (2011) show that trees can be represented in the form of wellfounded multisets.

Category theory has come to occupy a central position in most of the branches of mathematics, some areas of theoretical computer science where they correspond to *types* (a data type is a set of data with values having predefined characteristics like integer or

floating point, usually a limited number of such data are build into a language and this corresponds to a category), and mathematical physics where Categories are used to describe *vector spaces.* By now it has emerged as a powerful language or a conceptual framework providing tools to characterize the universal components of a family of structures of a given kind and their relationships. In Summary, category theory can be regarded as a mathematical theory of structures.

1.4 Objective of the research

In this thesis we aim to study fundamental aspects of category theory and to construct a Categorical model of multisets. In other words, we aim to present the theory of category of multisets.

1.5 Organization of the research

In addition to chapter 1, chapter 2 will cover literature review. In chapter 3 we propose to study some preliminaries for developing category theory. Chapter 4 is devoted to the fundamentals of category theory. Chapter 5 describes the theory of category of multisets. Finally, chapter 6 consists of the summary of the main results obtained and some directions for future research.

CHAPTER TWO

LITERATURE REVIEW

Categories were first introduced in the course of formulating algebraic topology, specifically with Samuel Eilenberg's observation that Saunders Maclane's calculations on a specific case of a group extension coincided precisely with Norman Steenrod's calculation of the homology of a *solenoid*. Eilenberg and Maclane's effort to make sense of this coincidence across apparently distinct areas of mathematical inquiry gave rise to introducing *category theory*. The central notion at work was that of *natural transformations*. In order to provide a broad mathematical perspective, the notion of *functor* was introduced for which they borrowed the term *category* from the philosophical writings of Aristotle, Kant and reiterated in C. S. Peirce.

Emmy Noether (one of Maclane's teachers), in formalizing abstract processes, realized that understanding of a mathematical structure in its proper perspective could be better achieved through a proper understanding of the processes preserving that structure. Maclane and Eilenberg proposed an axiomatic formalization of the relation between structures and the processes preserving them, which is considered as a first sustained formalization of Noether's intuitive notion of the concept of category.

Eilenberg and Maclane define *category* ***C*** as an aggregate ***Ob*** of abstract elements, called the *objects* of **C** and abstract elements ***Map***, called *mappings* of the category. The term ***Map*** is characterized as follows:

(C1) Given three mappings α_1, α_2, α_3, the *triple product* $\alpha_3(\alpha_2\alpha_1)$ is defined if and only if $(\alpha_3\alpha_2)\alpha_1$ is defined. That is, whenever either is defined, the associative law $\alpha_3(\alpha_2\,\alpha_1) = (\alpha_3\,\alpha_2\,)\,\alpha_1$ holds. This triple product is often written as $\alpha_3\alpha_2\alpha_1$.

(C2) The triple product $\alpha_3\alpha_2\alpha_1$ is defined whenever both the products $\alpha_3\alpha_2$ and $\alpha_2\alpha_1$ are defined.

(C3) For each mapping α, there is at least one identity ℓ_1 such that $\alpha\ell_1$ is defined and at least one identity ℓ_2 such that $\ell_2\alpha$ is defined.

(C4) The mapping ℓ_x corresponding to each object X is an identity.

(C5) For each identity ℓ there is a unique object X of **C** such that $\ell_x = \ell$.

It has been reported that objects play a secondary role and could be entirely done away from the definition; however, it would make the manipulation of the applications less convenient (see [Marquis (2010)], for details).

In fact, the definition of category formulated by Eilenberg and Maclane emerged as a helping tool to provide an explicit and rigorous formulation of the notions of functors and natural transformations.

As described above, a category is an algebraic structure consisting of a collection of *objects*, linked together by a collection of *arrows* (*morphisms*) that have two basic properties: the ability to compose the arrows associatively and the existence of an identity arrow for each object. *Objects* and *arrows* may be comprehended as abstract entities.

In the 1960s, categories were studied by Lambek as *deductive* systems. The notion of a *graph* consisting of two classes *Arrows* and *objects*, and two mappings between them,

s: Arrows $\rightarrow$ objects and t: Arrows $\rightarrow$ objects, namely the source and the target mappings. The arrows are usually called the *oriented edge* and the objects called *nodes* or *vertices* were put to work. After the appearance of Lawvere's Ph.D. thesis in universal algebra (1970), the theory of categories has finding applications in computer Science.

Blass (1984) emphasizes that category theory interacts with nearly everything; it is a remarkable empirical fact that the important structural properties of mathematical objects are often expressible in category-theoretic terms, specifically as a universal property. He further surveys the ways category theory interacts with set theory for better understanding of category theory. It is observed that many of the elementary concepts of category theory were introduced for the purpose of expressing familiar concepts of set theory and their generalizations in other areas of mathematics.

Fokkinga (1994), in his book titled *A Gentle Introduction to category theory*, studies category theory from applications point of view, specifically within *algorithmics (problem solving)*. It is observed that the language of category theory facilitates providing an elegant style of describing *expressions* and *proof (equation reasoning)* for use in algorithmics. This happens to be reasoning at the functional level without the need (and the possibility) to reduce arguments explicitly. The equational formulas often lead to a far-reaching generalization much more than the usual set-theoretic formulations. He defines category as a *data* characterized as follows:

The data :

(i) A collection of things called *objects*. We use A, B, C,...,X,Y,Z,... to denote objects.

(ii) A collection of things called *morphisms*, sometimes called *arrows*. We use f, g, h,..., and later on, also α, β, φ, ψ, χ, ..., to denote morphisms. The collection of all arrows of a category C is sometimes denoted ArrC.

(iii) A relation on morphisms and pairs of objects called *typing* of the morphisms. The relation is denoted f: A $\rightarrow$ B for morphism f and objects A, B. Here, A is the domain and B the codomain of f: A $\rightarrow$ B.

(iv) A binary partial operation on morphisms called *composition*. We use f; g to denote the composition of morphisms f and g. An alternative notation is g o f or g f.

(v) For each object A, a distinct morphism called *identity* on A. Note that, id_A, or id when A is clear from the context denotes the identity on object A.

The axioms:

There are three *typing* axioms and two axioms for *equality*.

Typing axioms:

(T_1) f : A $\rightarrow$ B and f : A′ $\rightarrow$ B′ implies A = A′ and B = B′. (Unique – Type)

(T_2) f : A $\rightarrow$ B and g : B $\rightarrow$ C implies f ; g : A $\rightarrow$ C. (Composition – Type)

(T_3) id_A : A $\rightarrow$ A. (Identity – Type)

A morphism term f is called *well-typed* if, a typing f: A $\rightarrow$ B can be derived for some objects A, B according to the aforesaid axioms.

Axioms for equality of morphisms:

(E_1) (f ; g) ; h = f ; (g ; h) (Composition –Assoc.)

(E_2) id ; f = f = f ; id (Identity)

Also, whenever a term is written it is assumed that the variables are *typed* (at their introduction-mostly an implicit *universal quantification* in front of the formula) in such a way that the term is well-typed. This allows us to simplify the formulations considerably, as illustrated in the axioms for equality of morphisms above.

Barr and Wells (1999) define a category **C** as a graph together with two functions c: $C_2 \rightarrow C_1$ and u: $C_0 \rightarrow C_1$ satisfying the properties C-1 through C-4. The elements of C_0 are called *objects,* those of C_1 are called *arrows,* and C_2 is the set of paths of length 2. The function c is called *composition* and if (g,f) is a composable pair, c(g,f) is written $g_{\circ}f$ and is called the *composite* of g and f. If A is an object of **C**, u(A) is denoted id_A, called the *identity* of the object A.

C-1 The source of g o f is the source of f, and the target of g o f is the target of g.

C-2 (h o g) o f = h o (g o f), whenever either side is defined.

C-3 The source and target of id_A are both A.

C-4 If f: A $\rightarrow$ B, then f o id_A = id_B o f = f.

Category theory is described as a mathematical structure. Subsequently, they describe Functional Programming Languages as Categories.

Landry (1999) observes that category theory provides a means for organizing and classifying the relationships between mathematical concepts in mathematical models. It is emphasized that the theory of category is the language of mathematical concepts and relations and thereby making it the language of mathematics.

Burgin (2000), while discussing unified foundations for mathematics, observes that morphism is the central concept in a category. A category **C** consists of the collection Mor_C of the morphisms of **C.** Objects of **C** are associated with identity morphisms 1_A, since 1_A is unique in each set $Mor_C(A, A)$ and uniquely identifies the object A.

Turi (2001) Observes that category theory is the mathematical study of universal properties and that, it brings to light, makes explicit, and abstracts out the relevant structures often hidden in following traditional approaches. It also looks for the universal properties holding in the categories of structures one is working with. He defines a Category **C** as consisting of the following:

(i) a collection Obj_C of objects A, B, C, ... , X, Y, Z, ...,

(ii) a collection **C**(A, B) of arrows f: $A \to B$ from A to B for each pair of objects A and B.,

(iii) for each object A, an identity arrow $id_A : A \to A$.,

(iv) for each pair of arrows f : $A \to B$, g: $B \to C$, a composite arrow g o f: $A \to C$.

These data satisfy the following generalised *monoid laws:*

1. Identity: If f: $A \to B$ then id_B o f = f o id_A
2. Associativity: If f: $A \to B$, g: $B \to C$ and h: $C \to D$ then

(h o g) o f = h o (g o f).

A monoid is a category with only one object and a pre-order is a category with at most one arrow between every two objects are most simple examples of categories.

Rydeheard and Burstall (2001), while describing *computational category theory*, try to illustrate the connection between *programming* and *category theory*. It is observed that

category theory seems to operate on the same level of generality as logic and computer programming. The essential virtue of category theory is to provide suitable means for making definitions which is the programmer's main task. Computer scientists are fascinated by category theoretic approaches because it is largely *constructive*. Theorems asserting the existence of objects are proven by explicit constructions. In this sense, category theory may be viewed as a *warehouse* of algorithms. They define a category as a graph (O,A,s,t) whose nodes *O* are called *objects* and edges *A arrows*. Associated with each object a in O, there is an arrow i_a: a $\rightarrow$ a, the identity arrow a, and to each pair of arrows f: a $\rightarrow$ b and g: b $\rightarrow$ c, there is an associated arrow gf: a $\rightarrow$ c, the composition of f with g. The following equations must hold for all objects a, b, c and d, and arrows f: a $\rightarrow$ b, g: b $\rightarrow$ c and h: c $\rightarrow$ d, (hg)f = h(gf) and $fi_a = f = i_b f$.

Oosten (2002) defines a category in line with Turi. It is observed that every directed graph can be made into a category where the objects are the vertices of the graph and the arrows are paths in the graph.

Laan (2003), in line with Turi and Oosten, observes that a Poset (P, $\leq$) can be regarded as a category P with the set of object P, along with the provision that if x, y $\in$ P, then P(x, y) consists of exactly one morphism if x $\leq$ y and is empty otherwise. Moreover, this shows that every set can be viewed as a discrete category i.e., a category where the only morphism is the identity morphism. Further, he describes functional programming languages as categories.

Streicher (2004) defines a category as consisting of the following:

(i) a class Ob(**C**) of objects of **C**.,

(ii) a family Mor(**C**) associating with every pair A, B ∈ Ob(**C**), a class Mor(**C**)(A, B) of morphisms from A to B.,

(iii) for all A, B, C∈Ob(**C**), a mapping $o_{A,B,C}$: Mor(**C**)(B, C) × Mor(**C**)(A, B) → Mor(**C**)(A, C), called *composition.,*

(iv) for A∈Ob(**C**) a distinguished morphism id_A∈Mor(**C**)(A, A), called the *identity* morphism for A.

These data satisfy the following:

1. For all A, B, C∈Ob(**C**) and f∈Mor(**C**)(A, B), g∈Mor(**C**)(B, C) and h∈Mor(**C**)(C, D), h o (g o f) = (h o g) o f i.e., $o_{A,C,D}(h.o_{A,B,C}(g.f)) = o_{A,B,D}(o_{B,C,D}(h.g).f)$.

2. For all A, B, C∈Ob(**C**) and f∈Mor(**C**)(A, B) and g∈Mor(**C**)(C, A), f o id_A = f and id_A o g = g i.e., $o_{A,A,B}(f.id_A) = f$ and $o_{C,A,A}(id_A.g) = g$.

Further, he elaborates that a category whose objects are sets, whose morphisms from A to B are the set-theoretic functions from A to B and where composition is given by (g o f)(x) = g(f(x)), is denoted **Set**. It follows that in **Set**, the identity morphism id_A takes every x∈A to itself.

Lawvere (2005) adjoins eight first-order axioms to the usual first-order theory of category to obtain an elementary theory of the category of sets where he shows that it provides a foundation for mathematics, which is quite different from the usual set theoretic formulation in the sense that much of number theory, elementary analysis, and algebra can be constructively developed within it.

Murfet (2006), while describing the *Grothendieck Universe* (or just a *universe*), define a category as an algebraic object similar to a group or a ring.

Further, he elaborates that if we desire to talk about the category of *all sets* , *all groups*, *all topological spaces*, along with not allowing the paradox of *too long sets* (Russell's Paradox) to appear, we need to introduce *fancy* set-theoretic ideas (like Grothendieck Universes: the full second order version of cumulative hierarchy).

Lawvere and Schanuel (2009) explore the concept of category theory which they call a *mathematical universe.* They first describe some examples and then, develop the theory from some basic definitions of maps, composition and algebra of composition. It is defined as a system consisting of:

(i) Objects and maps.

(ii) For each map f, one object as domain and one object as codomain.

(iii) For each object A, an identity map consisting A as domain and codomain.

(iv) A composite map gf: A $\rightarrow$ C for each pair of maps f: A $\rightarrow$ B, g: B $\rightarrow$ C satisfies identity and associativity laws.

Ivancevic and Ivancevic (2009) define category K as a pair (ob(K), mor(K)) of generic objects A, B,... in ob(K) and generic arrows f: A $\rightarrow$ B, g: B $\rightarrow$ C,... in mor(K) between objects, with associative composition and identity (loop) arrow.

Subsequently, it is observed that the theory was born with an observation that many properties of mathematical systems could be unified and simplified by a presentation with commutative diagrams of arrows.

Cockett (2010) describes category as a collection of maps which have a partial associative multiplication and a system of units. This is an interesting view as it indicates that the objects are actually redundant structure and their role can be replaced by the identity maps. It is further remarked that categories and functors form a category called **Cat.**

Marquis (2010) observes that the notion of category generalizes those of a preorder and monoid and that the theory provided unification within set theoretical environment, organizing and unifying much of mathematics.

As noted earlier, there are few contributions on category theory in multiset context. Monro (1987) is the first to notice that multiset and multiset functions naturally determine a category. Syropoulos (2003) describes a categorical approach to multiset and, in line with Monro, elaborates the idea that multiset and multiset morphisms form a category, denoted **Mul.**

We intend to present a systematization of the fundamental aspects of category theory and to construct a categorical model of multisets. In other words, we intend to present the theory of category of multisets as our main contribution in this thesis.

CHAPTER THREE

PRELIMINARIES FOR DEVELOPING CATEGORY THEORY

Introduction

In this chapter, we present some preliminaries for developing category theory.

The following definitions are borrowed from various sources.

3.1 Coretraction and Retraction

Coretraction:

Let α: S $\rightarrow$ T be a mapping (map for short). A map β_1 : T $\rightarrow$ S is called a *left inverse* of α if $\beta_1\alpha = \in_S$ (an identity map on S). If α has a left inverse, then it is called a *coretraction*. α is also called *left invertible* and S is then called a *retract* of T.

It follows that α is a coretraction $\Leftrightarrow$ α is mono (injective) $\Leftrightarrow$ α is left cancellable (i.e., if β,γ are maps such that $\alpha . \beta = \alpha . \gamma \Rightarrow \beta = \gamma$).

Retraction:

Let α: S $\rightarrow$ T be a map. A map β_1: T $\rightarrow$ S is called a *right inverse* of α if $\alpha\beta_1 = \in_T$ (an identity map on T). If α has a right inverse, then it is called a *retraction*. α is also called *right invertible*.

It follows that α is a retraction $\Leftrightarrow$ α is epi (onto or correspondence) $\Leftrightarrow$ α is right cancellable (i.e., if β,γ are maps such that $\beta . \alpha = \gamma . \alpha \Rightarrow \beta = \gamma$).

