A STUDY OF COMPATIBILITY RELATION AND ITS APPLICATION TO SWITCHED NETWORKS

BY

OPUBO Tamunokuro William-West

M.Sc/SCIE/09142/2010-2011

B.Sc (ABU, 2009)

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

DEPARTMENT OF MATHEMATICS

FACULTY OF SCIENCES

AHMADU BELLO UNIVERSITY, ZARIA

NIGERIA

MAY, 2013

DECLARATION

I declare that this research work entitled *A Study of Compatibility Relation and its Application to Switched Networks* has been performed by me under the supervision of Prof. D. Singh and Dr A.M. Ibrahim, Department of Mathematics. This work has not been presented for the award of any degree in any higher institution of learning. All sources of information used in this work are duly acknowledged by means of references.

Opubo Tamunokuro William-west

Sign ……

Date ……...

CERTIFICATION

This research work entitled *A Study of Compatibility Relation and its Application to Switched Networks* by Opubo Tamunokuro William-west meets the requirements governing the award of the degree of Master of Science in Mathematics of Ahmadu Bello University, Zaria and is approved for its contribution to knowledge and literary presentation.

Prof. D. Singh	Date
Chairman, Supervisory Committee	
Dr. A.M Ibrahim	Date
Member, Supervisory Committee	
Dr. B. Sani	Date
Head of Mathematics Department	
Dean, Post Graduate School	Date
External Examiner	Date

DEDICATION

This research work is dedicated with love to God, whose love, infinite mercy and grace saw me through; my parents Mr and Mrs Opubo W. West who assisted me to the best of their ability through this programme, and my brother, Michael O. West, as well as my sisters; Pamela West and Boma West, whose love encouraged me to go on.

ACKNOWLEDGEMENT

With the deepest gratitude and appreciation, I convey my sincere appreciation to those who contributed towards the success of this research work.

I express my sincere gratitude to my master and supervisor, Prof. D. Singh, Department of Mathematics, Ahmadu Bello University, Zaria for his fortitude in giving valuable advice and constant encouragement throughout the period of this research work. Indeed, I am indebted to him for all the correction made and for bringing out the better part of me in the course of my work with him. I am equally indebted to my minor supervisor Dr. A.M. Ibrahim.

I am grateful to Prof. B.K. Jha, Department of Mathematics, Ahmadu Bello University, Zaria who has greatly inspired me throughout my stay in the course of this programme.

I am also grateful to my respected Head of Department of Mathematics, Dr B. Sani, Prof J. Singh, Dr A.O. Ajibade and all the members of staff of the Department for their encouragement. In fact, the course work lectures which I received have been of great help to me in conducting this research.

I also acknowledge numerous assistance rendered to me by my colleagues, Shagaiya Daniel Yahaya, Paul Austine Ejegua, Aliyu Zakari, John Yaknan Gambo, Odunayo Aribisala, Abdulsamad Imam, I extend a hand of appreciation to my friends outside the department namely; Princess. O Samuels, Umaru Bello, Sunday A. Kingsley, Mohammed Jiddah, Daniel Ifraimu and Johnson Mbashinya.

It would of course, be impossible for me to find words that are appropriate enough to express my sincere gratitude to my parents Mr and Mrs Opubo W. West, without whose total support and understanding, this research work, may never have been realized.

Finally, I express my gratitude to God Almighty who saw me through the academic stress I encountered in the course of my studies at Ahmadu Bello University, Zaria.

May God be gracious to those who are gracious and be Omega to those who start a good thing.

ABSTRACT

In this thesis, we have studied the concept of compatibility relation and construction of an algorithm for determining maximal compatibility classes. We have identified some new characteristic properties of compatibility classes and constructed an alternative algorithm for computing maximal compatibility classes. Finally, we have described the concept of how maximal compatibility classes help in segmentation and decentralization of switched networks.

Content

Title page...……..…….....i

Declaration……..……..ii

Certification……...iii

Dedication……...iv

Acknowledgement……...v

Abstract……vii

Content……..….viii

1 Introduction **1**

1.1 Background of the study……..……...……...1

1.2 Statement of the problem……..……...……..3

1.3 Motivation for the Research…….……..……3

1.4 Objectives of the Research……..……..…….....3

1.5 Methodology…….…….....3

1.6 Organization of the thesis……..…..……..….3

2 Literature Review **5**

2.1 Literature Review……..……..5

2.2 Binary Relations and Their Properties….…….……9

2.3 Definition…….....9

2.4 Types of Relation…….……10

2.5 Power of Relation……..…….....……12

2.6 Representation of Relation …….....13

2.7 Matrix Representation of Relation…….....13

2.8 Graphical Representation of Relation……..15

2.9 Representing Relation using Sagittal diagram…..……...16

3 Compatibility Relation and Some related Concepts 19

3.1 Compatibility Relation and Some related Concepts……19

3.2 Compatibility Relation……19

3.3 Characteristic Properties of Compatibility Classes…… 23

3.4 Concept of Compatibles in Sequential Machines……....26

3.5 Maximal Compatible Chart…….28

3.6 Relation matrix of a Compatibility Relation and Construction of an algorithm to compute MCC ……..32

3.7 Algorithm to compute MCCs from CMCCs……...33

4 Application of Compatibility Relation to Switched Networks 44

4.1 Introduction……..... 44

4.2 Some Definitions..……..……..…….. 44

4.3 Throughputs……..……...46

4.3.1 Maximum theoretical throughput……....47

4.3.2 Peak measured throughput……...47

4.3.3 Maximum sustained throughput …….. 48

4.4 Concept of Switching……...49

4.4.1 Switches in Network…..50

4.4.2 Network Congestion……..51

4.4.3 Network Segmentation……51

4.4.4 Limitation of Network Segmentation Method (NSM)…….....52

4.4.5 Application of Compatibility Relation to Switched Network……...52

4.4.6 Compatibility Relation based segmentation procedure……....53

4.5 Network Decentralization……59

4.5.1 Some Basic Definitions……59

4.5.2 The Structure of a Decentralized Network……...60

4.5.3 Limitations of a Decentralized Network……...61

4.5.4 Application of the notion of maximal compatibles in network decentralization……..…62

4.5.5 Some Characteristics of Compatibility based Decentralized Network…….63

5 Conclusion and Future direction **65**

5.1 Conclusion…….65

5.2 Recommendations……65

References……..67

List of Figures and Tables

1. Figure 2.1: graphical representation of relation.
2. Figure 2.2: representation of reflexive and symmetric relation.
3. Figure 2.3: graphical representation of reflexive, symmetric and transitive relation.
4. Figure 2.4: sagittal diagram.
5. Figure 3.1: compatibility graph.
6. Figure 3.2: simplified graph of a compatibility relation.
7. Table IA: flow table.
8. Table IB: compatibility chart.
9. Table II: maximal compatibility chart.
10. Table III: maximal compatibility chart.
11. Table IVA: compatibility chart.
12. Table IVB: maximal compatibility chart.
13. Table A: compatibility table.
14. Table B: compatibility table representing CMCCs and MCCs.
15. Table C: compatibility table.
16. Table 1: compatibility table.
17. Table 2: compatibility table.
18. Table 3: compatibility table.
19. Table A1: compatibility table.
20. Table B1: compatibility table.
21. Figure 4.1: OSI model.
22. Figure 4.2: OSI model implementation diagram.
23. Figure 4.3: switches in network.

24. Figure 4.4: unsegmented network.

25. Figure 4.5: segmented network.

26. Figure 4.6: compatibility based segmentation procedure.

27. Figure 4.7: compatibility graph.

28. Figure 4.8: diagram representing links and queues.

29. Figure 4.9: star topology.

30. Figure 4.10: mesh topology.

31. Figure 4.11: graph of R_1.

32. Figure 4.12: graph of R_2.

33. Figure 4.13: decentralized network organized as middleware.

CHAPTER ONE

INTRODUCTION

1.1 Background of the study

It is difficult to trace since when the profound mathematics of *compatibility relation* came into existence. To our knowledge, Kurepa (1952- 1953) seems to be the earliest full- blown mathematical exposition on the study of reflexive symmetric relations and graphs. Since then, a number of works (Paull and Unger, 1959; William, 1962; McCluskey, 1962; Weissman, 1962; Grasselli (1962, 1965, 1966); Hill and Peterson, 1968; Kohavi, 1978; Pognowski, 1981; Chakraborty, 1987; Biswas, 1989; Yager, 1990; Yen, 1990; Zhao *et al*., 1997; Singh and William-west, 2012; etc.), dealing with fundamentals of compatibility relation as well as its applications has appeared.