A mapping α is both a coretraction and a retraction if it is *bijective*. In this case, the left and right inverses are equal i.e., α has a unique two-sided inverse.

3.2 Concept of Homomorphisms

Binary operation:

Let S,T,V be sets. A *binary relation* from SxT $\rightarrow$ V is any relation from SxT $\rightarrow$ V. A *binary operation* (or *binary composition*) from SxT $\rightarrow$ V is a map from SxT $\rightarrow$ V. A binary operation α: SxS $\rightarrow$ S is called a *binary operation* on S i.e., a binary operation on a set S is a binary relation that maps elements of the Cartesian product SxS to S.

Homomorphism or Morphism (Mor, for short)

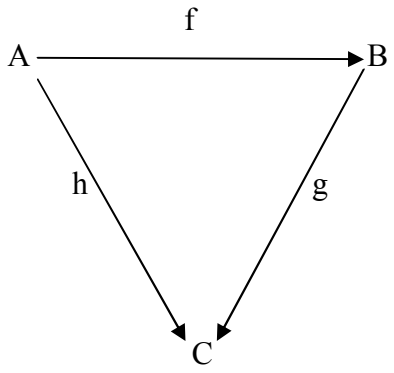
Let (S,α), (T,β) be two groupoids. A map γ : S $\rightarrow$ T is called *a homomorphism* if $\gamma(a\alpha b) = \gamma(a)\beta\gamma(b)$ i.e., $\alpha(a,b) = \beta(\gamma(a), \gamma(b))$ $\forall a,b \in S$. In other words, a morphism is a function which respects operations. A Mor γ is called a *monomorphism;* an *epimorphism;* an *isomorphism* according as γ itself is mono; epi; bijective, respectively. A Mor: S $\rightarrow$ S is called an *endomorphism*. An invertible *endomorphism* is called an *automorphism.*

3.3 Commutative Diagrams

A graph (diagram) is an abstract representation of a set of objects where some pairs of the objects are connected by links. The interconnected objects are represented by mathematical abstraction called *vertices*, and links that connect some pairs of vertices are called *edges*. Typically, a graph is depicted in diagrammatic form as a set of dots for the vertices, joined by lines or curves for the edges. The edges may be directed (asymmetric) or undirected (symmetric). For example, if the vertices represent people at a party or meeting, and there is an edge between two people if they shake hands, then this is an undirected graph, since if person A shook hands with person B, automatically person B

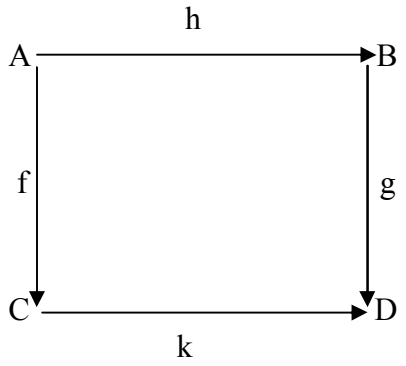
also shook hands with person A. On the other hand, if the vertices represent people at a party and there is an edge from person A to person B when person A knows person B, then this graph is directed because knowing some one is not a symmetric relation (i.e., one person knowing of another person does not necessarily imply the reverse). This latter type of graph is called a *directed graph* and the edges are called *directed edges* or *arcs*. Vertices are also called *nodes* or *points* and edges are also called *lines*. In other words, a directed graph is an ordered pair D = (V, A) where V is a set whose elements are called *vertices* or *nodes* and A is a set of ordered pairs of vertices called *arcs*, *directed edges* or *arrows*. An arrow a = (x, y) is considered to be directed from x to y; y is the direct successor of x and x is said to be a direct predecessor of y.

A commutative diagram is a diagram of objects (also known as vertices) and morphisms (also known as arrows or edges) such that all directed paths in the diagram with the same endpoints lead to the same result by composition. That is, when picking two objects, one can follow any path through the diagram and obtain the same result by composition. For example, the triangle

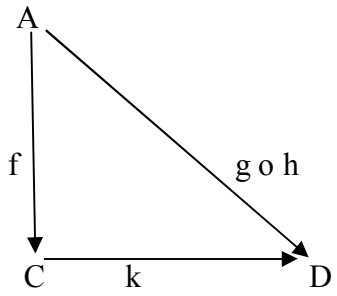


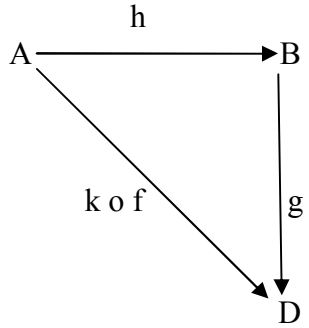
Commutes if and only if h = g o f.

Also the square below commutes if k o f = g o h.



In fact any commutative diagram (unless it involves an empty path) can be replaced by a set of commutative triangles. For example, the square above commutes if and only if the two diagrams





Commute (in fact if and only if either one does).

Composing maps is like following directed paths from one object to another (e.g., from set to set).

In general, a diagram is commutative if and only if any two paths along arrows that start at the same point finish at the same point yield the same homomorphism via

compositions along successive arrows. Commutativity of the whole diagram follows from commutativity of its triangular components.

3.4 Algebraic operations on functions

Recalling the arrow notation for functions between sets, a function f with domain X and codomain Y is denoted f: X $\rightarrow$ Y. The foundational operation on functions is composition; if f: X $\rightarrow$ Y and g: Y $\rightarrow$ Z, then we can define gof: X $\rightarrow$ Z by gof(x):= g(f(x)). For the composition to be defined, the codomain of f must be the same as the domain of g. More over, for each set X there is an identity function on X denoted id_x: X $\rightarrow$ X defined by $id_x(x)$: = x. These operations are governed by the associativity and unit laws viz., for f: X $\rightarrow$ Y, g: Y $\rightarrow$ Z and h: Z $\rightarrow$ W, we have (h o g) o f = h o (g o f), and f o id_x = f = id_y o f.

Notice that these equations are formulated purely in terms of the algebraic operations on functions without any reference to the elements of the sets X, Y, Z and W. We can refer to any concept pertaining to functions which can be defined purely in terms of composition and identities as *arrow-theoretic.* This leads us to begin to learn to think with arrows by seeing how we can replace some familiar definitions couched in terms of elements by their corresponding arrow-theoretic equivalents. In turn, it works as a basic notion towards describing the notion of a category.

CHAPTER FOUR

FUNDAMENTALS OF CATEGORY THEORY

4.1 Introduction

Category theory is an algebraic structure consisting of a collection of *objects* linked together by a collection of *arrows* that have two basic properties; the ability to compose the arrows associatively and the existence of an identity arrow for each object. Objects and arrows may be abstract entities of any kind. The theory generalizes all of mathematics in terms of objects and arrows independent of what the objects and arrows represent.

Category theory is based on the idea that the structure of a class of mathematical objects can be best understood by analyzing (instead of the objects themselves) the functions between them. Consider as an example, groups. A group is a set of elements endowed with a binary operation o, a urinary operation $^{-1}$ (inversing) and a special element ℓ such that the well known properties are satisfied. If we take a set of elements S, we can consider it either as a set or as a group, depending on whether we assume that the above mentioned operations are defined or not.

Category theory takes different view of the problem. Let **G** be the class of all groups. A mapping f: $\mathbf{G} \rightarrow \mathbf{G}$ is a function between elements of **G** considered as sets with the additional requirement that the following properties be satisfied.

$f(a \circ b) = f(a) \circ f(b)$, $f(\ell) = \ell$, $f(a^{-1}) = f(a)^{-1}$. This characterization of the functions that transform a group in to another is enough to characterize the structure of the groups themselves; a group is simply seen as a set belonging to a class for which functions like this are the morphisms. This example shows the basic idea of category theory; the structure of a certain class of a mathematical object can be studied conveniently by studying the properties of the functions that transform a member of the class in to another.

4.2 Definition of Category

A Category **C** consists of the following

(i) A Class Ob(**C**) of *objects*.

(ii) A class hom(**C**) of *morphisms or maps or arrows*.

(iii) A binary operation called *composition of morphisms*, such that for any three objects A, B and C, we have hom(B, C) x hom(A, B) $\rightarrow$ hom(A, C), (i.e., the composite of f: A $\rightarrow$ B and g: B $\rightarrow$ C is written g o f: A $\rightarrow$ C or gf: A $\rightarrow$ C), governed by two axioms

(a) Associativity: if f: A $\rightarrow$ B, g: B $\rightarrow$ C and h: C $\rightarrow$ D then h o (g o f) = (h o g) o f, and

(b) Identity: For every object X, there exists a morphism $1_X : X \rightarrow X$ called the *identity* morphism for X, such that for every morphism f: A $\rightarrow$ B, we have 1_B o f = f = f o 1_A.

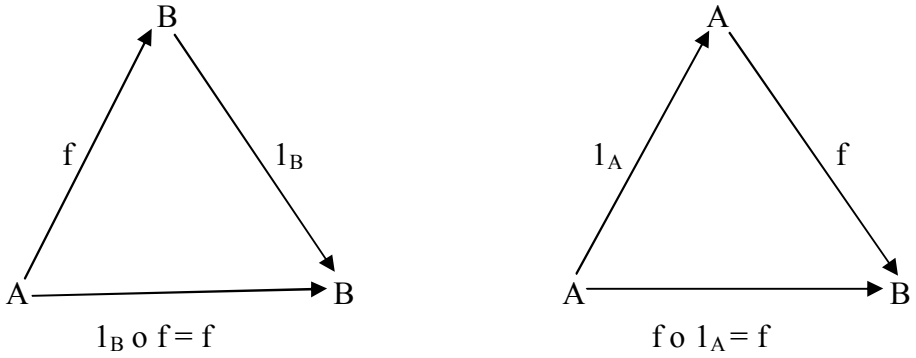


FIG 1 Identity axioms

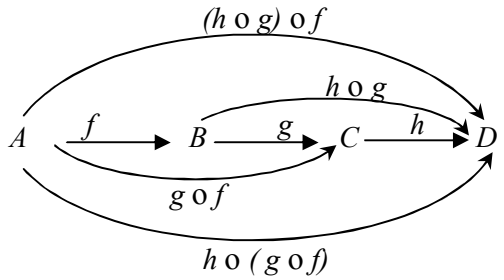


FIG 2 Associativity axiom

The type of objects depends upon a chosen mathematical structure. For example, in set theory the objects are sets, in group theory the objects are groups and in graph theory the objects are graphs.

Morphisms are structure-preserving maps. It is an abstraction derived from structure preserving mappings between two mathematical structures. Morphism is an arrow linking an object called the *domain* to another object called the *codomain*. The notion of

morphism can connotes differently depending on the type of space chosen. In set theory morphisms are functions, in group theory they are group homomorphisms and in graph theory they are graph homomorphisms.

A morphism has two parts: the domain (or source) and the codomain (or target). If a morphism f has domain A and codomain B, we write f: A $\rightarrow$ B and say that *f is* a *morphism from A to B*. Thus, morphism is represented by an arrow from its domain to its codomain. The collection (i.e., *hom-Class*) of all morphisms from A to B is denoted *hom(A, B)* (or *Hom(A, B)* or $hom_C(A, B)$ or *Mor(A, B),* or *C(A, B))*. If these collections are sets, then we call these *hom-sets* and say that the *category is enriched in sets*.

Note that, if the category **C** is enriched in sets, composition can be described as a family of maps, one for each triple of objects in **C**.

Proposition 1: There is exactly one identity morphism for every object.

Proof: Suppose A is an object of a category **C**, 1_A: A $\rightarrow$ A (i.e., $1_A \in \mathbf{C}(A, A)$) is its identity morphism. If $i_A \in \mathbf{C}(A, A)$ is another morphism satisfying the identity axiom, then $1_A = 1_A \circ i_A = i_A$.

From the above proposition, by replacing Ob(**C**) by identity morphisms, a category **C** can alternatively be defined entirely in terms of morphisms (identity morphisms) since each object A of **C** is uniquely determined by the identity morphism 1_A. In this case the objects are actually redundant structure and their role can be replaced by the identity maps (see [Cockett (2010)], for details).

4.3 Examples of categories

(i) The simplest category is the *empty category*; it has no object and no arrow.

(ii) The simplest nonempty category is the category with one object and exactly one arrow, denoted **1**. The one arrow is actually forced to be the identity map on the one object.

(iii) The category **2** has two objects, their identity arrow and one non-identity arrow as illustrated below.



Category **1** Category **2**

FIG 3 Categories **1** and **2**

(iv) A preordered (a preorder is a set X together with a binary relation α which is reflexive (for all $x \in X$, $x \alpha x$) and transitive (for all $x, y, z \in X$, $x \alpha y$ and $y \alpha z$ implies $x \alpha z$)) set (X, α) is a category $C(X, \alpha)$ or $C(X)$. The objects of $C(X, \alpha)$ are the elements of X. If $x, y \in X$ and $x \alpha y$, then $C(X, \alpha)$ has exactly one arrow from x to y. If x is not related to y by α, then there is no arrow from x to y. Notice that the composability of the arrows is guaranteed by the transitivity of the preorder i.e., if $x \alpha y \wedge y \alpha z \Rightarrow x \alpha z$. Moreover, the identity arrows of $C(X, \alpha)$ are those of the form (x, x) guaranteed by the reflexivity.

In other words, any preordered set (P, ≤) forms a *small* category (see 4.4). In this, the objects are the members of P, the morphisms are arrows pointing from x to y when $x \leq y$, between any two objects there can be at most one morphism. The existence of identity morphism and the composability of the morphisms are guaranteed by the reflexivity and the transitivity of the preorder. For example, for any three objects a, b, c in P having the morphisms $f : a \leq b$ and $g : b \leq c$ we have $g \circ f : a \leq c$ for any preordered set.

(v) Any monoid forms a small category with a single object A (here A is any fixed set). The morphisms from A to A are precisely the elements of the monoid, the identity morphism of A is the identity of the monoid, and the categorical composition of morphisms corresponds to multiplication of monoid elements. Hence the notion of category generalizes those of preorder and monoid (see [Marquis (2010)], for details).

Notice that when there is at most one arrow between any two objects, the value of the composite of any two maps is forced upon. Thus, it is simply a matter of whether maps exist between objects or not. Thus, we may view such a category as a *relation* on the objects.

(vi) Any partially ordered set (P, ≤) can be interpreted as a category; the objects are the elements of P, if $x, y \in P$ then there is exactly one arrow from x to y if and only if $x \leq y$; it is empty otherwise. Transitivity ensures that composition is defined and reflexivity ensures that identity exists. In fact, we do not need the antisymmetry axiom so that this construction works for preorders as explained earlier (see [Laan (2003)], for details).

(vii) Posets with monotone functions forms a category of partially ordered set, denoted **Pos**. The objects are posets, the morphisms are the monotone functions. The identity

function on a poset is clearly monotone because, if $(P, \leq)$ is a poset, then $f: P \to P$ implies $\forall x \in P, x \leq x \Rightarrow f(x) \leq f(x)$. The composite of two monotone functions is easily seen to be monotone; if (S, α), (T, β) and (X, λ) are posets such that $f: S \to T$, $g: T \to X$ are monotone, then $g \circ f: S \to X$ is monotone such that $\forall x, y \in S$, $x \alpha y$ in S, $(g \circ f)(x) \lambda (g \circ f)(y)$ in X.

(viii) The class of all preordered sets with monotonic functions forms a category, **Ord**. The category **Ord** has preordered sets as objects and monotone functions as morphisms. It is a category as the composition of two monotonic functions is monotonic and the identity map is monotonic.

(ix) Every Directed graph can be made into a category by taking the nodes as objects, and all other paths as morphisms typed with their start and end nodes. Composition is concatenation of paths, and the identities are the empty paths.

(x) The class of all sets together with all functions between sets, where composition is the usual function composition, forms a category denoted **Set**. In other words, the category of sets denoted **Set** is a category whose objects are sets, morphisms from A to B are the set-theoretic functions from A to B and where composition is given by $(g \circ f)(x) = g(f(x))$.

It is the most basic and most commonly used category in mathematics.

(xi) The category of finite sets **Fin** has finite sets as objects and functions between finite sets as morphisms.

(xii) The category of sets and partial functions **Pfn** (Note that, a partial function from a set A to a set B is a function with domain A_0 and codomain B, where A_0 is some subset of A) has sets as objects and partial functions as arrows. (If $f: A \to B$ and $g: B \to C$ are partial functions with f defined on $A_0 \subseteq A$ and g defined on $B_0 \subseteq B$, the composite

g o f: $A \to C$ is a partial function from A to C defined on the subset $\{x \in A_0 / f(x) \in B_0\}$ of A by the requirement $(g \text{ o } f)(x) = g(f(x))$.

(xiii) The category of relations, denoted **Rel,** has sets as objects and relations as arrows. Composition of R: $A \to B$ and S: $B \to C$ is $S \text{ o } R = \{(a, c) / \exists b\ (a, b) \in R \wedge (b, c) \in S\}$, identities: $1_a = \{(a, a) / a \in A\}$.

(xiv) The category of graphs, denoted **Grf,** has graphs as objects and homomorphisms of graphs as arrows.

Note that a homomorphism ϕ from a graph G to a graph H, denoted $\phi: G \to H$, is a pair of functions $\phi_0: G_0 \to H_0$ and $\phi_1: G_1 \to H_1$ with the property that if $u: m \to n$ is an arrow of G, then $\phi_1(u): \phi_0(m) \to \phi_0(n)$ in H.

Suppose $\phi: G \to H$ and $\psi: H \to K$ are graph homomorphisms, and suppose that $u: m \to n$ in G, then by definition $\phi_1(u): \phi_0(m) \to \phi_0(n)$ in H and so by definition $\psi_1(\phi_1(u)): \psi_0(\phi_0(m)) \to \psi_0(\phi_0(n))$ in K. Hence $\psi \text{ o } \phi$ is a graph homomorphism.

The identity homomorphism id_G is the identity function for both nodes and arrows.

(xv) The category of groups, denoted **Grp,** has groups as objects and group homomorphisms as arrows.

(xvi) The category of abelian groups, denoted **AbGrp,** is a category having abelian groups as objects and group homomorphisms as arrows.

(xvii) The category of rings, denoted **Rng,** has rings as objects and ring homomorphisms as morphisms.

(xviii) The category of topological spaces, denoted **Top,** is a category with topological spaces as objects and continuous functions as morphisms.

(xix) Functional programming language as a category

A functional programming language could be described as one that gives the user some *primitive types, operations* and some *constructors* from which one can produce more complicated types and operations. One writes a program by applying constructors to the types, constants and functions. *Running* a program consists of applying such an operator to constants of the input type to obtain a value. Notice that a functional programming language has:

(a) Primitive data type, given in the language.

(b) Constant of each type.

(c) Operations, which are functions between the types.

(d) Constructors, which can be applied to data types and operations to produce derived data types and operations of the language.

The language consists of the set of all operations and types derivable from the primitive data types and primitive operations. (By primitive we mean given in the definition of the language rather than constructed by a constructor).

In order to model a functional programming language (**L**) as a category, we make two *assumptions* about (**L**) and *one innocuous change*:

(A1) There exists a *do-nothing operation* id_A for each type A (Primitive and Constructed). It does nothing to the data.

(A2) An additional type **1**, which has the property that from every type A there is a unique operation to **1,** is added to the language. We interpret each constant c of type A as an arrow c: $\mathbf{1} \rightarrow A$. This incorporates the constants into the set of operations; they no longer appear as separate data.

(A3) The language is supposed to have composition constructor: an operation f that takes something of type A as input and produces something of type B, and another operation g that has input of type B and output of type C; then doing one after the other is a derived operation (program) typically denoted g;f, which has input of type A and output of type C.

Generally, functional programming languages have *do-nothing* operation and composition constructors i.e., A1 and A3 fit the concept as it appears. The language resulting from the change in A2 is operationally equivalent to the original language. Composition must be associative in the sense that if either of (h;g);f or h;(g;f) is defined, so is the other and they are the same operation. Also, we require for f: A $\rightarrow$ B, that id_B;f and f;id_A are defined and are the same operation as f i.e., we impose id_B;f = f;id_A = f on the language. Note that both the requirements are reasonable, since in any implementation, the two operations would be required to be the same in order to do the same thing.

Under those conditions, a functional programming language (**L**) has a category structure **C**(**L**) for which:

(a) The types of (**L**) are the objects.

(b) The operations (primitive and derived) of (**L**) are the arrows of **C**(**L**).

(c) The source and target of an arrow are the input and output types of the corresponding operation.

(d) Composition is given by the composition constructor written in the reverse order.

(e) The identity arrows are the do-nothing operations.

The aforesaid examples of categories illustrate how category theory organizes and unifies much of mathematics. It is also clear from various examples of categories given above that a category is mainly characterized by its morphisms and not by its objects. It is to be noted that, not necessarily the objects or the arrows of a category form a set. This is because some categories are too large for either their objects or their arrows to form sets.

4.4 Types of categories

(i) **Discrete category**

A category **C** is said to be discrete if every morphism of **C** is an identity.

For example, any set S can be regarded as a discrete category where the ob(S) is the elements of S and the morphisms of S can be regarded as the identity arrows $x \to x$ for every $x \in S$.

Discrete categories are the simplest kind of category.

(ii) **Small and large categories**

A category **C** is called *small* if both ob(**C**) and hom(**C**) are actually sets (small sets) and not proper classes, and large otherwise. A locally small category is a category such that for all objects A and B, the hom-class hom(A, B) is a set, called a *hom-set*. Many categories in mathematics, such as category of sets, although not small are at least locally small. While preorder and monoid are examples of small categories. **Set** is an example of a large category.

(iii) **Concrete category**

A category is called *concrete* if its objects are sets (usually with structure), morphisms are mappings (preserving that structure), composition of morphisms are composition of mappings and identity morphisms are identity mappings. For example, **Set, Grp, Ord** and **Pos** are concrete categories.

4.5 Consructions on categories

Various constructions can be performed on categories as can be performed in other branches of abstract algebra like rings or group theory.

(1) **Subcategory**

Let **C** be a category.

Now, **D** is a subcategory of **C** if:

(i) All objects of **D** are objects of **C** and all arrows of **D** are arrows of **C**.

(ii) The source and target maps for **D** are the restrictions of those for **C**. It follows that for any two objects A and B of **D**, $\hom_D (A, B) \subseteq \hom_C (A, B)$.

(iii) For any object A in **D**, 1_A is the same in Mor(**D**) and Mor(**C**).

(iv) If $f : A \to B$ and $g : B \to C$ in **D**, then the composite g o f (in **C**) is in **D** and is the composite in **D** i.e., the composite of any two morphisms in **D** is the same as their composite in **C**.

Moreover, a subcategory **D** of **C** is said to be full if **D**(A, B) = **C**(A, B) for all A,B in **D**. Thus **Mon** is a full subcategory of **Grp**, **Fin** is a full subcategory of **Set**, but **Set** is not a full subcategory of sets and partial functions. The category of sets and partial functions **Par** is a subcategory of **Rel** where the objects are sets i.e., same objects with **Rel**, but the

arrows are only those relations which are *deterministic* i.e., (a, b) and (a, b′) $\Rightarrow$ b = b′. The category of partial functions **Par** has the category of functions as its subcategory, having the same objects but with arrows restricted to those deterministic relations which are total. (A relation R: A $\rightarrow$ B is total if for every a$\in$A, $\exists$b$\in$B such (a, b) $\in$R.) Notice that neither of these subcategories is a full-subcategory since the set of maps between objects (in most cases) is strictly reduced.

(2) **Product category**

Let $C_1,C_2,\ldots,C_n$ be categories. The product category of $C_1,C_2,\ldots,C_n$, written $C_1xC_2x\ldots xC_n$ is defined as follows: $ob(C_1xC_2x\ldots xC_n) = \{(A_1,\ldots,A_n)/\ A_i\in C_i\}$

$Mor(C_1xC_2x\ldots xC_n) = MorC_1xMorC_2x\ldots xMorC_n = \{(\alpha_1,\alpha_2,\ldots,\alpha_n)/\alpha_i\in MorC_i\}$

where composition is defined by $(\beta_1,\beta_2,\ldots,\beta_n)(\ \alpha_1,\alpha_2,\ldots,\alpha_n) = (\beta_1\alpha_1,\beta_2\alpha_2,\ldots,\beta_n\alpha_n)$ i.e., the product is a family $(C_i)_{i\in I}$ of categories $\prod_{i\in I}\mathbf{C}_i$ obtained by putting the categories in parallel. Composition is then performed in parallel; $(\beta_i)_{i\in I}(\alpha_i)_{i\in I} = (\beta_i\alpha_i)_{i\in I}$. The empty product of categories is perfectly well defined; it has no object, the empty tuple ( ) and one arrow (the empty tuple); so it is the category **1**.

If **C** and **D** are categories, then the product category **C**x**D** is the category whose objects are ordered pairs (C, D) with C an object of **C** and D an object of **D**, and in which a morphism (f,g) : (C, D) $\rightarrow$ (C', D') is a pair of morphism f: C $\rightarrow$ C' in **C** and g: D $\rightarrow$ D' in **D**. The identity of (C, D) is (id_C, id_D). If (f',g'): (C', D') $\rightarrow$ (C", D") is another morphism then the composite (f',g') o (f,g) is defined by (f',g') o (f,g) = (f'of, g'og): (C, D) $\rightarrow$ (C", D").

(3) **The sum category**

Given a family $(\mathbf{C}_i)_{i\in I}$ of categories one can consider the disjoint union of categories $\coprod_{i\in I}\mathbf{C}_i$. This is a category whose objects (respectively maps) are the disjoint union of the

objects (respectively maps) of each $\mathbf{C}_i$. Clearly objects from different components will not be connected by any map. It is always worth considering the extremal examples; what happens when we sum the empty family of categories? Well of course, we get a perfectly good category; the empty category **0**, the one which has no arrows and no objects.

(4) **The dual category**

Like several other branches of mathematics, category theory supports a principle of duality. In the case of category theory, this arises from the invariance of the theory under the interchange of the source and target of arrows i.e., reversing arrows. Each categorical property has a dual obtained by reversing all arrows in the statement of the property. The invariance means that one proof gives two results; the theorem and its dual. Thus, given a category we may obtain a new category by keeping every thing the same except to switch the direction of the arrows. If we start with a category **C** and flip the direction of the arrows we obtain a new category written $\mathbf{C}^{\mathbf{op}}$**.** Observe that anything which is true of **C**, now holds in the dual form in $\mathbf{C}^{\mathbf{op}}$**.** Thus, if we prove a result there is always another result obtained by reversing the sense of the arrows which will also be true. Often the prefix *co* (as in coequalizer, coproduct) is an indication that this is the dual concept to that with this prefix removed. Given any category **C**, you can construct another category called *its dual* denoted $\mathbf{C}^{op}$ by reversing all its arrows. Formally, Let **C** be a category its dual or opposite $\mathbf{C}^{op}$ is the category where;

(i) Its objects and arrows are those of **C**.

(ii) If f: A $\rightarrow$ B is in **C**, then f: B $\rightarrow$ A is in $\mathbf{C}^{op}$ i.e., the source and target has been reversed.

(iii) If h = g o f in **C**, then h = f o g in $\mathbf{C}^{op}$.

(iv) The identity arrows are the same in both categories **C** and $\mathbf{C}^{op}$.

As an example, the category of Relations **Rel** has $\mathbf{Rel}^{OP}$ obtained by reversing the ordered pairs.

4.6 Properties of morphisms

Monomorphism and split monomorphism

A morphism f: A $\rightarrow$ B in a category **C** is called

(i) a *monomorphism* if it is left cancellable i.e., for every pair of morphisms g,h: C $\rightarrow$ A if f o g = f o h $\Rightarrow$ g = h.

(b) a *split monomorphism (or section or coretraction)* if it is left invertible i.e., there exists a morphism g: B $\rightarrow$ A such that g o f = 1_A. In that case A is called a *retract* of B.

Examples:

(i) In the category **Set**, the monomorphisms are exactly the injective mappings.

(ii) In the category **Top,** the monomorphisms are exactly the injective continuous mappings.

(See [Laan (2003)], for details).

Proposition 2: Every split monomorphism is a monomorphism.

Proof: Let f: A $\rightarrow$ B be a split monomorphism, then by definition $\exists$ k: B $\rightarrow$ A such that k o f = 1_A, and let f o g = f o h for g,h: C $\rightarrow$ A. We are to show that g = h, which follows as below:

g = 1_A o g = (k o f) o g = k o (f o g) = k o (f o h) = (k o f) o h = 1_A o h = h.

Propositions 3: (i) Every identity morphism is a split monomorphism; and (ii) Composite of two monomorphisms is a monomorphism.

Proof: (i) Let **C** be a category. 1_A: A $\rightarrow$ A is a split monomorphism if there exists a morphism, say k: A $\rightarrow$ A such that k o $1_A = 1_A$. But as for every A$\in$**C**, $1_A = 1_A$ o 1_A, the result follows.

(ii) Let f: A $\rightarrow$ B and k: B $\rightarrow$ D be monomorphisms. We are to show that if (k o f) o g = (k o f) o h for g,h: C$\rightarrow$ A $\Rightarrow$ g = h i.e., their composite k o f is a monomorphism.

We have, (k o f) o g = (k o f) o h $\Rightarrow$ k o (f o g) = k o (f o h) and since k is a monomorphism $\Rightarrow$ f o g = f o h. Again, since f is a monomorphism $\Rightarrow$ g = h. Hence k o f is a monomorphism.

Propositions 4: (i) If the composite k o f of two morphisms is a monomorphism, then f is a monomorphism; and

(ii) Composite of two split monomorphisms is a split monomorphism.

Proof: (i) Let f: A $\rightarrow$ B and k: B $\rightarrow$ D be morphisms and k o f be a monomorphism. Suppose f o g = f o h for g,h: C $\rightarrow$ A. We are to show that g = h.

We have, (k o f) o g = k o (f o g) = k o (f o h) = (k o f) o h $\Rightarrow$ g = h. Hence f is a monomorphism.

(ii) Let f: A $\rightarrow$ B and k: B $\rightarrow$ D be two split monomorphisms i.e., by definition $\exists$ t: B $\rightarrow$ A and s: D $\rightarrow$ B such that t o f = 1_A, s o k = 1_B. We are to show that k o f, the composite of f and k, is a split monomorphism i.e., $\exists$t o s: D $\rightarrow$ A such that (t o s) o (k o f) = 1_A.

We have, (t o s) o (k o f) = t o (s o k) o f = t o 1_B o f = t o f = 1_A. Hence k o f is a split monomorphism.

Proposition 5: If the composite k o f of two morphisms is a split monomorphism, then f is a split monomorphism.

Proof: Let f: A $\to$ B and k: B $\to$ D be morphisms and k o f be a split monomorphism i.e., $\exists$ s: D $\to$ A such that s o (k o f) = 1_A. Since, s o (k o f) = (s o k) o f = $1_A \Rightarrow$ f is a split monomorphism.

Epimorphism and split epimorphism

A morphism f: A $\to$ B in a category **C** is called

(i) an *epimorphism* if it is right cancellable i.e., for every pair of morphisms g,h: B $\to$ C, g o f = h o f $\Rightarrow$ g = h;

(ii) a *split epimorphism (or retraction)* if it is right invertible i.e., there exists a morphism g: B $\to$ A such that f o g = 1_B.

Examples:

(i) In **Set,** the epimorphisms are exactly the surjective mappings.

(ii) In **Top,** the epimorphisms are exactly the surjective continuous mappings.

It is clear that every identity morphism is a split epimorphism, because for every B$\in$**C**, $1_B = 1_B$ o 1_B.

Proposition 6: Every split epimorphism is an epimorphism.

Proof: Let f: A $\to$ B be a split epimorphism i.e., $\exists$ k: B $\to$ A such that f o k = 1_B.

Suppose f o k = 1_B and g o f = h o f, we are to show that g = h.

We have, g = g o 1_B = g o (f o k) = (g o f) o k = (h o f) o k = h o (f o k) = h o 1_B = h.

Hence, f is an epimorphism.

Proposition 7: The composite of two epimorphisms is an epimorphism.

Proof: Let f: A $\to$ B, k: B $\to$ C be epimorphisms and g o (k o f) = h o (k o f), we are to show that g = h for every pair g,h: C $\to$ D.

We have, g o (k o f) = h o (k o f) $\Rightarrow$ (g o k) o f = (h o k) o f and since f is an epimorphism, g o k = h o k. Moreover, g o k = h o k $\Rightarrow$ g = h since k is an epimorphism.