As of now, it can be said that the actual import of compatibility relation has gone far beyond its ordinary linguistic connotation and even that of its mathematical characterization. It has found several applications in different fields of knowledge. Essentially, from application points of view, compatibility relation is useful in solving a class of minimization problems, particularly when the problems are incompletely specified:

(a) In switching theory, particularly for incompletely specified problems (McCluskey, (1962); Singh and William-west, (2012));
(b) In incompletely specified sequential networks for reduction of the number of internal states (Charkraborty, 1987);
(c) In designing of a class of digital control Units (Grasselli, 1962);
(d) In graph theory (Hills and Peterson, 1968);
(e) In solving some combinatorial problems, such as scheduling of traffic control (Grasselli, 1966);
(f) In phonology (Pognowski, 1981);

just to mention a few, (see (Grasselli, 1962) for some further details).

Further, the notion of compatibility relation has been used in augmenting the theory of evidence(Dempster- Shafer Structure, in particular) where generalized compatibility relation is combined, based on possibility theory to extend Dempster' rule (Yen, 1990).

Also, the application of compatibility relation is made in deriving inference about one variable, based upon the knowledge about another variable, in *Belief structure* (Yager, 1990).

In recent years, some application of compatibility relation can be found in defining *Tolerance multisets*in (Pognowski, 1981), *Tolerance rough sets* in (Lipovaski, 1994) and *Symmetrized Max-plus Algebra* (Singh *et al.*, 2010), etc.

In fact, compatibility relation is known to have a number of applications in Computer science. For example, in a sequential circuit, when redundant states need to be removed to minimize failures, particularlywhen the number of memory elements is directly related to the number of states, compatibility relation is used (Zhao *et al.*, 1997).

Usually, in applications of compatibility relation, such as *state minimization of incompletely specified sequential machines (ISSM)*, it is required to derive the set of all maximal compatibility classes of a finite set endowed with a compatibility relation. To derive these classes for an n- set, when n is relatively small; it can be obtained by means of a compatibility diagram. However, for a large n, determining maximal compatibility classes by means of diagrams becomes difficult to visualize. Moreover, in order to compute maximal compatibility classes, a diagrammatic representation would not suffice because computers are not good at viewing pictures, etc. In view of this, a necessity to have an algorithm arises. Consequent to this, to our knowledge, an algorithm for determining the maximal compatibility classes of a compatibility relation was first presented in (Grasselli, 1966) and recently, in (Biswas, 1989).

1.2 Statement of the Problem

This research work endeavours to identify some new properties of compatibility classes of a finite set on which a suitable compatibility relation is defined, construct an alternative algorithm to compute maximal compatibility classes and demonstrate their applications in solving decentralization problems of a switched network.

1.3 Motivation for the Research

In view of having found numerous applications of a compatibility relation in different fields, in particular, that of maximal compatibility classes, we found it significant to study and examine its applications to a switched network.

1.4 Objectives of the Research

The main objective of this research is to construct an alternative algorithm to compute maximal compatibility classes of a compatibility relation defined on a finite set, and apply it to a switched network.

1.5 Methodology

The exploratory research approach is adopted in this study. A survey of the works on compatibility relation and its application will be conducted, properties of compatibility classes will be identified, and an alternative algorithm to compute maximal compatibility classes will be constructed. The data used in this research work is largely gathered from published academic works on this subject.

1.6 Organization of the Thesis

The thesis is organized as follows:

Besides the background of the study, objective of the research and methodology given in chapter one, chapter two provides a detailed literature review on the subject matter.

Chapter three provides some fundamentals of the theory of relations, mathematics of compatibility relation, characteristic properties of maximal compatibility classes and an algorithm to compute them.

Chapter four presents an application of compatibility relation to a switched network.

Chapter five gives conclusion and recommendations.

CHAPTER TWO

LITERATURE REVIEW

2.1 Literature Review

The notion of *relation*, besides being a linguistic phrase has been at the core of logic and mathematics. Roughly, it could be traced to De Morgan (1860), Schroder (1895), Peirce (1933), Tarski (1952), and many others.

Usually, a binary relation can be viewed as *predicates with two unknowns, or as sets of pairs of objects*. Binary relation is a special case of n- ary relation. The calculus of binary relation developed by De Morgan (1860) has been applied to so many areas of Computer science. For example, the Tarski relational algebra is applied in Structural Manipulation of software Architecture (Holt, 1998). In this research work, we have narrowed down our study of binary relation to a special kind of relation called *compatibility* relation. In the recent years, it could be traced to the doctoral thesis (On reflexive symmetric relations and graphs) of Duro Kurepa (1952-1953).

As mentioned in the previous chapter, applications of compatibility relation abound in different fields.

Paull and Unger (1959) have studied compatibility relation and its applications to incompletely specified *sequential switching functions*. It was observed that forthe class of functions which are not completely specified, the flow tables will have blank entries corresponding to the unspecified outputs or the unspecified next state conditions (or both).In the same work, Paull and Unger (1959) made useof the concept of minimization and presented a systematic procedure for finding the set of maximum compatibles. From the maximum compatibles (which we may also call MCC), they resortto the process of enumeration in order to find the minimum *closed* set of compatibles. The

enumeration process entails a trial and error procedure in order to eliminateoverlapping states in different maximal compatibles.

In (McCluskey, 1962), a problem of obtaining minimum-state sequential circuits for incompletely specified flow tables, in particular, the case when only maximum compatibles need to be considered in forming the minimum state sequential circuit was discussed. Although, McCluskey (1962) observed that for some special type of flow tables, the compatibility relation has certain properties which are not generally satisfied for arbitrary flow tables. In the same work, it was observed that, it is this special property of compatibility relation which form basis for simplified procedure to be derived.

In theory of evidence (*Dempster- Shafer*(D-S) Structure, 1967), *Dempster's* rule results in the development of uncertainty modeling framework which can be used to provide a generalized probability by allowing the imprecise knowledge of probabilities (Yager, 1990). This knowledge about the probabilities of the various events is usually expressed in terms of an interval in which it lies. In most cases, this expression lacks specification. Consequent on this, the notion of compatibility relation is been used within D-S framework to determine allowable solutions to the primary variable, particularly, when knowledge about one (primary) variable is required, given an information about another (secondary) variable.

According to Yager (1990), the information about one variable can be obtained using information about another variable if we have some knowledge about how they are related; this forms the basis of most inference systems. For an illustration, assume U and V are two variables taking their values in the sets X and Y respectively. In D-S structure, compatibility relation is used to represent this knowledge by defining a relation between U and V on $X \times Y$ to a binary set $\{0, 1\}$, such that for each $x \in X$ there exists at least one $y \in Y$ such that $x\ and\ y$ are compatible and for each $y \in Y$ there exists at least one $x \in X$ such that $y\ and\ x$ are compatible. This is represented by:

$A_i = \{y \mid R(x_i, y) = 1\}$ and $B_j = \{x \mid R(x, y_j) = 1\}$, with the requirement that $A_i \neq \emptyset \; and \; B_j \neq \emptyset$. Moreso, monotonic compatibility relation was introduced and thereby, extended to allow for the representation of non- monotonic knowledge in the framework of D-S theory. Following Yager (1990), a compatibility relation is called monotonic if an increase in information about the primary variable cannot result in a loss of information about the secondary variable.

Further, in *possibility* theory, Kramosil (1995) applied compatibility relation to *belief* functions. In the same work, a compatibility relationis generalized to a relation between sets of states of a system say S and sets of empirical values of an appropriate experiment say E. In this context, Kramosil (1995) defined compatibility relation between internal state s of a system S and an empirical value x of an experiments as thus; s is compatible with x if the possibility that s is the actual internal state of the system when x was observed cannot be logically avoided on the grounds of the laws governing the system and its environment. More generally, the concept of compatibility relation is used to resolve the question- whether $s \in T$ holds for some proper subset $T \subset S$, if $x \in F$ holds for some subset $F \subset E$. In such case, the compatibility relation R is a total *mapping* taking the Cartesian product $S \times E$ into the binary set $\{0, 1\}$ in such a way that for each $s \in S$ and each $x \in F$, $R(s, x) = 0$ holds if and only if the actual internal state s_0 of the system cannot be s due to the laws governing the system itself as well as the source outputting the empirical values, suppose that the empirical value actually obtained is x. In the opposite case, $R(s, x) = 1$ implies the state s is compatible with the empirical value, that is, the possibility that $s = s_0$ cannot be avoided.

In general, the study of compatibility relations and theirapplications is becoming an active area of research.A variety of applications in different fields require the study of compatibility relation. The following list is not meant to be exhaustive:

a) Reduction of the number of internal states in incompletely specified sequential networks (Paulland Unger, 1959) and (Grasselli and Luccio, 1965).

b) Design of certain class of digital control units (Grasselli, 1962).

c) Problems in graph theory (Weissman, 1962).

d) Scheduling problems (i.e., scheduling of lecture periods).

Compatibility relation is seen to be applicable in Symmetrized Max- plus Algebra.In Singh *et al.* (2010),those properties of balances which are only reflexive and symmetric but not transitive were considered. Compatibility relation defined on $M^+ \times M^+$ was introduced, where M^+ depicts Max-plus algebra ($M^+$is a class of path algebras called the *tropical calculus* or *minima algebra*).

Further, it is promising that compatibility relationcould be successfully applied to Fuzzy algebra and Relations (Filep, 1998).

In Tremblay and Manohar (1975), Singh *et al.,*(2010),and others it was observed that in the context of graph theory, the maximal compatibility classes could be viewed as the largest complete polygons where a complete polygon is one in which every vertex is related to every other vertex. This has unfolded a significant perspective of multi- faceted application of compatibility relation. As a matter of fact, compatibility relations play a vital role in graph theory, codes and free monoids (Karki, 2003).

In view of its enormous applications in different fields, we propose to study mathematics of compatibility relations, attempt to investigate the nature of maximal compatibility classes and propose an alternative algorithm for computing maximal compatibility classes.

Finally, in this research work, we propose an application of compatibility relations to a switched network and describe schematically how compatibility relation based network congestion control procedure does work.

The rest of this chapter is abstracted from (Wikipedia, 2012; Rosen, 1999).

2.2 Binary Relations and their Properties

It is well known that binary relations are generalized mathematical functions or mappings. Contrary to a function from X to Y, a binary relation R on $X \times Y$, written as $R(X \times Y)$, may assign to each element of X two or more elements of Y, This is why a relation need not be a function. Some basic operations on functions such as inverse and composition are applicable to binary relations as well.

2.2 Definition

Let X and Y be non- empty sets. Abinary relation R or, simply, a relation from X to Y is a subset of$X \times Y$, where$X \times Y = \{(x, y) \mid x \epsilon X \text{ and } y \epsilon Y\}$.

Suppose R is a relation from X to Y. Then R is a set of ordered pairs where each first element comes from X and each second element comes from Y. That is, foreach $x \in X$ and each $y \in Y$, exactly one of the following is true:

i. $(x, y) \in R$; we then say that x *is R-related to y*, written as xRy

ii. $(x, y) \notin R$; we then say that x *is not R-related to y*.

The domain of a relation R from X to $Y$is the set of all first elements of the ordered pairs which belong to R, so it is a subset of X; and the range is the set of all second elements, and so it is a subset of Y.

Sometimes R is a relation from a set X to itself, that is, R is a subset of $X^2 = X \times X$. In such case, we say that R is a relation on X.

Examples 2.1

1. Let Z and Z^+ be the set of integers and the set of positive integers respectively. For $x, y \in Z$ and $n \in Z^+$, the modulo n defined by $(x, y) \in R$ if $x - y = kn$ is a relation, where $k$is a scalar in Z^+.

2. Let $x, y \in X$, define $(x, y) \in R$ if x is a prefix of y. This is a relation that assigns letters to words.

We shall use $xRy$to mean x *is related to* y, instead of $(x, y) \epsilon R$.

2.3 Types of Relation

a) Reflexive: A relation R on a set X is reflexive if for each $x \in X$, xRx. That is, every element of X relates to itself.

b) Symmetric: A relation R on a set X is symmetric if for xRy, it is also the case that yRx, where $x, y \epsilon X$.

c) Transitive: A relation R on a set X is transitive if xRz, whenever xRy and yRz, where x, $y, z \epsilon X$.

d) Anti-symmetric:A relation R on a set X is called anti-symmetric if xRy and yRx imply$x = y$, where $x, y \in X$.

e) Asymmetric: A relation R on a set X is asymmetric if xRy but not yRx.

f) Irreflexive: A relation R on a set X is irreflexive if x does not relate to itself, for$x \in X$.

Example 2.2

Examples on the types of Relations.

i. Let$X = \{1, 2, 3, 4\}$, a relation $R \subseteq X \times X$ is reflexive if and only if $R = \{(1, 1), (2, 2), (3, 3), (4, 4)\}$. Observe that; given a set X, with $|X| = n$, we have $|X \times X| = n^2$. So that the number of relations on $X$is equal to the cardinal of the set of $X \times X$. And it is exactly 2^{n^2}. Among them $2^{n^2-n}$is reflexive.

ii. Let $X = \{1, 2, 3\} \in N$, the relation R on X given by: $R = \{(1, 2), (3, 1), (2, 1), (1, 3)\}$ is a symmetric relation.

iii. Let Z^+denote the set of positive integers. The relation R defined on Z^+byxRy if x divides y is a transitive relation on Z^+ which need not be symmetric.

iv. Let X and Y be non-empty sets. The relation "$\subseteq$ "is anti- symmetric. For if $X \subseteq Y$ and $Y \subseteq X$, then $X = Y$.

v. Let the relation R mean "is a father of". Then R is irreflexive. For if a is a human being a cannot be a father of a.

Remark2.1

For a given set $= \{a_1, a_2, a_3, \dots, a_n\}, A \times A = A_1 \cup A_2$, where $A_1 = \{(a_i, a_i): i = \{1,2,3,\dots,n\}$ and $A_2 = \{(a_i, a_j): i \neq j = 1,2,3,\dots,n\}$. A_1contains n- pairs and A_2contains $(n^2 - n)$- pairs. A_2 contains $1/2(n^2 - n)$ subsets $S_{ij}$of the form $\{(a_i, a_j), \{(a_j, a_i): i \neq j\}$. And so, we have totally $n^2 \times 2^{1/2(n^2 - n)}$ symmetric relation on A.

Remark 2.2

i. Asymmetric is not the negation of symmetric.

ii. Irreflexive is the negation of reflexive.

iii. Anti- symmetric is not the negation of symmetric.

2.2 Definition

A relation R on a set X is called an equivalence relation on X if it is reflexive, symmetric and transitive.

Examples 2.3

1. For a finite set $X, X \times X$ is the largest equivalence relation on X. The equality relation is the smallest equivalence relation on X.

2. Let f: $X \longrightarrow Y$ be a function from X to Y, define a relation R on X by xRy if and only if $f(x) = f(y)$. Then R is an equivalence relation on X.

If for $x, y \in X$, we have xRy or yRx. Then the relation R is said to be *connected*.

Remark 2.3

The properties of relations discussed above are not independent of each other. An irreflexive and transitive relation is always an asymmetric relation, and every asymmetric relation is always irreflexive.