Proposition 8: The composite of two split epimorphisms is a split epimorphism.

Proof: Let f: A $\rightarrow$ B, k: B $\rightarrow$ D be split epimorphisms i.e., $\exists$ s: B $\rightarrow$ A, t: D $\rightarrow$ B such that f o s = 1_B, k o t = 1_D. We are to show that k o f is a split epimorphism i.e., (k o f) o (s o t) is the identity in D.

We have, (k o f) o (s o t) = k o (f o s) o t = k o 1_B o t = k o t = 1_D.

Proposition 9: If the composite of morphisms f and k is a split epimorphism, then k is a split epimorphism.

Proof: Suppose k o f: A $\rightarrow$ D is a split epimorphism i.e., $\exists$ s: D $\rightarrow$ A such that (k o f) o s = 1_D. Since (k o f) o s = k o (f o s) = 1_D k is a split epimorphism.

Bimorphism

A morphism f: A $\rightarrow$ B is called a *bimorphism* if it is both a monomorphism and an epimorphism.

It is clear that every identity morphism is a bimorphism since, as shown earlier, every identity morphism is a monomorphism as well as an epimorphism and hence a bimorphism.

Proposition 10: The composite of two bimorphisms is a bimorphism.

Proof: Let f: A $\rightarrow$ B, k: B $\rightarrow$ D be bimorphisms i.e., f o g = f o h $\Rightarrow$ g = h for g,h: C $\rightarrow$ A and g o f = h o f $\Rightarrow$ g = h for g,h: B $\rightarrow$ X. Also, k o g = k o h $\Rightarrow$ g = h for g,h: C $\rightarrow$ B and g o k = h o k $\Rightarrow$ g = h for g,h: D $\rightarrow$ X. We are to show that (k o f) o g = (k o f) o h $\Rightarrow$ g = h for g,h: C $\rightarrow$ A and g o (k o f) = h o (k o f) $\Rightarrow$ g = h for g,h: D $\rightarrow$ X.

We have, (k o f) o g = k o (f o g) = k o (f o h), and since k is a bimorphism $\Rightarrow$ f o g = f o h. Also, f o g = f o h $\Rightarrow$ g = h, since f is a bimorphism.

Moreover, g o (k o f) = (g o k) o f = (h o k) o f $\Rightarrow$ g o k = h o k since f is a bimorphism and g o k = h o k $\Rightarrow$ g = h since k is a bimorphism. Hence, k o f is a bimorphism.

Isormorphism

A morphism f: A $\rightarrow$ B is called an *isomorphism* if it is both split monomorphism and split epimorphism i.e., if it is invertible.

Two objects A and B of a category **C** are said to be isomorphic if there exists an isomorphism f: A $\rightarrow$ B. We write A $\cong$ B.

For example, in **set**, a morphism is a coretraction (resp. retraction) if and only if it is injective (resp. surjective). Thus f is an isormorphism in **set** if and only if it is bijective.

A category **C** is called a *groupoid* if every arrow of **C** is an isomorphism i.e., a category in which every arrow is invertible. In other words, a groupoid is a category in which every arrow is invertible.

Proposition 11: If f: A $\rightarrow$ B has an inverse, it has only one i.e., the inverse of an arrow in a category is unique.

Proof: Let f: A $\rightarrow$ B be a morphism in a category and suppose g: B $\rightarrow$ A and h: B $\rightarrow$ A has the property g o f = h o f = 1_A and f o g = f o h = 1_B. We are to show that g = h. We have, g = g o 1_B = g o (f o h) = (g o f) o h = 1_A o h = h.

Proposition 12: Every identity morphism is an isomorphism and the composite of two isomorphism is an isomorphism.

Proof: Suppose f is identity morphism i.e., f: A $\rightarrow$ A. For f to be an isomorphism, we need to have $\exists$ g: A $\rightarrow$ A such that g o f = 1_A and f o g = 1_A which holds as g o f = f o g = 1_A.

Suppose f: A $\rightarrow$ B, k: B $\rightarrow$ D are isomorphisms i.e., $\exists$ s: B $\rightarrow$ A such that s o f = 1_A, f o s =1_B and $\exists$t: D $\rightarrow$ B such that t o k = 1_B, k o t = 1_D. Now, to show that k o f: A $\rightarrow$ D is an isomorphism we are to show that $\exists$ g: D $\rightarrow$ A such that g o (k o f) = 1_A and (k o f) o g = 1_D. We have, g o (k o f) = (g o k) o f = 1_A (from g o k: B $\rightarrow$ A, f: A $\rightarrow$ B $\Rightarrow$ (g o k) o f = 1_A), also, (k o f) o g = k o (f o g) = 1_D (from k: B $\rightarrow$ D, f o g: D $\rightarrow$ B $\Rightarrow$ k o (f o g) = 1_D).

Proposition 13: If f and g o f are isomorphisms then g is an isomorphism.

Proof: Let f: A $\rightarrow$ B, g: B $\rightarrow$ D where f is an isomorphism i.e., $\exists$ k: B $\rightarrow$ A such that k o f = 1_A, f o k = 1_B. Suppose g o f: A $\rightarrow$ D is an isomorphism $\Rightarrow$ $\exists$ s: D $\rightarrow$ A such that s o (g o f) = 1_A, (g o f) o s = 1_D. Now, to show that g is an isomorphism, we are to show that $\exists$t: D $\rightarrow$ B such that t o g = 1_B, g o t = 1_D.

We have, (g o f) o s = g o (f o s) = g o t = 1_D (from f o s = t). Also, f o k = f o (s o g) = (f o s) o g = t o g = 1_B (from k = s o g). Hence g is an isomorphism.

Proposition 14: If any of f, g and f o g are isomorphisms, so is the third.

Proof: Let f: A $\rightarrow$ B, g: B $\rightarrow$ D and g o f: A $\rightarrow$ D and suppose f,g are isomorphisms (i.e., $\exists$ h: B $\rightarrow$ A such that h o f = 1_A and f o h = 1_B, $\exists$ n: D $\rightarrow$ B such that n o g = 1_B and g o n = 1_D). We are to show that g o f is an isomorphism i.e., $\exists$ K: D $\rightarrow$ A such that k o (g o f) = 1_A and (g o f) o k = 1_D.

We have, k o (g o f) = (k o g) o f = h o f = 1_A and (g o f) o k = g o (f o k) = g o n = 1_D.

Let f and g o f be isomorphisms, we are to show that g is an isomorphism i.e., $\exists$ n: D $\rightarrow$ B such that n o g =1_B and g o n = 1_D.

We have, (g o f) o k = g o (f o k) = g o n = 1_D and f o h = f o (k o g) = (f o k) o g = n o g = 1_B.

Suppose g, g o f are isomorphisms, we are to show that f is an isomorphism i.e., $\exists$ h: B $\rightarrow$ A such that h o f = 1_A and f o h = 1_B.

We have, k o (g o f) = (k o g) o f = h o f = 1_A, and n o g = n o (g o (f o h)) = (n o g) o (f o h) = 1_B o (f o h) = f o h = 1_B.

Proposition 15: A morphism is an isomorphism if it is a monomorphism and a retraction (split epimorphism).

Proof: Let f: A $\rightarrow$ B be a monomorphism and a retraction. For f to be an isomorphism it has to have both right and left inverses. Since f is a retraction it has a right inverse i.e., for k: B $\rightarrow$ A, f o k =1_B. It remains to show that f has a left inverse.

We have, (f o k) o f = 1_B o f, and (f o k) o f = f o (k o f) = f o 1_A i.e., f o (k o f) = f o 1_A. Since f is a monomorphism (i.e., left cancellable) $\Rightarrow$ k o f = 1_A. Hence f is an isomorphism.

Proposition 16: Every isomorphism is a bimorphism.

Proof: Let f: A $\rightarrow$ B be an isomorphism i.e., $\exists$ k: B $\rightarrow$ A such that k o f = 1_A, f o k = 1_B and suppose f o g = f o h for g,h: C $\rightarrow$ A , g o f = h o f for g,h: B $\rightarrow$ C, we are to show that g = h in both cases.

We have, g = 1_A o g = (k o f) o g = k o (f o g) = k o (f o h) = (k o f) o h = 1_A o h = h i.e., g = h

Also, g = g o 1_B = g o (f o k) = (g o f) o k = (h o f) o k = h o (f o k) = h o 1_B = h i.e., g = h

Hence f is a bimorphism.

As a typical example of isomorphism, any group can be seen as a category with a single object in which every morphism is invertible.

As defined in **4.3**, any partially ordered set (poset) can be regarded as a category. In such a category, every morphism is a bimorphism since between any two objects there is at most one morphism. Also, isomorphisms are only the identity morphisms.

A category in which every bimorphism is an isomorphism is said to be *balanced.* For example, **Set** and **Grp** are balanced.

Endomorphism

A morphism f: A $\rightarrow$ A is called an *endomorphism* of A. An invertible endomorphism is called an *automorphism*.

For example, any identity morphism in a category is an endomorphism (i.e., an identity map, also called *endo-map,* is an example of a map which starts and ends at the same place). A category can have and, often, will have many endo-maps. More generally, a category can have many different parallel maps from one object A to another object B, these maps can be collected in to a *hom-object* which, in this case, is a set denoted by **C**(A, B) as explained in **4.2**.

Idempotent

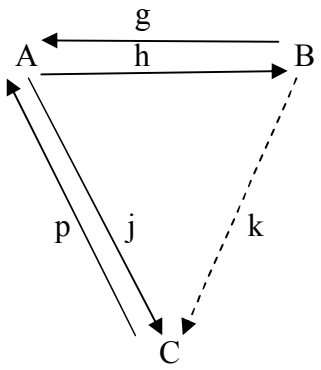
An endo-map e: A $\rightarrow$ A is idempotent if e o e = e.

Notice that if h: A $\rightarrow$ B is a retraction with right inverse g: B $\rightarrow$ A, then g o h is an idempotent. Since (g o h) o (g o h) = g o (h o g) o h = g o 1_B o h = g o h.

An idempotent e is split if there is a retraction h with right inverse g such that e = g o h. It is worthy to note that while an idempotent may be split in many different ways, there is

an important sense in which there is essentially just one splitting. It is demonstrated in the following proposition below:

Proposition 17: Suppose e: A $\rightarrow$ A is idempotent and h: A $\rightarrow$ B has right inverse g: B $\rightarrow$ A and j: A $\rightarrow$ C has right inverse p: C $\rightarrow$ A, with e = g o h = p o j; then there is a unique isomorphism k: B $\rightarrow$ C such that k o h = j and p o k = g.



k o h = j and p o k = g.

FIG 4 Uniqueness of splitting of an idempotent

Proof: Set k = j o g. As k^{-1} = h o p, k is an isomorphism since (j o g) o (h o p) = j o (g o h) o p = j o e o p = j o p = 1_C. Similarly, for the other composite, (h o p) o (j o g) = h o e o g = 1_B.

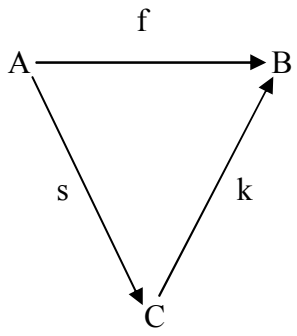
Suppose k′ also satisfies k′ o h = j, then k′ o h = k o h. As h is a retraction, therefore epimorphism, it follows that k = k′.

We can conclude that the splitting of an idempotent is unique up to unique isomorphism. It is certainly not the case that idempotent will generally split in a category; however, if it does, it is unique.

Right and left equivalent

Two morphisms f: A $\rightarrow$ B, k: C $\rightarrow$ B are said to be right equivalent if there exists an isomorphism s: A $\rightarrow$ C such that f = k o s. Dually, two morphisms f: A $\rightarrow$ B, k: A $\rightarrow$ C are said to be left equivalent if there exists

an isomorphism s: C $\rightarrow$ B such that f = s o k.



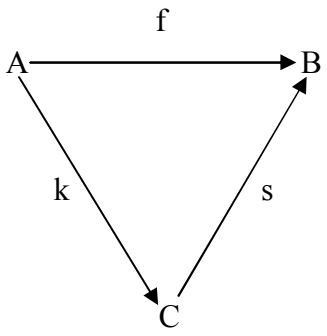


FIG 5 Right equivalence Left equivalence

Proposition 18: If f: A $\rightarrow$ B is a monomorphism, then so is its right equivalent.

Proof: If f: A $\rightarrow$ B is a morphism, k: C $\rightarrow$ B is its right equivalent if $\exists$ an isomorphism s: A $\rightarrow$ C such that f = k o s. Since s: A $\rightarrow$ C is an isomorphism $\Rightarrow \exists$ t: C $\rightarrow$ A such that t o s = 1_A, s o t = 1_C.

Let f: A $\rightarrow$ B be a monomorphism and k o g = k o h for g,h: X $\rightarrow$ C. In order to show that k is a monomorphism, we need to show that g = h. Since f o t = k; from k o g = k o h, we have (f o t) o g = (f o t) o h $\Rightarrow$ f o (t o g) = f o (t o h). Since f is a monomorphism $\Rightarrow$ t o g = t o h. Thus, it follows that g = h, since t is left cancellable being an isomorphism.

Proposition 19: The left equivalent of an epimorphism is an epimorphism.

Proof: A morphism f: A $\rightarrow$ B has k: A $\rightarrow$ C as its left equivalent if $\exists$ an isomorphism s: C $\rightarrow$ B such that f = s o k. Now, s: C $\rightarrow$ B being an isomorphism $\Rightarrow \exists$ t: B $\rightarrow$ C such that s o t = 1_B, t o s = 1_C.

Suppose f: A $\rightarrow$ B is an epimorphism, and g o k = h o k for g,h: C $\rightarrow$ X, we are to show that g = h.

We have, k = t o f i.e., g o (t o f) = h o (t o f) $\Rightarrow$ (g o t) o f = (h o t) o f, since f is an epimorphism $\Rightarrow$ g o t = h o t, and since t is an isomorphism $\Rightarrow$ g = h.

4.7 Properties of objects

Initial and terminal objects

An object I in a category **C** is called an *initial object* of **C** if for every object A of **C**, there exists a unique morphism I $\rightarrow$ A.

The terminal object is the dual notion of this. An object T of a category **C** is called a *terminal object* of **C** if for every object A in **C**, there exist a unique morphism A $\rightarrow$ T. Initial object is usually denoted **0** while terminal object is denoted **1.** The zero (null) object is an object that is both initial and terminal. Note that **0** and the null object are distinct. For example in **Set** and **Fin** the empty set is initial and any one-element (singleton) set is terminal. Thus the category of sets has a unique initial object but many terminal objects, and there is no zero object. In general, initial and terminal objects are not required to exist in a given category. However, if they do exist, they are essentially unique.

Proposition 20: Any two terminal (respectively, initial) objects in a category are isomorphic.

Proof:

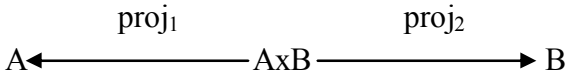
Let T and T' be terminal objects. Since T is terminal, there is an arrow f: T' $\rightarrow$ T. Similarly, there is an arrow g: T $\rightarrow$ T', since T' is terminal. The arrow f o g: T $\rightarrow$ T is an arrow with target T. Since T is a terminal object of the category, there can be only one arrow from T to T. Thus f o g must be the identity arrow of T. An analogous proof shows that g o f is the identity arrow of T'. Hence, T and T' are isomorphic.

Suppose A, A' are initial objects in a category. Since A is initial, there is an arrow f: A $\rightarrow$ A' and again, since A' is initial, there is an arrow g: A' $\rightarrow$ A. Now their composite is an arrow g o f: A $\rightarrow$ A. But 1_A: A $\rightarrow$ A is another arrow from A to itself. Hence, due to the uniqueness of the arrow from an initial object, g o f = 1_A. Likewise, f o g = $1_{A'}$. So A and A' are isomorphic. Moreover, if A is an initial object then any object isomorphic to A is also an initial object. The same is true for terminal objects.

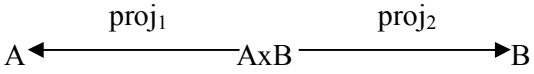
The one-element monoid is both initial and terminal in the category of monoid, thereby making it a zero object. The category of partially ordered set has an initial object if and only if P has least element and a terminal object if and only if P has a greatest element.

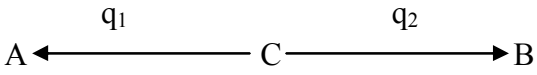
Product of two objects

Recall that the Cartesian product of two sets A and B, denoted AxB, is the set of ordered pairs with first coordinate in A and second coordinate in B i.e., AxB = {(a,b)/ a∈A and b∈B}. The coordinates in categorical terms are functions $proj_1$: AxB → A and $Proj_2$: AxB → B, called *coordinate projections* or just *projections:*

Note that products are specified by triples 

where $proj_i$'s denote projections.

Let A and B be two objects in a category **C.** A product of A and B is an object AxB together with a pair of arrows 

such that for every triple 

(i.e., for any object C and arrows q_1: C → A and q_2: C → B) there exists a unique arrow q (or < q_1,q_2>): C → AxB such that the following diagram commutes i.e., $proj_1$ o q = q_1 and $proj_2$ o q = q_2

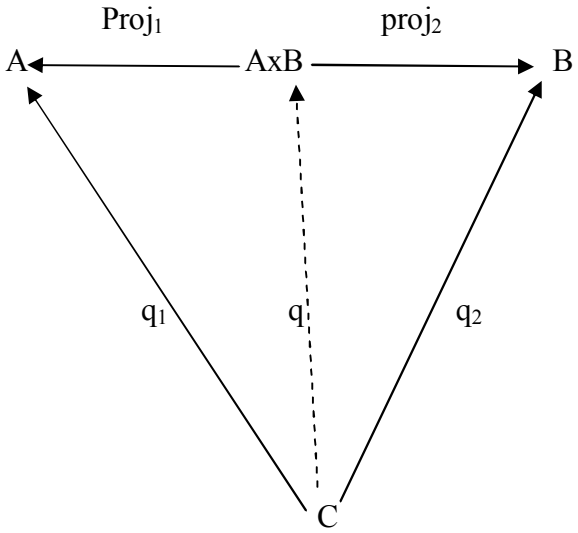


FIG 6 *Product*

Examples:

(i) In **Set**, products are the usual Cartesian products.

(ii) In a poset category, products are the greatest lower bounds.

Coproducts

Coproduct is the dual notion of product. That is, coproducts in **C** are just products in $\mathbf{C}^{\mathbf{op}}$, interpreted back in **C**.

Let A, B be objects in a category **C**. A coproduct of A and B is an object A+B together with a pair of arrows $A \xrightarrow{inj_1} A+B \xleftarrow{inj_2} B$ such that for every arrow (triple) $A \xrightarrow{f} C \xleftarrow{g} B$ there exists a unique morphism [f,g]: A+B → C such that the following diagram commutes i.e.,[f,g] o inj_1 = f and [f,g] o inj_2 = g.

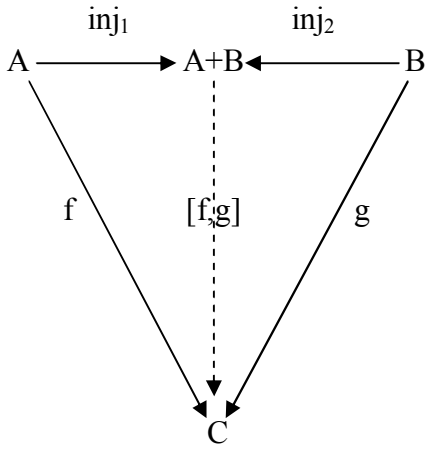


FIG 7 *Coproduct*

Here inj_i's denote injections and [f,g] denotes copairing of f and g.

Coproducts in **Set** is given by disjoint union of sets, e.g. (a⊎b) which can be equivalently define as X+Y:= {1}xX ∪ {2}xY. We can represent injections inj_1: X → X+Y, inj_2: Y → X+Y as $X \xrightarrow{Inj_1} X+Y \xleftarrow{inj_2} Y$

Or alternatively as $inj_1(x) := (1, x)$, $inj_2(y) := (2, y)$.

Also, given functions $f: X \to Z$ and $g: Y \to Z$, we can define $[f,g]: X+Y \to Z$ as $[f,g]\,(1, x) := f(x)$, $[f,g]\,(2, y) := g(y)$.

It can be observed that the aforesaid representation may suggest that coproducts allow for definition by cases.

Equalizer

If $f: A \to B$ is an arrow in a category and for some arrow $g: C \to B$ there is an arrow $h: A \to C$ for which $f = g \circ h$, we say f factors through g. The use of the word *factor* shows explicitly that categorists are to work with functions. Also, if S and T are sets and $f,g: S \to T$ are functions, it is a familiar fact that we can form the subset $Eq(f,g) \subseteq S$, consisting of all elements $s \in S$ for which $f(s) = g(s)$. These observations lead to the following definition:

Let **C** be a category and $f,g: A \to B$ be a parallel pair of arrows. An equalizer of f and g is an object E together with an arrow $j: E \to A$ such that $f \circ j = g \circ j$ and satisfying a universal mapping property: for any arrow $h: C \to A$ for which $f \circ h = g \circ h$, there is a unique arrow $k: C \to E$ such that $j \circ k = h$.

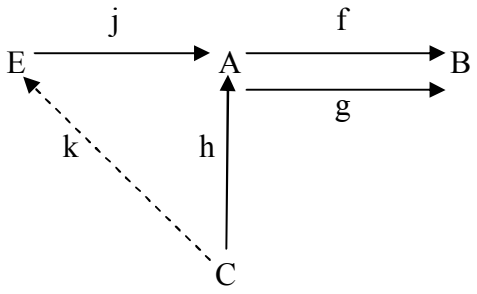


FIG 8 *Equalizer*

It is easy to see that elements *a* of A such that *fa* = *ga* are precisely those which factor through j. Usually, E is referred to as an equalizer of f and g without referring to j. Nevertheless, j is a crucial part of the data.

For example, in **Set,** an equalizer E of the function $(x, y) \mapsto x^2 + y^2$ and $(x, y) \mapsto 1$ from $\mathbf{R}x\mathbf{R} \rightarrow \mathbf{R}$ is the circle $x^2 + y^2 = 1$. The arrow j: E $\rightarrow$ **R** x **R** is the inclusion.

Coequalizer

Coequalizer is the dual notion of equalizer. Let **C** be a category with parallel arrows f, g: A $\rightarrow$ B. An arrow h: B $\rightarrow$ C is called a *coequalizer* of f and g provided h o f = h o g and for any arrow k: B $\rightarrow$ D for which k o f = k o g, there is a unique arrow t: C $\rightarrow$ D such that t o h = k.

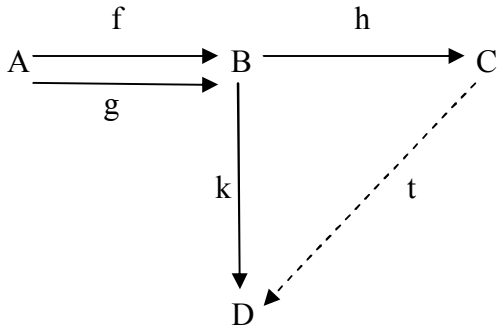


FIG 9 *Coequalizer*

For example, suppose f, g: S $\rightarrow$ T are pair of arrows in the category of sets. From above, for h: T $\rightarrow$ U to be a coequalizer, we must have h o f = h o g and that given any arrow k: T $\rightarrow$ V, the equation k o f = k o g implies the existence of a unique arrow t: U $\rightarrow$ V such that t o h = k. It entails that if such an arrow t exists, then necessarily k o f = k o g. The pair f and g determines a relation R $\subseteq$ T x T which consists of $\{f(s), g(s)/ s \in S\}$. The equation k o f = k o g is equivalent to the statement that $k(t_1) = k(t_2)$ for all $(t_1, t_2) \in R$.

4.8 Functors

As emphasized earlier, a category itself is a type of mathematical structure. Thus, we can look for processes that preserve this structure. In the literature of category theory, such a process is called a *functor* or *covariant functor*. Accordingly, a functor is a structure preserving map between categories in the same way that a homomorphism is a structure preserving map between graphs or monoids. It is a morphism of categories preserving the structure; namely, identity and composition.

Formally, a Functor (covariant functor) F: **C** $\rightarrow$ **D** is a pair of functions F_0: $\mathbf{C_0} \rightarrow \mathbf{D_0}$ and F_1: $\mathbf{C_1} \rightarrow \mathbf{D_1}$ for which the following hold:

F-1 If f: A $\rightarrow$ B in **C**, then $F_1(f)$: $F_0(A) \rightarrow F_0(B)$ in **D**. That is, if there is an arrow between objects A, B of **C;** then there is an arrow between f(A), f(B) of **D**.

F-2 For any object A of **C**, $F_1(id_A) = id_{F_0(A)}$. That is, the identity arrow of A in **C** is the identity arrow of f(A) in **D**.

F-3 If g o f is defined in **C**, then $F_1(g)$ o $F_1(f)$ is defined in **D** and $F_1(g \text{ o } f) = F_1(g)$ o $F_1(f)$.

The notation for the constituents F_0: $\mathbf{C_0} \rightarrow \mathbf{D_0}$ and F_1: $\mathbf{C_1} \rightarrow \mathbf{D_1}$ are not standard. It is to be noted that, for a category **C, C$_0$** is taken to be the collection of objects of **C** while **C$_1$** is the collection of arrows of **C** (same for **D**).

In other words, a functor from a category **A** to a category **B,** written F**: A** $\rightarrow$ **B,** assigns to each object A of **A** an object FA of **B**, and assigns to each arrow f: A $\rightarrow$ A′ of **A** an arrow Ff: FA $\rightarrow$ FA′ of **B** so that it preserves the domain and codomain and must also preserve identity arrows and composition. That is, applying F to an identity arrow 1_A in **A** gives the identity arrow on FA in **B** (i.e., $F(1_A) = 1_{FA,}$), and applying it to the composite of

arrows f: A $\rightarrow$ A′ and g: A′ $\rightarrow$ A″ in **A** gives the composite of Ff and Fg in **B;** represented F(gf) = FgFf or by the diagram below:

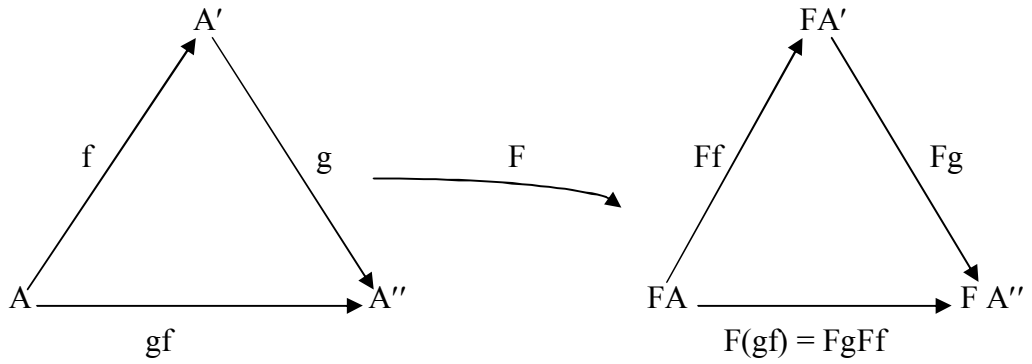


FIG 10 *A functor F applied to composite of arrows*

It may be observed that functors preserve identity morphisms and composition of morphisms.

Two important consequences of the functor axioms are as follows:

(i) F transforms each commutative diagram in **A** into a commutative diagram in **B**.

(ii) If f is an isomorphism in **A**, then F(f) is an isomorphism in **B**.

Thus, functors preserve all commuting diagrams and isomprphisms.

Functors between sets are functions, between preorders are monotone functions, between monoids are monoid homomorphisms, between groups are isomorphisms.

Examples of functors

(i) For every category **C**, there is an *identity functor* id_C: **C** $\rightarrow$ **C.**

(ii) A monoid homomorphism f: M $\rightarrow$ N determines a functor from C(M) to C(N). On objects, a homomorphism f must take the single element of C(M) to the single element of C(N). Since all arrows in both C(M) and C(N) have the same domain and the same codomain, F-1 is trivially verified as it is saying if there is an arrow between the

object M, then there is an arrow between f(M). F-2 and F-3 indicate that f is a monoid homomorphism. Conversely, every functor is determined in this way by monoid homomorphism (See [Barr & Wells (1999)], for details).

(iii) There is a functor between two posets. That is, f: S $\rightarrow$ T is a functor where S and T are posets. F-1 says if there is an arrow from x to y, then there is an arrow from f(x) to f(y) i.e., if $x \leq y$ then $f(x) \leq f(y)$. Thus, f is a monotone map. F-2 and F-3 impose no additional conditions on f since each of them asserts the equality of two specified arrows between two specified objects and in poset as category, all arrows between two objects are equal.

(iv) Let **C** be a category such that for every pair (X, Y) of objects the class **C**(X, Y) of arrows from x to y is a set. It may be recalled that such a **C** is *locally small*. For any object C of **C,** there is a functor h_C: **C** $\rightarrow$ **Set** which assigns to any object C′, the set **C**(C, C′) and any arrow f: C′ $\rightarrow$ C″ gives, by composition, a functor **C**(C, C′) $\rightarrow$ **C**(C, C″). A functor of this form is called a *representable functor*.

Types of functors

(i) **Contravariant functors**

A *contravariant functor* F from a category **C** to a category **D** is a mapping that

(a) associates to each object A$\in$**C,** an object F(A)$\in$**D**; and

(b) associates to each morphism f: A $\rightarrow$ B$\in$**C,** a morphism F(f): F(B) $\rightarrow$ F(A)$\in$**D** such that (1) $F(id_A) = id_{F(A)}$, for every object A$\in$**C**; and

(2) F(g o f) = F(g) o F(f), for all morphisms f:A$\rightarrow$B and g: B$\rightarrow$C.

It may be observed that contravariant functors or cofunctors reverse the direction of compositions.

(ii) **Constant functor**

A functor F: **C**→**D,** which maps every object in **C** to a fixed object A in **D** and, every morphism in **C** to the identity morphism on A, is called a *constant* or *selection* functor.

(iii) **Endo-functor** is a functor that maps a category to itself. For example, as an identity functor in a category **C,** written $1_{\mathbf{C}}$ or $id_{\mathbf{C}}$**,** maps an object to itself and a morphism to itself, it is an endo-functor.

(iv) **Forgetful functor** is a functor which forgets some structure. For example, the functor U: **Grp** → **Set,** which maps a group (G) to its underlying set of elements (U(G)) and a group homomorphism (f:G → H) to its underlying function of sets (U(f):U(G)→U(H)), is a forgetful functor. It can be observed that U forgets about the group structure and forgets the fact that the morphisms are group homomorphism. Forgetful functors are sometimes called *underlying functors*.

(v) **Faithful functors**: A functor F: **C** → **D** is called *faithful* if the induced mapping is injective on every hom set. Thus, if f: A → B and g: A → B are different arrows, then F(f) ≠ F(g). However, it is allowed that f: A → B and g: C → D may be different arrows, with F(A) = F(C), F(B) = F(D) and F(f) = F(g), provided that either A ≠ C or B ≠ D. For example, the underlying functors are typically faithful.

Moreover, one can compose functors i.e., if F is a functor from **A** to **B** and G is a functor from **B** to **C**, then one can form the composite functor G o F from **A** to **C** namely (G o F)(A) = G(F(A)) and (G o F)(f) = G(F(f)). Whenever, the composition of functors is defined, it is associative. Identity of composition of functors is identity functor. This shows that functors can be considered as morphisms in category of categories, for example in the category of small categories (denoted **Cat**).

Note that a small category with a single object is the same thing as monoid; the morphism of a one-object category can be thought of as elements of the monoid and composition in the category is thought of as the monoid operation. Functors between one-object Pcategories correspond to monoid homomorphisms. Thus, in a sense, functors between arbitrary categories are a kind of generalization of monoid homomorphisms to categories with more than one object. It is easy to see that **Cat** is the category whose objects are all small categories and whose morphisms are functors between categories. However, the category **Cat** itself is a large category and, therefore, not an object of itself. In order to avoid problems similar to Russell's paradox, (see Drake and Singh (1996)), it is emphasized that the category of all categories does not exist. Nevertheless, it is possible to form a *quasicategory* of all categories.

Proposition 21: Every functor preserves isomorphisms.

Proof: This is because the concept of isomorphism is defined in terms of equations involving composition and identity. If f: A $\rightarrow$ B is an isomorphism with inverse g, then F(g) is the inverse of F(f).