Definition 2.3

Let X, Y, Z be non- empty sets and $R_1 \subseteq X \times Y$. The composite relation R of R_1 *and* $R_2 \subseteq Y \times X$denoted by $R = R_1 \, o \, R_2$, is a relation defined $by\ R = \{(x,z): \exists\ y \in Y: (x,y) \in R_1 and\ (y,z) \in R_2\}$.

Remark2.4

The composite relation is associative.

2.4 Power of Relation

We define the power of relation R inductively by:

a) $R^1 = R$

b) $R^{n+1} = R \, o \, R^n, n \geq 1$.

Example2.4

If $R = \{(1,2),(1,3),(2,4),(3,2)\}$, then R^2 is given as: $R^2 = \{(1,4),(1,2),(3,4)\}$ and R^3 will be: $R^3 = R \, o \, R^2 = \{(1,4)\}$.

2.5 Representation of Relation

A binary relation R on a set X can be represented in the following forms.

1) Matrix form
2) Graphical form
3) Sagittal diagram

2.6 Matrix representation of Relation

Example 2.5

Consider the relations: $R_1 = \{(1,x),(2,x),(3,y),(3,z)\}$ and $R_2 = \{(w,5),(x,6)\}$.

The matrix representation of R_1denoted by $M\ (R_1)$ is given by:

$$M(R_1) = \begin{array}{c} \\ (1) \\ (2) \\ (3) \\ (4) \end{array} \begin{array}{c} \begin{array}{cccc} (w) & (x) & (y) & (z) \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}.$$

Similarly,

$$M(R_2) = \begin{array}{c} \\ (w) \\ (x) \\ (y) \\ (z) \end{array} \begin{array}{c} \begin{array}{ccc} (5) & (6) & (7) \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

Remark 2.5

For any two given relations R_1 and R_2, we have $M(R_1).M(R_2) = M(R_1 o R_2)$. Let the matrices A *be an* $n \times m$ *and* B *be an* $m \times r$, then the composition, AoB denoted C, is given by $C_{ij} = \bigvee_{k=1}^{m}(a_{ik} \wedge b_{kj}), i = 1,2,\dots,n; j = 1,2,\dots,r$, where $\wedge$ is bit- ANDing and $\vee$ is bit- ORing (i.e., as defined in standard logic used in computer science).

To see this, consider the matrices below.

$$M(R_1).M(R_2) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{array}{c} \\ (1) \\ (2) \\ (3) \\ (4) \end{array} \begin{array}{c} \begin{array}{ccc} (5) & (6) & (7) \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

$= M(R_1 o\, R_2)$, Since $R_1 o\, R_2 = \{(1,6),(2,6)\}$.

Remark 2.6:

Let X be a finite set, and R a relation on X. If $M(R)$ is the relation matrix for R, then

a) $M(R) = 0$, iff $R = \emptyset$

b) $M(R) = I$, if and only if $R = X \times X$

c) $M(R^m) = [M(R)]^m$

2.7 Graphical Representation of Relation

A relation can be represented by a graph. This is done by considering the set X on which the relation is defined, as the vertex set and the relation R ($\subseteq X \times X$) as the edge set.

Example2.6: Let $X = \{1, 2, 3, 4, 5\}$ and $R = \{(1, 1), (1, 2), (1, 4), (3, 2)\}$. Then R can be represented graphically as seen below:

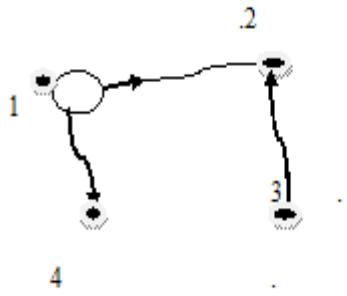


Figure 2.1: Graphical representation of relation

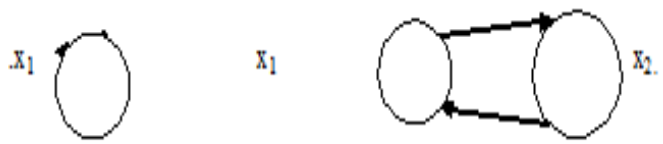


Figure 2.2: Representation of Reflexive and Symmetry relation

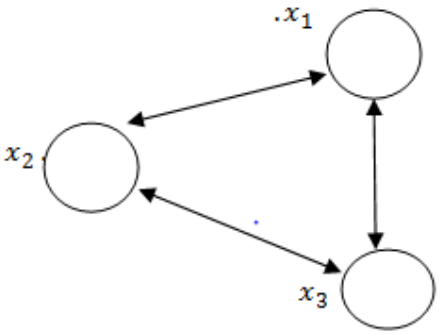


Figure 2.3: Graphical representation of reflexive, symmetric and transitive relations.

Remark 2.7

If a relation R is represented graphically, then its terms strongly apply to the relations defined on the set. Therefore, we have the following:

1) $R$is reflexive if and only if its directed graph contains a loop at each vertex.
2) $R$is symmetric if and only if its digraph may be drawn only by loops and undirected edges.
3) R is anti- symmetric if and only if for $x \neq y$ the graph contains at most one of the edges (x, y) or (y, x).

4) $R$is an equivalence relation if and only if its graph consists of disjoint union of complete graphs augmented by loops at each vertex.

2.8 Representing Relation(s) using Sagittal diagram

Another useful representation of binary relation is a *sagittal* diagram. Each of the sets X *and* Y is represented by a set of nodes. Elements of $X \times Y$ with non- zero membership grades in R are represented in the diagram by lines connecting the respective nodes. Each of these lines represents the value of the relation of the elements.

We will illustrate the sagittal diagram using *fuzzy* idea (as shown below).

Let $X = \{x_1, x_2, x_3, \dots, x_7\}$ and $Y = \{y_1, y_2, \dots, y_5\}$and the relation $R = \{(x_1, y_1) = 0.6, (x_1, y_2) = 0.7, (x_1, y_3) = 0.2, (x_2, y_2) = 0.4, (x_2, y_4) = 0.5, (x_3, y_1) = 0.2, (x_4, y_3) = 0.1, (x_4, y_4) = 0.3, (x_5, y_4) = 0.1, (x_5, y_5) = 0.2, (x_6, y_4) = 0.6, (x_6, y_5) = 0.7, (x_7, y_1) = 0.8, (x_7, y_2) = 0.8, (x_7, y_5) = 0.5\}$, where $R: X \times Y \to [0, 1]$ is the mapping defined by $R(x, y) = r \in [0, 1]$.

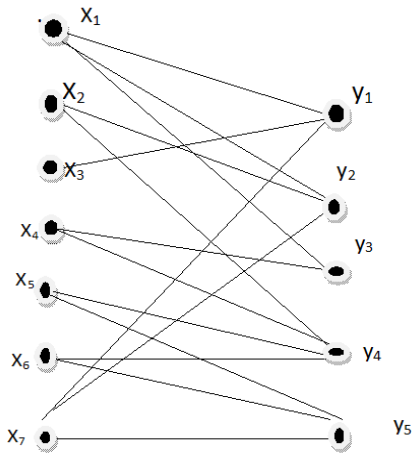


Figure 2.4: Sagittal diagram.

CHAPTER THREE

COMPATIBILITY RELATION AND SOME RELATED CONCEPTS

3.1 Compatibility Relation

A compatibility relation is a relation defined among elements of a set $S = \{x_1, x_2, \dots, x_n\}$. We shall indicate that the relation holds between elements $x_i$and x_j by writing: $x_i \approx x_j$. A compatibility relation is both reflexive and symmetric, i.e.:

a) For any i: $x_i \approx x_i$(reflexive)

b) For any i, j:$x_i \approx x_j$. implies: $x_j \approx x_i$(symmetric).

Elements x_i and x_j are said to be compatible whenever:$x_i \approx x_j$holds.

Let S be a set with n elements, usuallyannoted$n$- set. A family $\{A_1, A_2, \dots, A_n\}$of non- empty subsets of S is called a *covering* of Sif$S = \bigcup_{k=1}^{n} A_k$. For example, the set $S_1 = \{a\}$ has only a single cover, namely, {{a}}. However, there are five covers of the set $S_2 = \{a, b\}$, namely,$\{\{a\}, \{b\}\}, \{\{a, b\}\}, \{\{a\}, \{a, b\}\}, \{\{b\}, \{a, b\}\}$ and $\{\{a\}, \{b\}, \{a, b\}\}$. Note that A_k's are not necessarily disjoint and hence, it may not define a partition.

A cover of a finite set S is called *minimal* if none of its proper subclasses covers S.

Note that compatibility relation, not being necessarily transitive, may not define a partition. However, it does define a covering (Singh *et al*., 2010). Essentially, a compatibility relation defined on a finite set decomposes the set into its possibly pairwise non-disjoint subclasses, henceforth called *compatibility classes* (CC*)*. It follows that the elements of a compatibility class are pairwise compatible.

A compatibility classC_b is a subclass of S such that all the members of C_b are pairwise compatible: i.e., :$x_i \approx x_j$holds if$x_i$and x_j are in C_b. A compatibility class is *maximal* if it is not a proper subclassof any other compatibility class (Grasselli, 1962).

The compatibility relation on a set $S = \{x_1, x_2, \dots, x_n\}$can be represented by means of a graph G (*compatibility graph*) having nodes $x_1, x_2, \dots, x_n$and edges (x_i, x_j)whenever$x_i \approx x_j$. One such graph is drawn below:

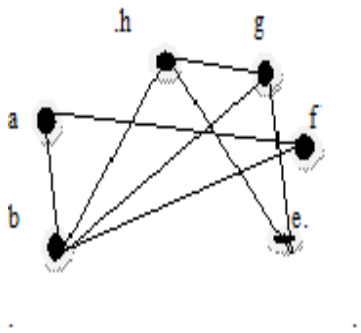


Figure 3.1: Compatibility graph.

Self- edges are excluded, since they carry no information.

Let a subgraph $G_i$of G be *complete* if every couple of nodes of G_i is joined by an edge; a complete subgraph corresponds to a compatibility class. For example, in figure 3.1 nodes a, b, f and the edges joining them constitute a complete subgraph. Let a complete subgraph be *maximal* if it is not a subgraph of any other complete subgraph. In figure 3.1 nodes h, g, e and the edges joining them constitute a maximal compatibility class. Thus the determination of the collection of maximal compatibility classes coincides with the determination of the set of maximal subgraphs of G.

Alternatively, a subclass$M$ of S is called a *maximal* compatibility class if any element of M is compatible to every other element of M and no other element of $S - M$is compatible to all its elements.

Graphically, a maximal compatibility class for a given compatibility relation R can also be viewed as the largest complete polygon in the graph of R. A polygon is complete if each of its nodes is related to all other nodes. A triangle is always a complete polygon and, for a quadrilateral to be a complete polygon, we need both the diagonals. Note that a complete polygon is a compatibility class, which need not be a maximal compatibility class, unless it satisfies the criteria of being so. Also, any element of S that relates only to itself is a maximal compatibility class, and any two elements of S which are compatible to one another but to no other elements of S form a maximal compatibility class (Singh *et al.*, 2010). The set of all compatibility classes is called maximal compatibility class for a given relation.

While the graphical representation of a compatibility graph is possible for small problems to visualize the relation, it becomes rapidly unmanageable when the problems grow large (Grasselli, 1965).

Remark 3.1

If $R_1$and R_2 are compatibility relations on an n- set S, then $R_1 \cap R_2$ (if it exists) is a compatibility relation on S.

For an illustration, let $R_1 \cap R_2$ exist. We only need to show that intersection of a

a) Reflexive relation is reflexive, and
b) Symmetric relation is symmetric

a.) : By defining a reflexive factor $R = \{(x,x): xR_1x \; and \; xR_2x\}$, for all $x \in S$. Since $(x,x) \in R$ whenever$(x,x) \in R_1 and \; (x,x) \in R_2$. It becomes clear that $R \subseteq R_1 \cap R_2$. And by definition of R, it follows that (a) holds.

b.) : Define $S^* = \{(x,y): (x,y),(y,x)) \in R_1 \; and \; (x,y),(y,x)) \in R_2, for \; x,y \in S\}$. This means that, for any $(x,y) \in S^*$, we have $(y,x) \in S^*$. By definition, $S^* \subseteq R_1 \cap R_2$. It follows that $S^*$is a symmetric relation. And so, (b) holds.

Here $R \; and \; S^*$ are the reflexive and symmetric relations contained in $R_1 \cap R_2$ respectively. R and S^* together form $R_1 \cap R_2$. Hence by definition of compatibility relation, we have that $R_1 \cap R_2$ is a compatibility relation.

From the above remark, it is in order to say that the intersection of all compatibility relations on an n- set S, if it exists, is the smallest compatibility relation on S.

Obviously, all equivalence relations are compatibility relations. We shall, however, be concerned here with those compatibility relations which are not equivalence relations. For an illustration, let us consider the following example:

Example 3.1

Let $S = \{\{2,1,6,6\},\{2,4,3,7,8\},\{3,7,5\},\{8,7,5,4,3,2\},\{4,5,7,8,9\}\} = \{x_1, x_2, \ldots, x_5\}$, where $x_1 = \{2,1,6.6\}$, $x_2 = \{2,3,4,7,8\}, \ldots etc$, respectively; and R be given by:

$R = \{(x_i, x_j) | x_i, \; x_j \in S \wedge x_i R x_j \; if \; x_i \; and \; x_j \; contain \; some \; common \; elements\}$. Clearly, $\{2,1,6,6\} \; R \; \{2,4,3,7,8\} \; and \; \{2,4,3,7,8\} \; R \; \{3,7,5\} \nRightarrow \{2,1,6,6\} \; R \; \{3,7,5\}$. That is, R is not transitive.

Note that the elements of S could be *multiset* as well (a multiset is a collection in which objects are allowed to repeat).

The following is a simplified graph for R.

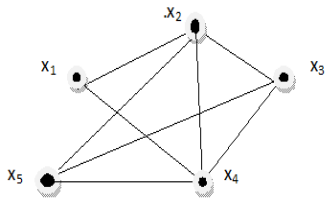


Fig. 3.2: Simplified graph of a compatibility relation R.

Since R is reflexive and symmetric, loops at each node are not drawn and only one of x_iRx_j and x_jRx_i is drawn.

From the diagram, it follows that the elements in each of the classes$C_1 = \{x_1, x_2, x_4\}$, $C_2 = \{x_2, x_3, x_4\}, C_3 = \{x_2, x_4, x_5\}$, $C_4 = \{x_3, x_4, x_5\}$ and $C_5 = \{x_2, x_3, x_4, x_5\}$ are mutually compatible and the classes are not mutually disjoint. These are the compatibility classes of S. The class $\{x_1, x_3\}$ does not have compatible elements. Clearly, $M_1 = \{x_1, x_2, x_4\}$ and $M_2 = \{x_2, x_3, x_4, x_5\}$ are the only maximal compatibility classes and these two classes define a (minimal) covering of S. Observe that x_2Rx_5 holds, x_2 belongs to both $M_1$and M_2, but x_5 belongs to M_2 and does not belong to M_1.

3.2 Characteristic Properties of Compatible Classes

Let S be an n- set $\{x_1, x_2, \dots, x_n\}$ on which a compatibility relation R is defined. In other words, for every x_i there exists an x_j such that x_iRx_j. Note that the x_i's may be representing *nodes* in a network system or *states* in a finite machine or *vertices* in a graph.

The following conditions hold:

(i) There must exist a family $\{A_1, A_2, \dots, A_n\}$ of non-empty compatibility classes of S such that $S = \bigcup_1^n A_k$. It is to be observed that every subclass of S need not be a compatibility class.