The proof follows as below.

$F(g) \circ F(f) = F(g \circ f) = F(id_A) = id_{F(A)}$.

Similarly, the other part follows.

$F(f) \circ F(g) = F(f \circ g) = F(id_B) = id_{F(B)}$.

4.9 Natural Transformation

Similar to the study of categories and functors, the mappings in categories can be applied to functors to form mappings between the functors, called *Natural transformations.* Besides categories and functors, natural transformation is one of the basic notions of category theory and consequently appears in the majority of its applications (see Oosten (2002) and Streicher (2004), for details). Natural transformation provides a way of transforming one functor into another while respecting the internal structures (i.e., composition of morphisms) of the categories involved. Hence, a natural transformation can be said to be a morphism of functors and this can indeed be formalized to define the so-called *functor categories*. The functor category is the category where the collections of all functors $\mathbf{A} \rightarrow \mathbf{B}$ form the objects of the category while morphisms in this category are natural transformations between functors.

Formally, if F and G are functors between categories **A** and **B**, then a natural transformation τ from F to G associates to every object A in **A** a morphism τ_A: $F(A) \rightarrow G(A)$ in **B,** called the *component of* τ *at A,* such that for every morphism f: $A \rightarrow A'$ in **A,** we have $\tau A'$ o $F(f) = G(f)$ o τA. This equation can conveniently be expressed by the commutative diagram below:

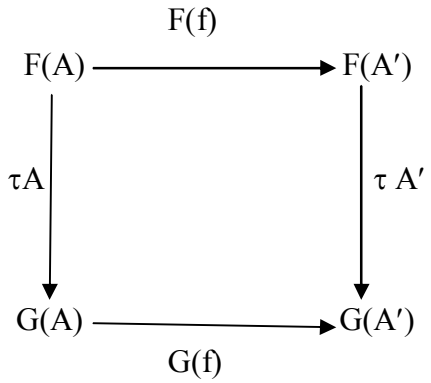


FIG 11 Natural transformation

If both F and G are contravariant, the horizontal arrows in this diagram are reversed. If τ is a natural transformation from F to G, we write τ: F $\rightarrow$ G or τ: F $\Rightarrow$ G which can also be expressed by saying that the family of morphisms τ_A: F(A) $\rightarrow$ G(A) is natural in A.

In other words, given functors F,G: **A $\rightarrow$ B,** both from a category **A** to a category **B**, a natural transformation from F to G, often written τ: F $\rightarrow$ G, is a family of arrows in **B** which carry values of F over to values of G i.e., it assigns to each object A of **A** an arrow τ_A: FA $\rightarrow$ GA between the values satisfying the *naturality* condition: for every arrow f: A $\rightarrow$ A′ in **A,** the square on the right commutes in **B**.

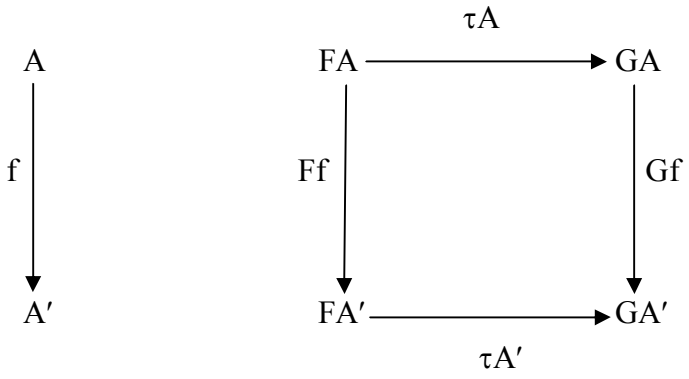


FIG 12 Natural transformation

Suppose F,G**: A $\rightarrow$ B** are functors, each carrying the *network* of **A** over to the *network* of **B**, then a natural transformation τ: F $\rightarrow$ G links each vertex in the F-image of that network to the corresponding vertex in the G-image so that the F-image *flows over* onto the G-image. The diagram below shows some arrows among the network of category **A,** while those on the right are their F-images and G-images and the components of τ among the network of arrows of category **B**.

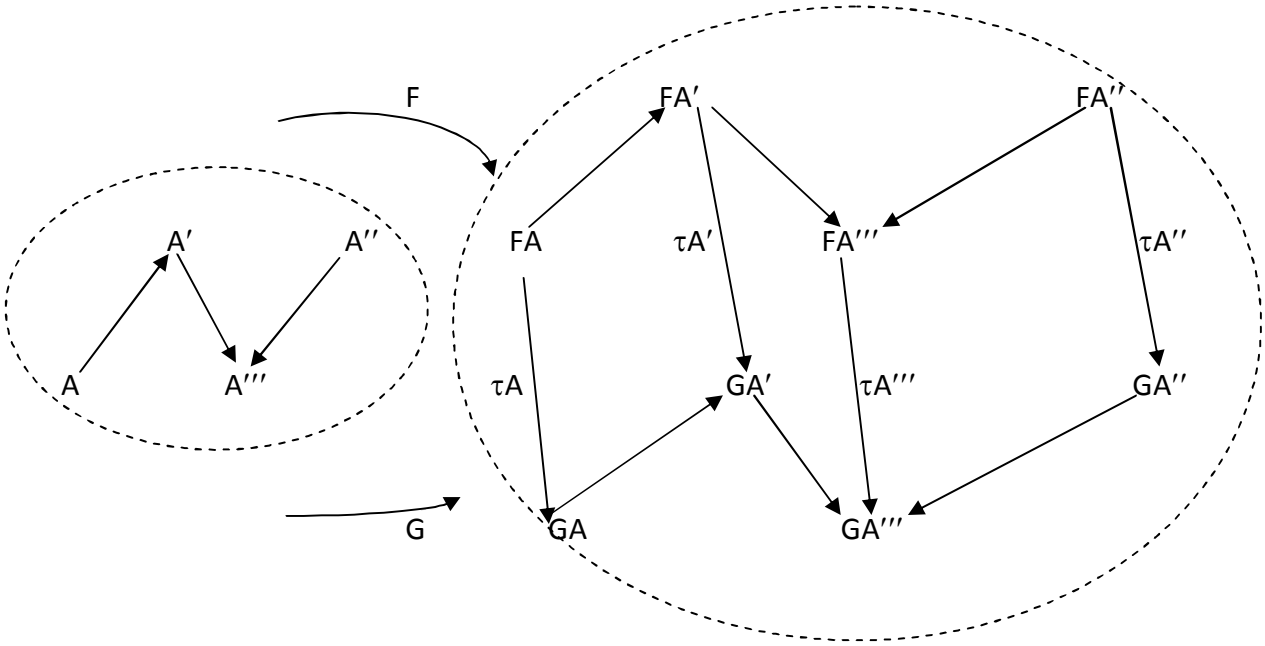


FIG 13 Network of categories in natural transformation

If for every object A in **A,** the morphism τ_A is an isomorphism in **B**, then τ is said to be a *natural isomorphism* (sometimes called *natural equivalence* or *isomorphism of functors*). Two functors F and G are called *naturally isomorphic* or simply *isomorphic* if there exist natural isomorphism from F to G.

For a given pair of categories **A, B;** each functor F: $\mathbf{A} \rightarrow \mathbf{B}$ has an identity transformation 1_F: F $\rightarrow$ F where the component arrow for each object A of **A** is the identity arrow for FA. These identity arrows clearly meet the naturality condition reflected in the diagram below:

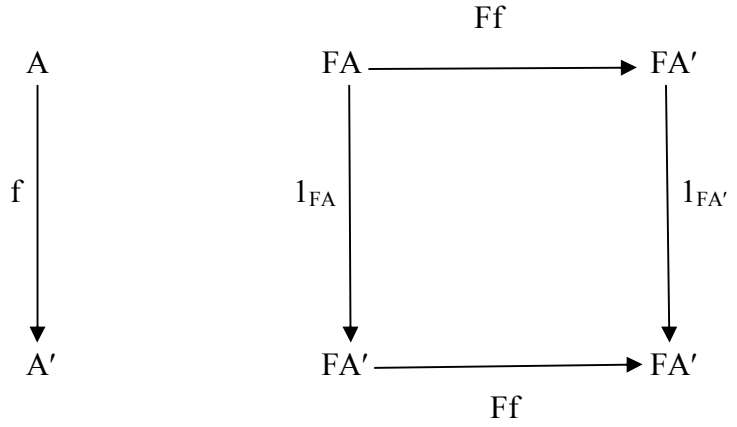


FIG 14 Identity transformations

Also, given any functors F,G,H: **A** → **B** (all from category **A** to **B**) and natural transformations τ: F → G and σ: G → H, there is a natural composite transformation στ: F → H with components σAτA. A diagrammatic representation is as follows:

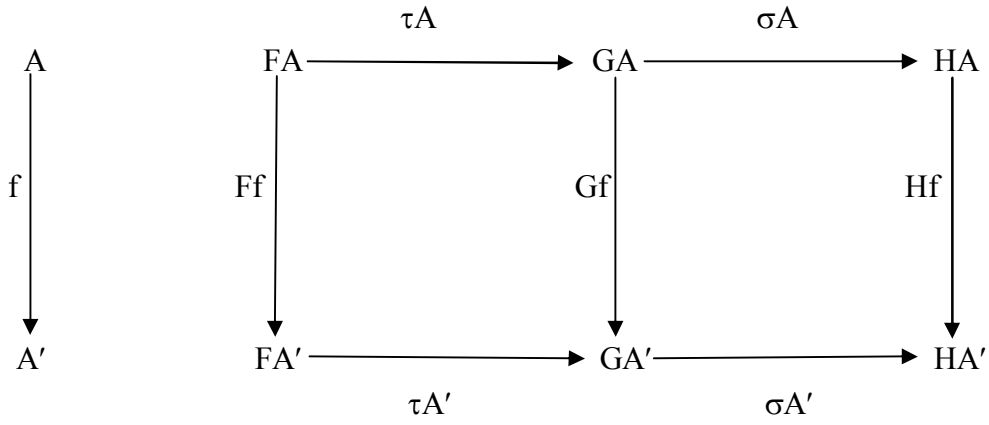


FIG 15 Composite transformations

Thus, the functors from **A** to **B** and the natural transformations between them also satisfy the Eleinberg- Maclane axioms (the axioms of a category) i.e, there is a functor category from **A** to **B**, with objects, functors: **A** → **B**, and natural transformations between them as morphisms.

Examples of natural transformations

(i) Let M and N be two monoids regarded as categories with one object, then a functor $F: M \to N$ is just the same as a homomorphism of monoids. Given two such functors, say $F,G: M \to N$, a natural transformation $F \Rightarrow G$ is (given by) an element n of N such that $nF(x) = G(x)n$, for all $x \in M$.

(ii) Let P and Q be two preorders regarded as categories. A functor $P \to Q$ is a monotone function and there exists a unique natural transformation $F \Rightarrow G$ exactly if $F(x) \leq G(x)$ for all $x \in P$.

CHAPTER FIVE

CATEGORY OF MULTISETS

5.1 Multisets

In this chapter, we briefly describe the mathematics of multisets and use it for developing the category of multisets.

Let us recall that a standard (or ordinary) set is a welldefined collection of disjoint elements. In course of time, specially in order that set theoretic tools could be made applicable to solve problems which would require some generalizations of too restrictive conditions intrinsic to ordinary sets, various *nonstandard set theories* have appeared. In particular, if multiple but finite occurrences of any element of a set are allowed, we get a generalization of the notion of a set which is called a *Multiset (mset, for short)*. Multisets have found many applications in mathematics and computer science (see [Singh *et al.* (2007)] and [Blizard (1991)], for details).

As suggested in (Monro [1987]), collections admitting objects with finite multiplicities could be viewed in two different ways: Sets with distinguishable repeated elements (e.g., people or vehicles sharing a common property), called *multisets;* and sets with indistinguishable repeated elements (e.g., *soup* of elementary particles), called *multinumbers*. However, in the literature of multisets, Monro's terminology is usually reversed. Syropoulos (2001), in particular, calls the former *Real multiset* and the latter *multiset*.

It is important to emphasize that both the concepts viz. real multisets and multisets are associated with a set equipped with an *equivalence relation* or a *function*, respectively. However, taking developments pertaining to admitting multiple occurrences of an object

in a system, it is the notion of multiset which has mostly been exploited. We, in this work, particularly to develop category of multisets, put emphasis on the notion of multisets associated with a set and an equivalence relation.

Definition and Representation of Multisets

A Multiset is an unordered collection of objects in which, unlike a standard (Cantorian) set, duplicates or multiples of objects are admitted. In other words, an mset is a collection in which objects may appear more than once and each individual occurrence of an object is called its *element.* All duplicates of an object in an mset are indistinguishable. The objects of an mset are the distinguishable or distinct elements of the mset. The distinction made between the terms *object* and *element* does enrich the multiset language. However, use of the term *element* alone may suffice if there does not arise any confusion (see[Wildberger (2003)], for details).

The use of square brackets to represent an mset is quasi-general. Thus, an mset containing one occurrence of *a,* two occurrences of *b*, and three occurrences of *c* is notationally written as [[*a, b, b, c, c, c*]] or [*a, b, b, c, c, c*] or $[a, b, c]_{1,2,3}$ or [*a,* 2*b,* 3*c*] or [*a.*1, *b.*2, *c.*3] or [1/*a,* 2/*b,* 3/*c*] or [a^1, b^2, c^3] or [a^1 b^2 c^3]. For convenience, the curly brackets are also used in place of the square brackets.

For various application purposes, we may regard a multiset [*a, a, b*] as being really of the form [*a, a′, b*] where *a* and *a′* are different objects of the same sort and *b* is of different sort from that of *a* and *a′*. In this regard, when elements of multisets are considered, elements of distinct sorts will generally be denoted by distinct letters and elements of the same sort will be denoted by the same letter with dashes distinguishing different elements of that sort (see [Monro (1987)], for details).

Formally, a multiset A is a pair (A_0, ρ), where A_0 is a set and ρ an equivalence relation on A_0. The set A_0 is called *the field of the multiset*. Elements of A_0 in the same equivalence class will be said to be of the same sort, elements in different equivalence classes will be said to be of different sorts. For example, an mset [*a, a, b, c, c, c, d*] is the multiset [*a, a′, b, c, c′, c″, d*], where all are seen to behave as different objects; but *a, a′* are of the same sort and *c, c′, c″* are also of the same sort but different from that of *a, a′*, while *b, d* are each of different sorts from the others. In other words, various equivalence classes determine the sorts. Whenever a multiset A is mentioned, its field will usually be denoted A_0.

The pair (A_0, ϕ), where A_0 is a set and ϕ denote the empty relation on A_0, is actually an ordinary set.

Multiset Functions

Let $A = (A_0, \rho)$ and $B = (B_0, \sigma)$ be multisets. A multiset function (or morphism) from A to B, written as $f: A \rightarrow B$, is a function $f: A_0 \rightarrow B_0$ which respects sorts i.e., if $a, a' \in A_0$ and $a\rho a'$, then $f(a)\sigma f(a')$. For example, f: [a, b, c] → [d, d', e] defined by f(a) = d', f(b) = e, f(c) = d; g: [a, a', b] → [c, c', d, e, f] defined by g(a) = c, g(b) = d, g(a') = c'; and g(a) = g(a') = d, g(b) = c are all multiset functions, while h: [a, a', b] → [c, c', d] defined by h(a) = c', h(a') = d and h(b) = c is not an mset function since $a, a' \in A_0$ and $a\rho a'$ but $h(a) \not\sigma h(a')$.

Let $\theta = (\phi_0, \phi)$, where ϕ_0 is an empty set and ϕ an empty relation on ϕ_0. Then, θ is said to be an *empty* multiset. That is, an empty set together with an empty equivalence relation defined on it determine an empty multiset.

Injective, surjective and bijective mset functions:

Let $A = (A_0, \rho)$, $B = (B_0, \sigma)$ be multisets. Then,

(i) f: $A \to B$ is injective iff f: $A_0 \to B_0$ is injective;

(ii) f: $A \to B$ is surjective iff f: $A_0 \to B_0$ is surjective; and

(iii) f: $A \to B$ is bijective iff f: $A_0 \to B_0$ is bijective.