(ii) There must exist a family $\{M_1, M_2, \ldots, M_i\}$ of maximal compatibility classes of S such that $\bigcup_{k \leq i} M_k = S$. Moreover, the least number of M_k's that cover S will constitute a minimal cover of S. Note that the number of minimal covers of S may be more than one.

This prompts us to conclude that an R which guarantees a family of minimum number of maximal compatibility classes that cover S would be a preferable choice for application in network systems. However, the choice of such an Rcalls for a tradeoff between a stability and the control of the system.

(iii) Every element of S must be an element of at least one of the maximal compatibility classes of S. Also, whenever $x_i R x_j$, then $\{x_i, x_j\}$ must be a subclass of at least one maximal compatibility class of S. Note however that, in general, for a pair of compatible nodes, the inclusion of one of them in a maximal compatibility class does not necessarily imply the inclusion of the other.

The following results hold:

(iv) For any compatibility class C of S, either C is itself a maximal compatibility class or C is a subclass of some other maximal compatibility class of S. It can be seen as follows:

Let C be a compatibility class of S. If C contains only one element of S, the proof is trivial by definition of a maximal compatibility class. Otherwise, let C contain some k (> 1) elements of S. Suppose $S - C = \emptyset$, the result follows immediately.

Now, let $S - C \neq \emptyset$, then there exists at least one element $x \in S$ such that one of the following holds:

(a) x is not compatible to any element of C

(b) x is compatible to some element of C, but not all.

In either case, (iv) holds.

(v) Let $\{M_1, M_2, \ldots, M_i\}$ be a family of all maximal compatibility classes of S, then M_i's are pairwise incomparable with respect to $\subset$.

(vi) S has at least two maximal compatibility classes $M_{i=1,2}$ if and only if for $x_i, x_j \in S$ such that $x_i R x_j, x_i \in M \implies x_j \notin M$.

(vii) Any compatibility class of S with a maximum cardinality is a maximal compatibility class.

To see this, let M be a compatibility class of S with maximum cardinality.

Suppose that M is a compatibility class of S with maximum cardinality, which is not a maximal compatibility class. Then by (iv), M must be a proper subclass of some other maximal compatibility class. This means that there exists a compatibility class M^* containing at least an element $x \notin M$ which is compatible to every element of $M \subset M^*$. And so, $|M^*| > |M|$. This contradicts the fact that M is of maximum cardinality. Hence M must be a maximal compatibility class.

Example 3.1 above verifies the characteristic properties presented above.

In sequential circuit theory, where only maximal compatibility classes need to be considered in forming the minimum- state sequential circuit, aforesaid observations would guarantee the inclusion of internal state S_j^{α} in at least one of the maximal compatibility classes whenever it contains internal state S_i^{α} of a particular type of flow table, for S_i^{α} and S_j^{α} both stable (for input state X^{α}) and compatible (McCluskey, 1962).

3.3Concepts of Compatible states in Sequential Machines

According to William (1962), a compatible is a set of states of a *flow table* that constitute one member of some *closed* grouping. The entries in the implication table are made in the following manner.

1) If there is a contradictory pair of outputs for a given row-pair in any input column, then an X is placed in the corresponding cell of the implication table and no further attention need be given to that row-pair. This is true for (13), (35), and (57) in Table IB.

2) If for a row-pair ab the outputs are not contradictory, then enter all row-pairs implied by ab in the ab cell. (Self-implication need not be entered into the table). Thus in cell (12), (23) is entered; and in cell (15), (26) is entered, etc. In cells (14), (37), (45), (46), and (67), there are no pairs to be entered, so a check mark is inserted.

A second inspection of the implication table is often necessary. If a row-pair ab is an incompatible, then an X is placed in all cells that imply ab, and any other entries in such cells will be ignored. This process is to be repeated until no other incompatible row-pairs are found. A check of the Implication Table IB indicates no other modifications to be made; thus it is also the final implication table. All row-pairs with entries which are not X's are compatibles, and those with X's are incompatibles.

Table IA: flow table

N	I_1	I_2	I_3
1	21	– –	– –
2	3 –	5 –	4 –
3	– 0	10	7 –
4	– –	2 –	– –
5	6 –	–1	40
6	4 –	– –	3 –
7	4 –	1 –	– 1

Table IB: compatibility chart

2	23					
3	X	47 15				
4	/	25	12			
5	26	36	X	/		
6	24	34	37	/	46 34	
7	24	15 34	/	12	X	/
	1	**2**	**3**	**4**	**5**	**6**

From Table IB the maximal compatibles can be found (as shown in table II).

A maximal compatible (MC) is defined as a set of rows which form a compatible class and which is not included in any larger compatible class.

A procedure presented by Paull and Unger (1959) for finding the MCs, will be stated. The procedure is based on the use of incompatible pairs to break down large sets of rows into smaller sets until only MCs remain. Start with Column **1** of the implication table and work to the right.

1) Write down a set of all states except **1.** Then write down a set of states consisting of **1** followed by all states corresponding to non-X rows in Column **1.** In our example, we would have at this point (234567) and (124567).

2) Move to the next column containing X's. If ab is an incompatible, then those candidate sets containing both a and b must be split into two sets, one with a omitted and the other with b omitted. The process is repeated through the last column.

In our example the process continues from the candidate sets from Column 1 as follows**:**

1) (234567) (124567).

2) No change.

3) (24567) (23467) (124567).

4) No change.

5) (2456) (2467) (23467) (12456) (12467).

The maximum compatibles are (23467), (12456) and (12467). The sets (2456) and

(2467) are subsetsof the others. Thus they cannot be MCs. Subsequently, we will be concerned with the methodof finding the smallest collection of compatibles without enumeration. By replacing a compatible with a subset of itself, the reduced flow table will, in general, have more unspecified entries. A procedure will be outlined utilising some set of rules which eliminate the overlapping states, where possible.

3.4 Maximal Compatible Chart

For a given flow table, not all of the derived MCs are required to obtain a *closure* (or a covering).This section discusses a suitable method for an appropriate selection of the MCs from the MC chart which would cover all the states in the flow table.

In order to select a suitable minimum set to cover all the states from the group of MCs, the maximum compatible chart is constructed in the following manner. In Table II each column carries a number at the top corresponding to one of the states in the flow Table IA. Each row corresponds to one of the MCs derived insection3.3, as identified by the letters A, $B$and C in the chart. In each row, a cross (X) is placed under each internal state contained in the MC represented row. Thus in Table I, the MC A contains states 2,3,4,6 and 7, so that the crosses in the A row are in the respective columns. The remainder of the chart is completed in a similar manner.

Table II: maximal compatibility chart

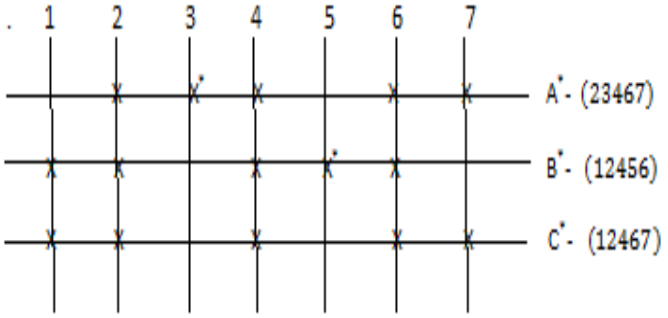
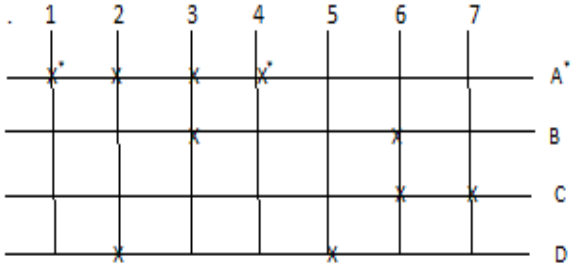


Table III: maximal compatibility chart



An inspection of the MCs shows that MCs *A*and *B* are necessary, since the internal states 3 and 5are only included in MCs *A* and *B*, respectively. This is exhibited very clearly from the MC chart, since Columns 3 and 5 have exactly only one cross in them. If an asterisk is placed on each of the crosses which stand alone in a column, then each row that contains an asterisk cross will be called an *Essential Maximum Compatible* (EMC). A single asterisk is placed at the end of each row corresponding to an *MC* which is essential. In this example a check of the EMCs thus indicates a coverage of all the Internal states of the flow table. However, if the EMCs do not cover all the internal states, such as in Table III, it is not obvious which of the other combinations of MCs is to be chosen in order to insure closure.

This problem is discussed in details in (William, 1962).In any case, one or more choices of the other MCs are needed to insure full coverage of internal state such a choice of MC will be called a *Secondary Maximum Compatible* (SMC).

An interesting configuration of an MC chart arises when a flow table has a set of incompatibles as shown in Table IV A.The pairwise incompatibles are (12), (34) and (56).Thus the MCs are (245). (246), (136), (135), (235), (236), (145) and (146). A careful inspection of the MC chart Table IV B shows that there is exactly the same number of crosses in each column, thus making the choice of EMCs and SMCs rather difficult; however, this type of MC occurs in very special cases of the flow table. The selection of the appropriate compatibles can still be made from an expanded table called the *closure table* defined in (William, 1962). The MC chart with the above properties is called a *cyclic chart*.

Following William (1962), the necessary and sufficient condition for a chart to be cyclic is that the flow table must have an even number of internal states (m),and the pairwise incompatibles must occur in ascending order- (12), (34) .... (-m).

Table IV A: compatibility chart

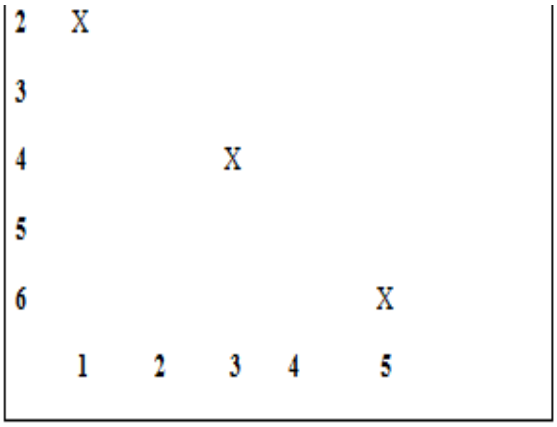
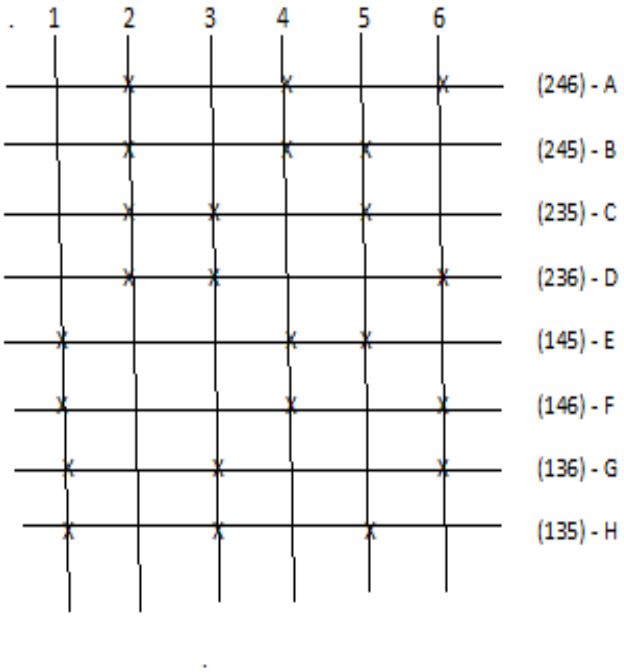


Table IV B: maximal compatibility chart



The sufficiency requirement places the following restrictions on the flow table:

1) No other row-pairs may imply an incompatible pair (12), (34) ... (-m). The incompatible pairs may imply any combination of the internal states in the cell, the implied states in this case do not affect the determination of MCs.

2) In any one of the input columns of the flow table there must be only one contradictory pair of output conditions. They must occur in the positions of (12) or (34) or ... (-m). Thus all but two of the output entries in any input column must be unspecified.

The restrictions imply the existence of a lower limit on the number of inputs for a given flow table. If there are more inputs than the lower limit, then all of the output entries in some of the input columns must be unspecified or they must be identical to the output entries in some of the other columns.

From the technique to compute maximal compatibility classes presented in Paull and Unger (1952); Grasselli (1962) and Biswas (1989), we thought of constructing the following algorithm (in section 3.6).

3.5 Relation matrix of a Compatibility Relation and Construction of an algorithm to compute MCC

In order to construct the relation matrix of a compatibility relation R, for each pair of nodes $(x_i,\ x_j)$, a one (1) is assigned to it if $x_i R x_j$ and, a zero (0) if it is not the case that $x_i R x_j$. Consequently, the table for the relation matrix R has an all $-$ 1 leading diagonal (due to reflexive property of R). Moreover, since the relation matrix of R is symmetric, it is sufficient to tabulate only the elements of its lower triangular parts. However, we will consider the complete matrix for discussion in this thesis.

The following compatibility matrix is a 0 $-$ 1 representation of the compatibility chart presented in (Paull and Unger, 1959).

Table A: Compatibility table.

	x_1	x_2	x_3	x_4	x_5	x_6
x_2	1					
x_3	0	1				
x_4	1	1	1			
x_5	1	1	0	1		
x_6	1	1	1	1	1	
x_7	1	1	1	1	0	1

3.1 Definition

(a) A candidate maximal compatibility class (CMCC) is a compatibility class (CC) if its submatrix is all-1, which is an MCC if there does not exist any state (element) in its complement which is compatible with all its constituent states.

3.6 Algorithm to compute MCCs from CMCCs

We consider the following results due to Biswas (1989).

Theorem3.5.1

A candidate maximal compatibility class (CMCC) is a maximal compatibility class (MCC) if and only if it is one of the least cardinality CMCC among all the CMCCs and all its constituent states are pairwise compatible.

Proof: Sufficiency: If all the constituent states of a CMCC are pairwise compatible, then the CMCC is an MCC if it is not contained in a larger MCC. Now, since the CMCC has the least cardinality, the generating state (i.e., a state which is compatible with only and every constituent state of that CMCC) of the CMCC has all its compatible states included in the CMCC and therefore there can be no MCC with greater cardinality which may contain the generating state within it. Therefore the CMCC cannot be contained in another larger MCC. Hence the CMCC is an MCC.

Necessity: It is obvious that the two conditions which are sufficient are also necessary to make a CMCC an MCC.

Corollary 3.5.1

A CMCC of cardinality 1 or 2 is an MCC.

Proof: Obvious.

Theorem 3.5.2

A CMCC with the least cardinality which is itself not an MCC will generate more than one MCC with the generating state as one of its members.

Proof: The CMCC with the least cardinality fails to be an MCC only when one (or more) pairs in it is (are) not compatible. Therefore, each of the subclasses which produces an all-1 matrix has all its states pairwise compatible. Since the subclass is not contained in a larger subclass, it is an MCC.

Corollary 3.5.2

If there are two and only two zeros in the sub-compatibility matrix of a CMCC, then the CMCC produces two maximal compatibility classes.

Proof: If the element a_{ij} of a sub-compatibility matrix (SCM) is a 0, then the element a_{ji}will also be a 0, since the SCM is a symmetric matrix. Now, this pair of zeros can be eliminated from the SCM in

two ways, by either the row or column *i*or the row and column *j*. Hence, the SCM will produce two all-1 submatrices and, therefore two MCCs.

A modification of the algorithm presented in (Biswas, 1989) is applied to the compatibility table IB above presented by Paull and Unger (1959) to obtain the MCCs in tableB.

The following table is due to Biswas (1989).

Table B: Compatibility table (Biswas, 1989).