That is, a multiset function is injective, surjective or bijective if the function between the fields of its domain and codomain is respectively injective, surjective or bijective. For example, f: [a, b, c] $\to$ [d, e, e′] defined by f(a) = d, f(b) = e, f(c) = e′ is an mset function that is both one-to-one and onto and therefore bijective.

Identity mset function:

Let $A = (A_0, \rho)$ be an mset, then a function f: $A \to A$ is an mset function f: $A_0 \to A_0$ called an *identity mset* function on A. In other words, a multiset function on an mset A is called an *identity function* on A, denoted 1_A and defined $1_A(a) = a$, $\forall a \in A_0$.

Composition of mset function:

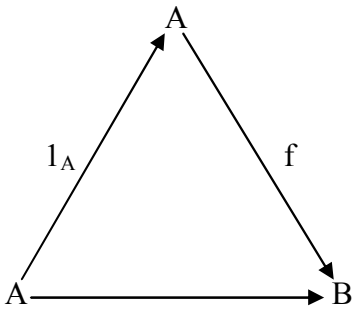
Let $A = (A_0, \rho)$, $B = (B_0, \sigma)$, and $C = (C_0, \tau)$ be multisets and f: $A \to B$, g: $B \to C$ be multiset functions, then the *composition* or *composite* or *relative product* of f and g, denoted gof (or fog, depending on the convention one follows) is the mset function gof: $A \to C$ which respects sorts (i.e., if a, $a' \in A_0$ and $a \rho a'$, then gof(a) τ gof(a'), satisfying gof(a) = g(f(a)), $\forall a \in A_0$). Sometimes, if gof is defined, it is called the *left-composition* of g with f or g is called *left-composable* with f. Moreover, on the same lines, for mset functions f: $A \to A$ and g: $A \to A$, the composite mset functions fog, gof, fof, gog, etc; can be computed. The composition of mset functions is associative.

5.2 Category of multisets

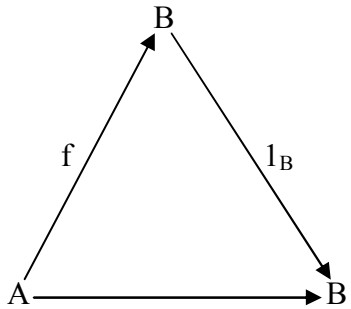
Multisets (considered as objects) and multiset functions (considered as morphisms) together determine the category of multisets, denoted **Mul.** That is, in **Mul**, the *objects* are *multisets* and the *morphisms* are *multiset functions*.

Let us consider an object $A = (A_0, \rho)$. A morphism from A to A, defined by $1_A(a) = a$ $\forall a \in A_0$, is said to be the *identity morphism* for A and for every morphism $f: A \rightarrow B$,

$f \circ 1_A = f$ and $1_B \circ f = f$, where 1_A and 1_B are the identity morphisms for A and B respectively as illustrated below:



$f \circ 1_A = f$



$1_B \circ f = f$

Typically, multisets can themselves be regarded as categories such that if A is a multiset and $a,b \in A$, there is an arrow from *a* to *b* if *a* and *b* are of the same sort, and no arrows from *a* to *b* if *a* and *b* are of different sorts.

Definitions in Mul:

Let $A = (A_0, \rho)$, $B = (B_0, \sigma)$ be objects, then a morphism $f: A \rightarrow B$ is

(i) a monomorphism iff $f: A_0 \rightarrow B_0$ is injective;

(ii) an epimorphism iff $f: A_0 \rightarrow B_0$ is surjective;

(iii) a bimorphism iff f: $A_0 \rightarrow B_0$ is bijective; and

(iv) an isomorphism iff f: $A_0 \rightarrow B_0$ is a bijection and also has the property that $a\rho a'$ iff $f(a)\ \sigma\ f(a')$.

For example, f: [a, b, c] $\rightarrow$ [d, e, e′], defined by f(a) = d, f(b) = e′, f(c) = e, is an mset morphism which is a monomorphism and an epimorphism, and therefore a bimorphism, but not an isomorphism: since f(b) and f(c) are of the same sort but b and c are of the different sorts. The multiset morphism f: [a, a′, b] $\rightarrow$ [c, c′, d], defined by f(a) = c′, f(b) = d, f(a′) = c, is an isomorphism. Moreover, the morphism g: [a, b] $\rightarrow$ [c, d], defined by g(a) = c, g(b) = d, is also an isomorphism. Thus, a bimorphism in **Mul** is not necessarily an isomorphism and consequently **Mul** is not *balanced*, whereas **Set** is balanced since every bimorphism is an isomorphism in it.

Proposition 1: In **Mul,** the monomorphisms are exactly the injective functions.

Proof:

Suppose f: A $\rightarrow$ B is injective and f o g = f o h for g,h: X $\rightarrow$ A. Then for every $x \in X$, (f o g)(x) = (f o h)(x) $\Rightarrow$ f(g(x)) = f(h(x) $\Rightarrow$ g(x) = h(x) since f is injective. Thus g = h i.e., f is a monomorphism.

Proposition 2: If f: A $\rightarrow$ B and k: B $\rightarrow$ D are injective, then k o f is also injective.

That is, the composite of two injective arrows is also injective.

Proof: let $a_1, a_2 \in A$ such that $(k \circ f)(a_1) = (k \circ f)(a_2)$ and hence $k(f(a_1) = k(f(a_2)$. Now, since k is injective, $f(a_1) = f(a_2)$ and since f is injective, $a_1 = a_2$. $\Rightarrow$ k o f is injective.

Proposition 3: In **Mul,** a split monomorphism is a monomorphism.

Proof: Let f: A $\rightarrow$ B be a split monomorphism (i.e., $\exists$ k: B $\rightarrow$ A such that k o f = 1_A) and suppose f o g = f o h for g,h: C $\rightarrow$ A. We are to show that g = h. Now for each c$\in$C, g(c) = (1_A o g)(c) = ((k o f) o g)(c) = (k o (f o g))(c) = (k o (f o h))(c) = ((k o f) o h)(c) = (1_A o h)(c) = h(c) $\Rightarrow$ g = h. Since g = h, f is a monomorphism.

Proposition 4: The composite of two monomorphisms in **Mul** is a monomorphism.

Proof: Let f: A $\rightarrow$ B and p: B $\rightarrow$ D be monomorphisms. We are to show that if (p o f) o g = (p o f) o h for g,h: C$\rightarrow$ A, then g = h i.e., p o f is a monomorphism.

Now for each c$\in$C, ((p o f) o g)(c) = ((p o f) o h)(c) $\Rightarrow$ ((p o f)g)(c) = ((p o f)h)(c) $\Rightarrow$ p(f(g(c))) = p(f(h(c))). Since p and f are monomorphisms, we have g(c) = h(c) i.e., g = h. Hence p o f is a monomorphism.

Proposition 5: In **Mul,** a surjective arrow is an epimorphism.

Proof: Let f: A $\rightarrow$ B be surjective and g o f = h o f for g,h: B $\rightarrow$ C. Now, in order to show that f is an epimorphism, we need to show that g = h. As for every a$\in$A, (g o f)(a) = (f o h)(a) i.e., g(f(a)) = h(f(a)), we have g = h.

Proposition 6: In **Mul,** a split epimorphism is an epimorphism.

Proof:

Let f: A $\rightarrow$ B be a split epimorphism i.e., $\exists$ k: B $\rightarrow$ A such that f o k = 1_B.

Let g o f = h o f for g,h: B $\rightarrow$ X. We are to show that g = h.

Now, for each b$\in$B, we have g(b) = (g o 1_B)(b) = (g o (f o k))(b) = ((g o f) o k)(b) = ((h o f) o k)(b) = (h o (f o k))(b) = (h o 1_B)(b) = h(b). Hence, g = h i.e., f is an epimorphism.

Proposition 7: In **Mul**, the composite of two epimorphisms is an epimorphism.

Proof: Let f: A $\rightarrow$ B, q: B $\rightarrow$ C be epimorphisms and g o (q o f) = h o (q o f). We need to show that g = h for every pair g,h: C $\rightarrow$ D. Now, for each a$\in$A, we have (g o (q o f))(a) =

(h o (q o f))(a) which implies g(q(f(a))) = h(q(f(a))), and since f and q are epimorphisms; g = h.

Proposition 8: In **Mul**, the composite of two bimorphisms is a bimorphism.

Proof: Let f: A $\rightarrow$ B, k: B $\rightarrow$ D be bimorphisms. We have, f o g = f o h $\Rightarrow$ g = h for g,h: C $\rightarrow$ A; g o f = h o f $\Rightarrow$ g = h for g,h: B $\rightarrow$ X; k o g = k o h $\Rightarrow$ g = h for g,h: C $\rightarrow$ B; and g o k = h o k $\Rightarrow$ g = h for g,h: D $\rightarrow$ X. We need to show that (k o f) o g = (k o f) o h $\Rightarrow$ g = h for g,h: C $\rightarrow$ A, and g o (k o f) = h o (k o f) $\Rightarrow$ g = h for g,h: D $\rightarrow$ X. Now, for each c$\in$C, ((k o f) o g)(c) = (k o (f o g))(c) = (k o (f o h))(c) i.e., k(f(g(c))) = k(f(h(c))). Since k and f are bimorphisms, we have g(c) = h(c) i.e., g = h.

Moreover, for each a$\in$A, (g o (k o f))(a) = ((g o k) o f)(a) = ((h o k) o f)(a) i.e., g(k(f(a))) = h(k(f(a))). Since f and k are bimorphisms, we have g = h. Hence, k o f is a bimorphism.

Proposition 9: In **Mul**, every isomorphism is a bimorphism.

Proof: Let f: A $\rightarrow$ B be an isomorphism i.e., $\exists$ k: B $\rightarrow$ A such that k o f = 1_A, f o k = 1_B and suppose f o g = f o h for g,h: C $\rightarrow$ A; g o f = h o f for g,h: B $\rightarrow$ C: We need to show that g = h in both the cases.

Now, for each a$\in$A, g(a) = (1_A o g)(a) = ((k o f) o g)(a) = (k o (f o g))(a) = (k o (f o h))(a) = ((k o f) o h)(a) = (1_A o h)(a) = h(a) i.e., g = h.

Also, for each b$\in$B, g(b) = (g o 1_B)(b) = (g o (f o k))(b) = ((g o f) o k)(b) = ((h o f) o k)(b) = (h o (f o k))(b) = (h o 1_B)(b) = h(b) i.e., g = h. Hence f is a bimorphism.

It may be recalled (see 5.2) that the converse need not hold in **Mul**.

Dual of a Mul

The dual of a **Mul,** denoted $\mathbf{Mul}^{op}$**,** is a category in which

(i) objects and arrows are those of **Mul**;

(ii) if f: A $\rightarrow$ B is in **Mul**, then f: B $\rightarrow$ A is in $\mathbf{Mul}^{op}$ i.e., the source and target are reversed;

(iii) if h = gof in **Mul**, then h = fog in $\mathbf{Mul}^{op}$; and

(iv) the identity arrows in **Mul** and $\mathbf{Mul}^{op}$ are the same.

Objects in Mul

An object I in a category **C** is said to be an *initial* object of **C** if for every object X of **C** there is a unique morphism I $\rightarrow$ X. In this definition, the nature of a particular object I is described in terms of its relation to all other objects X in the category. Since every object X in **Mul** there is a unique morphism $\theta \rightarrow$ X, the empty multiset θ can be regarded as an initial object in **Mul.**

Also, an object T in a category **C** is said to be a *terminal* object of **C** if for each object X of **C** there is exactly one morphism X $\rightarrow$ T. In **Mul,** an object satisfying this universal mapping property is an one-element multiset, denoted **1**. Indeed, for each object X of **Mul**, there is a unique morphism X $\rightarrow$ **1**. Hence, any singleton mset is a terminal object in **Mul.** As we may have more than one singleton, they must be isomorphic.

Since a zero object is an object that is both initial and terminal, there is no zero object in **Mul**.

Thus the category of multisets has a unique initial object, but many terminal objects which are isomorphic to each other. The proof follows:

Proposition 10: All singletons in **Mul** are isomorphic.

Proof: Suppose M and M′ are singletons. Thus, there are morphisms g: M → M′ and h: M′ → M. The composite g o h is an arrow M′ → M′ which is the identity $1_{M'}$. Since M′ is a singleton, there is only one arrow M′ → M′, and hence, g o h = $1_{M'}$.

Similarly, h o g = 1_M.

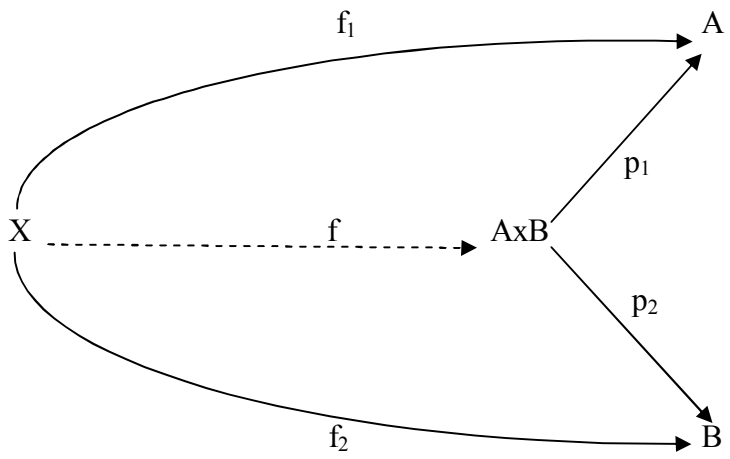
Points of an object in Mul

A point of an object X in a category **C** is an arrow **1** → X, where **1** is terminal. Hence, in the category **Mul**, the points of an object A are precisely the elements of A. Suppose A = [a, a′, b] is an object in the category **Mul**, then A has three points and, each point of A can be seen to *point to* exactly one element of A and every element of A is the value of exactly one point of A. Moreover, if f: A → B and *a* is a point of A, then f*a* is a point of B. The terminal object in **Mul** has one point. The terminal object **1** in **Mul** helps in separating arbitrary arrows. For the parallel arrows f,g: A → B, if for each point *a* of A we have f*a* = g*a*, then f = g. Thus, if two arrows *agree* on points, they are the same arrows. For example, given f: A → B and g: A → B, if fa = ga for every point a: **1** → A, then we can conclude that f = g. In other words, if f ≠ g, then there exists at least one point a: **1** → A for which fa ≠ ga. Similar to **Set**, two arrows in **Mul** are equal if they have the same domain and codomain and if they have the same value at every element of their domain.

Product of Objects in Mul

An object AxB, together with a pair of maps AxB $\xrightarrow{p1}$ A, AxB $\xrightarrow{p2}$ B, is called the *product* of the objects A and B if for every object X and each pair of maps X $\xrightarrow{f1}$ A, X $\xrightarrow{f2}$ B, there

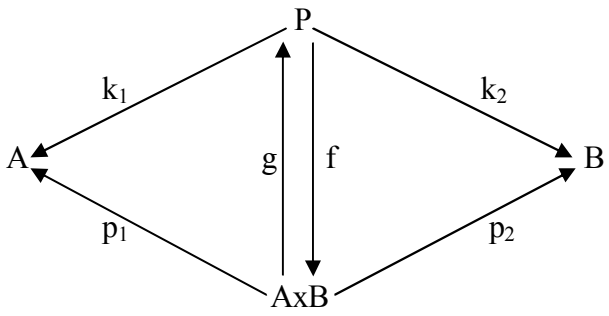
is a unique map $X \xrightarrow{f} AxB$ such that $f_1 = p_1 f$, $f_2 = p_2 f$. Diagramatically, it can be put as follows:



Since f is uniquely determined by f_1 and f_2, we can denote it by $\langle f_1, f_2 \rangle$. The maps p_1 and p_2 are called *projection maps* for the product or simply *projections*, (i.e., projections of the product to its factors).

In **Mul**, each point of the product of the objects A, B can be uniquely represented in the form $\langle a, b \rangle$, where *a* is a point of A and *b* is a point of B. That is, the points of the product of two objects are the pairs of points, one from each factor. We recall that if two arrows agree on points, they are the same i.e., an arrow is completely determined by the values of its points. Therefore, as soon as we know the points of the objects, we can determine their product.

Proposition 11: In **Mul,** if an object P and the functions k_1: P $\rightarrow$ A and k_2: P $\rightarrow$ B have the product property, then P is isomorphic to AxB and, there is exactly one isomorphism f: P $\rightarrow$ AxB with $k_1 = p_1$ o f and $k_2 = p_2$ o f. Diagramatically, it can be put as follows:



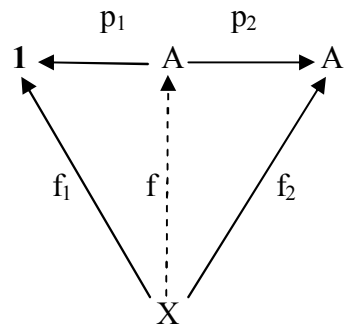
Proof: Using the product property of P, k_1 and k_2, we define g: AxB $\rightarrow$ P. Let f: P $\rightarrow$ AxB with $k_1 = p_1$ o f and $k_2 = p_2$ o f, and g: AxB $\rightarrow$ P with $p_1 = k_1$ o g and $p_2 = k_2$ o g. For a unique isomorphism between them, we need to show that g o f = 1_P and f o g = 1_{AxB} hold. We have, k_1 o (g o f) = (k_1 o g) o f = p_1 o f = k_1 and, k_2 o (g o f) = (k_2 o g) o f = p_2 o f = k_2. Since, 1_P: P $\rightarrow$ P is the only arrow that will compose with k_1 and k_2 to give k_1 and k_2 respectively g o f must be the identity arrow i.e., g o f = 1_P.

Similarly, since p_1 o (f o g) = (p_1 o f) o g = k_1 o g = p_1 and, p_2 o (f o g) = (p_2 o f) o g = k_2 o g = p_2; we have f o g = 1_{AxB}.

Proposition 12: In **Mul,** every object A is isomorphic to the product Ax**1**, where **1** is terminal.

Proof: An object A, with P_1: A $\rightarrow$ **1** and P_2: A $\rightarrow$ A as projections, has the product property, i.e., A with P_1: A $\rightarrow$ **1** and P_2: A $\rightarrow$ A as projections such that $f_1 = P_1$ o f, $f_2 = P_2$ o f (figure **a**).

Clearly Ax**1,** with j_1: Ax**1** $\rightarrow$ **1** and j_2: Ax**1** $\rightarrow$ A as projections, has the product property i.e., an mset Ax**1** with j_1: Ax**1** $\rightarrow$ **1** and j_2: Ax**1** $\rightarrow$ A as projections such that $q_1 = j_1$ o q, $q_2 = j_2$ o q (figure **b**).



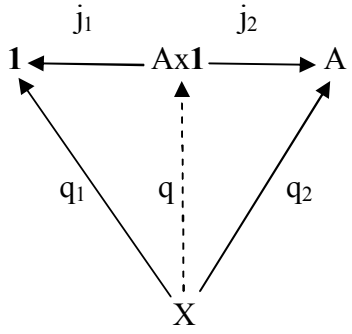
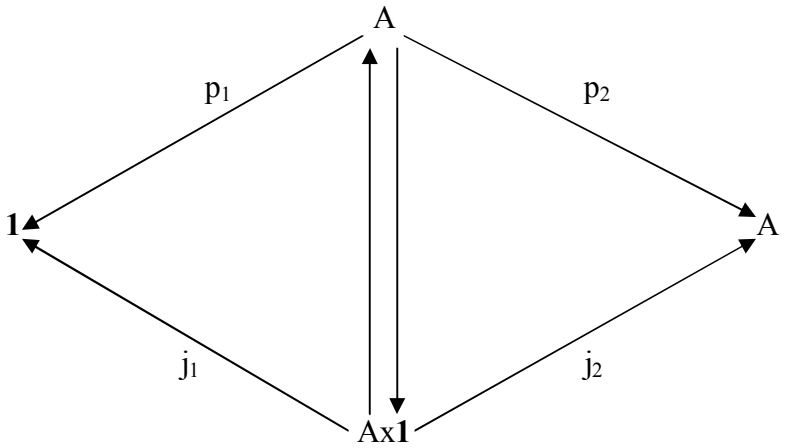


Figure **a** Figuer **b**

Taking the aforesaid facts and the diagram below into consideration, it follows from proposition 11 A is isomorphic to Ax**1,** where **1** is terminal in **Mul**.

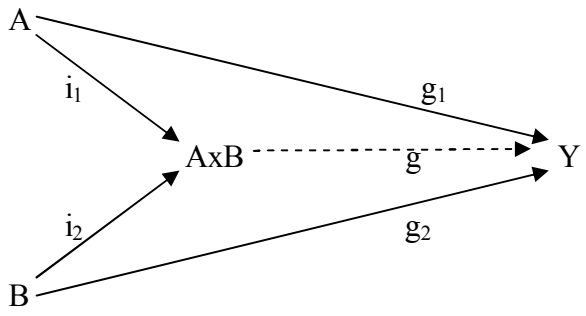


Dual of Product-Projections

Dualizing the notion of product-projections, we get:

An object A+B together with a pair of maps A $\overset{i1}{\rightarrow}$ A+B, B $\overset{i2}{\rightarrow}$ A+B in a category is said to be a coproduct of A and B if for each pair of maps A $\overset{g1}{\rightarrow}$ Y, B $\overset{g2}{\rightarrow}$ Y, there is a unique map A+B $\overset{g}{\rightarrow}$ Y such that $g_1 = g \circ i_1$ and $g_2 = g \circ i_2$.

That is, the following diagram commutes:



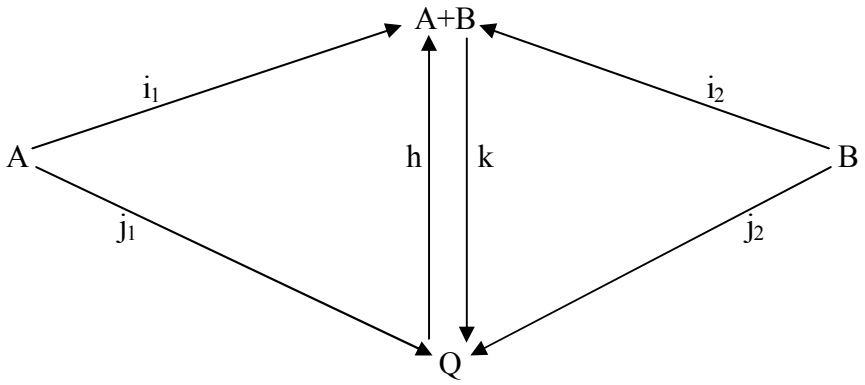
Thus, $g_1 = g \circ i_1$ and $g_2 = g \circ i_2$, where i_1 and i_2 are called *injections* for the coproduct. Recall that in **Mul**, the coproduct is called the *sum*. The functorial extension of this operation from objects to all functions follows just as in the case of the product operation. Also, *Sums* have the property that any point of A+B comes via injection from a point of exactly one of A, B.

Proposition 13: There is exactly one isomorphism with injections between any two *Sums* of the objects A and B in **Mul**.

Proof:

Suppose Q, with j_1: A $\rightarrow$ Q and j_2: B $\rightarrow$ Q as injections, has the sum property like that of A+B, i_1 and i_2, then there is an isomorphism k: A+B $\rightarrow$ Q such that $j_1 = k \circ i_1$ and $j_2 = k \circ$

i_2. Using the Sum property we define h: Q $\rightarrow$ A+B with i_1 = h o j_1 and i_2 = h o j_2. Diagramatically, it can be put as follows:



We need to show that k o h = 1_Q and, h o k = 1_{A+B}.

Since, (k o h) o j_1 = k o (h o j_1) = k o i_1 = j_1 and (k o h) o j_2 = k o (h o j_2) = k o i_2 = j_2; and also 1_Q o j_1 = j_1 and 1_Q o j_2 = j_2; by the uniqueness property, we have k o h = 1_Q. Similarly, h o k = 1_{A+B}.

CHAPTER SIX

SUMMARY, CONCLUSION AND FUTURE DIRECTIONS

6.1 Summary of our major findings

In chapter 2 through chapter 4, a critical survey of the literature describing the theory of categories, fundamental notions related to categories and various examples of categories are provided. This part of our study attempts to illustrate how category theory treats the notion of structure in a uniform manner and thereby organizes and unifies much of mathematics.

Our major research contribution in this thesis lie in developing theory of category of multisets as described in chapter 5. We regard a multiset [*a, a, b*] as being really of the form [*a, a′, b*] where *a* and *a′* are different objects of the same sort and *b* is of different sort from *a* and *a′*. From this idea we consider a multiset A as a pair (A_0, ρ), where A_0 is a set and ρ an equivalence relation on A_0. The set A_0 is called the *field* of the multiset. Elements of A_0 in the same equivalence class will be said to be of the same sort, elements in different equivalence classes will be said to be of different sorts.

Multisets as objects and multiset functions as morphisms determine a category denoted **Mul**. The notion of injection, surjection and bijection are introduced in **Mul** on a par with their introduction in set categories. Further, following the definition of multiset functions in Monro (1987), extension is made for the composition of multiset functions which is proved to be associative. It is observed that contrary to **Set** where every bimorphism is an isomorphism, in **Mul** a bimorphism is not necessarily an isomorphism. Therefore **Mul** is not *balanced*, while **Set** is balanced (since every bimorphism is an isomorphism).

It is further observed that in **Mul,** the monomorphisms are exactly the injective functions while the epimorphisms are the surjective functions and every split monomorphism is a monomorphism and every split epimorphism is an epimorphism. Also, it is proved that in **Mul,** the composite of two monomorphisms, epimorphisms and biomorphisms are respectively monomorphisms, epimorphisms and biomorphisms. It is to be recalled from chapter 5 that in **Mul,** a morphism f is a biomorphism iff it is bijective and every isomorphism is a bimorphism but the converse is not necessarily true.

It is shown that a singleton mset is a terminal object and the empty mset is an initial object. Also, there is no zero object in **Mul,** since a zero object is an object that is both initial and terminal.

We prove that in **Mul** the product of two objects is unique as is the case for the sum.

That is, there is exactly one isomorphism with injections between any two *Sums* of the objects A and B in **Mul**.

An important observation in regard to a distinguishing property of **Mul** from general category is as follows:

In a category, a morphism f: A $\rightarrow$ B is called an isomorphism if it is invertible i.e., if $\exists$ a morphism g: B $\rightarrow$ A with g o f = 1_A and f o g = 1_B. In the category of multisets, suppose A = (A_0, ρ) and B = (B_0, σ) are multisets, then the isomorphisms are the bijective mappings that has the property aρa′ iff f(a) σ f(a′). That is, any arrow, which is both a monomorphism and an epimorphism and has the property that aρa′ iff f(a) σ f(a′), is an isomorphism.

It is observed (see 5.2) that, the terminal object **1** in **Mul** helps in separating arbitrary arrows. For the parallel arrows f,g: A $\rightarrow$ B, if for each point *a* of A we have f*a* = g*a*, then

f = g. Thus, if two arrows *agree* on points, they are the same arrows. For example, given f: A $\rightarrow$ B and g: A $\rightarrow$ B, if fa = ga for every point a: **1** $\rightarrow$ A, then we can conclude that f = g. In other words, if f $\neq$ g, then there exists at least one point a: **1** $\rightarrow$ A for which fa $\neq$ ga. Similar to **Set**, two arrows in **Mul** are equal if they have the same domain and codomain and if they have the same value at every element of their domain.

6.2 Conclusion and some future directions

In course of developing this thesis, it has been sufficiently illustrated that the structure of a certain class of mathematical objects can be best studied by identifying the properties of the functions that transform a member of the class into another constitutes the basic idea of the theory of categories. The theory is seen to provide a unification within set theoretic environment, organizing and unifying thereby much of mathematics. This thesis emphasizes that the theory of categories, by now, has emerged as a powerful language or a conceptual framework providing tools to characterize the universal components of a family of structures of a given kind and their relationships.

As Chu categories are large categories, it may provide a uniform framework for different objects of mathematics and hence, the study of Chu spaces are currently being studied.

Also, considering the numerous applications of multisets and category theory, and as very few contributions on category of multisets (**Mul**) are available, it needs to be studied for future research.

REFERENCES

1. Barr, M. and Wells, C. (1999). Category Theory. *Lecture Notes for ESSLLI (European Summer School of Logic, Language and Information)*, pp. 1-128.

2. Blass, A. (1984). The interaction between category theory and set theory, Contemporary Mathematics. *American Mathematical Society*, vol.30, PP. 5 – 29.

3. Blizard, W. D. (1989). Multiset Theory. *Notre Dame Journal of formal logic*, Vol. 30, PP. 36 – 66.

4. Blizard, W. D. (1991). The development of Multiset theory, *Modern Logic*, vol.1, PP. 319 – 352.

5. Burgin, M. (2000). Unified Foundations for Mathematics. *Department of mathematics, University of California Los Angeles, U. S. A.*, pp. 1-28.

6. Cockett, J.R.B. (2010). Categories and Computability, *University of Calgary, Canada*, pp. 1-93.

7. Drake, F. R. and Singh, D. (1996). *Intermediate set Theory*, John Wiley, pp. 1-35.

8. Fokkinga, M. M. (1994). A Gentle Introduction to Category Theory: the calculational approach, *university of Twente, the Netherlands*, pp. 1-58.

9. Ivancevic, V. G. and Ivancevic, T. T. (2009). Applied categories and functors for undergraduates, lecture notes, pp. 1-28.

10. Kuku, A. O. (1992). *Abstract Algebra.* Ibadan University Press, chapter 8, pp. 353-387.

11. Laan, V. (2003). Introduction to category theory, Lecture Notes, pp. 1 -52.

12. Landry, E. (1999). Category theory: The language of mathematics, Proceedings of the 1998 Biennial Meetings of the philosophy of science Association, *University of Chicago Press*, vol.66, pp. S14-S27.

13. Lawvere, F. W. (1964). An elementary theory of the category of sets: Proceedings of the National Academy of sciences, U. S. A., vol. 52, PP. 1506- 1511.

14. Lawvere, F. W. (2005). An elementary theory of the Category of Sets (Long version) with commentary, Reprints in theory and applications of Categories, No. 11, PP. 1 – 35.

15. Lawvere, F. W. and Schanuel, S. H. (2009). *Conceptual Mathematics: a first introduction to categories*, (second edition), Cambridge University Press, pp. 1-205.

16. Marquis, J.P. (2010). Category Theory, Metaphysics Research Lab, CSLI (Centre for the study of Language and Information), *Stanford University*, pp. 1-22.

17. Mc Larty, C. (2008). Introduction to Categorical Foundations for Mathematics, Rough notes after discussion at Roskilde version of August 14, 2008, pp. 1-79.

18. Monro, G. P. (1987). The concept of multiset. *Zeitschrift für mathematische Logik und Grundlagen der mathematic [mathematical logic quarterly]*, vol. 33, issue 2, pp. 171-178.

19. Murfet, D. (2006). Foundations for category theory. Lecture Notes, PP. 1 – 9.

20. Oosten, J. V. (2002). Basic Category Theory. Lecture Notes, *Utrecht university, Netherlands*, Revised edition, pp. 1- 86.

21. Rydeheard, D. E. and Burstall, R. M. (2001). Computational Category Theory, *University of Edinburg Publication*, pp. 1-263.

22. Singh, D., Ibrahim, A. M., Tella, Y. and Singh, J. N. (2007). An Overview of the Applications of Multisets. *NOVI Sad J. Math*, vol. 37, No. 2, PP. 73 – 92.

23. Singh, D., Isah, A. I., and Ibrahim, A. M. (2012). A Survey of the Development of the Theory of Categories. *International Journal of Innovation Management and Technology*, vol.3, No. 2, pp. 156-159.

24. Singh, D. and Peter, C. M. (2011). Multiset-based Tree Model for Membrane Computing. *Computer Science Journal of Moldova,*vol.19, no. 1, pp. 55.

25. Streicher, T. (2004). Introduction to category theory and categorical logic, *Polimetrica*, PP. 1 – 116.

26. Syropoulos, A. (2001). Mathematics of multisets, C.S.Calude *et al*. (Eds.): Multiset processing, LNCS 2235, pp. 347-358.

27. Syropoulos, A. (2003). Categorical models of multisets. *Romanian Journal of Information Science and Technology*, vol.6, numbers 3-4, pp. 393-400.

28. Turi, D. (2001). Category theory. Lecture Notes, Laboratory for foundation of computer science, *University of Edinburgh*, pp. 1-58.

29. Wildberger, N. J. (2003). A new look at multisets, School of Mathematics, UNSW, *Sydney* 2052, Australia, PP. 1- 21.