States	$x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7$	CMCCs	α	MCCs
x_1	1 1 0 1 1 1 1	$\{x_1, x_2, x_4, x_5, x_6, x_7\}$	6	$\{x_1, x_2, x_4, x_5, x_6\}$
x_2	1 1 1 1 1 1 1	$\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$	7	$\{x_1, x_2, x_4, x_6, x_7\}$
x_3	0 1 1 1 0 1 1	$\{x_2, x_3, x_4, x_6, x_7\}$	5	$\{x_2, x_3, x_4, x_6, x_7\}$
x_4	1 1 1 1 1 1 1	$\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$	7	~
x_5	1 1 0 1 1 1 0	$\{x_1, x_2, x_4, x_5, x_6\}$	5	~
x_6	1 1 1 1 1 1 1	$\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$	7	~
x_7	1 1 1 1 0 1 1	$\{x_1, x_2, x_3, x_4, x_6, x_7\}$	6	~

The MCCs in the last column of table B have been obtained by the following steps outlined in the algorithm below. Also, the column α represents the weight (cardinality) of each CMCC in table B and the symbol ~ represents repeated MCC.

Following the procedure for obtaining MCCs in (Biswas, 1989), the observations that follow is in order.

Observations:

i. It is possible that two CMCCs of distinct *generators* give rise to the same MCC. In that case, one must be a subclass of the other; we take the largest CMCC and ignore all those which are its subclasses.

ii. The algorithm may produce the same CMCC and hence, the same MCC in different rows. In such cases any one of them could be picked for minimalization purposes and all others are ignored.

Following Biswas (1989), we construct the following matrix below. However, instead of column for weight of CMCCs, we provide a column for cardinality of MCCs. Moreover, we add a column for listing the cardinalities of MCCs in decreasing order.

In table C below, the column β represents the cardinality of MCCs and γ represents the cardinality of MCCs when arranged in decreasing order.

Step1: Compute all CMCCs using matrix table (proposed in Biswas, 1989).

Step2: Begin with the CMCC of row 1. If it is of cardinality 1 or 2, list it in the column of MCCs. The same holds for all other rows of the compatibility table.

Step3: If the CMCC in consideration is of cardinality greater than 2, check whether or not there exists a zero (0) at any intersection of rows and columns of the submatrix generated by the constituent states of that CMCC i.e., if the submatrix is an all-1 submatrix or not. If the submatrix is all-1, list it in the column for MCCs.

Step4: If the matrix in step 3 is not all-1, ignore the row(s) and column(s) containing the largest number of zero (0) and list the truncated matrix, which is all-1, in the column for MCCs. If not repeat step 4 until an all-1 matrix is obtained.

Step5:

a) Compute the cardinality of each MCC in a column.

b) Arrange the cardinality of the MCCs in decreasing order.

For an illustration of the above algorithm, let R be a compatibility relation on a set $S = \{x_1, x_2, \ldots, x_7\}$ defined by the matrix as follows:

Table C: Compatibility table.

States	x_1 x_2 x_3 x_4 x_5 x_6 x_7	CMCCs	CCs	MCCs	β	γ
x_1	1 1 0 1 1 1 1	$\{x_1,x_2,x_4,x_5,x_6,x_7\}$	$\{x_1,x_2,x_4,x_6,x_7\}$	$\{x_1,x_2,x_4,x_5,x_6\}$	5	5
x_2	1 1 1 1 1 1 1	$\{x_1,x_2,x_3,x_4,x_5,x_6,x_7\}$	–	$\{x_1,x_2,x_4,x_6,x_7\}$	5	5
x_3	0 1 1 1 0 1 1	$\{x_2,x_3,x_4,x_6,x_7\}$	$\{x_2,x_3,x_4,x_6,x_7\}$	$\{x_2,x_3,x_4,x_6,x_7\}$	5	5
x_4	1 1 1 1 1 1 1	$\{x_1,x_2,x_3,x_4,x_5,x_6,x_7\}$	–			
x_5	1 1 0 1 1 1 0	$\{x_1,x_2,x_4,x_5,x_6\}$	$\{x_1,x_2,x_4,x_5,x_6\}$			
x_6	1 1 1 1 1 1 1	$\{x_1,x_2,x_3,x_4,x_5,x_6,x_7\}$	–			
x_7	1 1 1 1 0 1 1	$\{x_1,x_2,x_3,x_4,x_6,x_7\}$	–			

Remark 3.2

The last column in table C is introduced to obtain a (minimal) cover (for a system$S$) with as *few* MCCs as possible. It is possible to obtain a (minimal) cover by means ofan MCC whose cardinality appears in the first row of the last column in table C together with an MCC of cardinality next to it or with the MCC in the next instance (i.e., recursively). However, if there is a *freak* MCC (M_k, say) with aminimal cardinality, whose elements are not all contained in any other MCC, the program may list all MCCs of cardinality greater than or equal to the cardinality of M_k (with M_k as well) to obtain a (minimal) cover for the system under consideration.

The algorithm presented above, computes MCCs from CMCCs, but if CCs are given or known, the following algorithm may be used.

We present an alternative algorithm for deriving maximal compatibility classes of a finite set endowed with a compatibility relation from its relation matrix by means of the following compatibility table obtained from the example 3.9 given above:

To start with, it is significant to note that any row or column with a 1 at only one or two place(s) in CT_1 is trivially a maximal compatibility class. Also, a compatibility class with the maximum cardinality is a maximal compatibility class. Hence, $M_2 = \{x_2, x_3, x_4, x_5\}$ is a maximal compatibility class.

Table 1: Compatibility table (CT_1).

	x_1	x_2	x_3	x_4	x_5
x_1	1	1	0	1	0
x_2	1	1	1*	1*	1*
x_3	0	1*	1	1*	1*
x_4	1	1*	1*	1	1*
x_5	0	1*	1*	1*	1

Delete entries of compatibility class C_i if it is a proper subclass of some other compatibility class C_j. Note that deleting an entry 1 does not imply that it is replaced with 0, since 0 is meant to signify incompatibility.

Table 2: Compatibility table (CT_2).

	x_1	x_2	x_3	x_4	x_5
x_1	1	1	0	1	0
x_2	1	1			
x_3	0		1		
x_4	1			1	
x_5	0				1

Delete the entries in a row or column if all entries of that row is zero, except one.

Table 3: Compatibility table (CT_3).

	x_1	x_2	x_4
x_1	1	1	1
x_2	1	1	
x_4	1		1

It follows from the compatibility table (CT_3) that the first row or column gives a maximal compatibility class of S, namely, $M_1 = \{x_1, x_2, x_4\}$.

Accordingly, the algorithm can now be summarized as follows:

Begin

DERIVE_ COMPATIBILITY_ TABLE of the given compatibility relation

LIST_ ALL COMPATIBILITY CLASSES with a 1 at only one or two place(s) of its row or column and

COMPUTE_ COMPAIBILIY CLASSES with the highest cardinality

DERIVE_ NEXT_ TABLE and

IF Compatibility class C_i is a proper subclass of some compatibility class C_j THEN

DELETE_ ENTRIES at the intersection of a row or column inC_i

DERIVE_ NEXT_ TABLE and

IF All but one entry of row r_i is unity, for all i THEN

DELETE_ ROW r_i

COMPUTE_ MAXIMAL COMPATIBILITY CLASSES by collecting nodes in row or column with unit entries, which form non- proper subclass of any compatibility class

End

Similar to the algorithm proposed in (Biswas, 1989) to compute MCC, the number of compatibility tables which need to be derived, will not exceed the number of elements of the largest maximal compatibility class.

In summary, the algorithm constructed in this thesis differs from the ones presented in (Paull and Unger, 1959) in that it does not require the use of incompatiblepairs to break down large setsof rows into smaller setsuntil only MCCsare obtained.

In fact, our future direction is to investigate the complexities and run time of the algorithms constructed in this research work and in (Grasselli, 1962; Biswas, 1989).

Remark 3.3

For a given family of maximal compatibility classes $M_1, M_2, \ldots, M_k$ $(k \leq n)$ of an n- set endowed with a compatibility relation R, a compatibility table (CT)which retrieves R could be obtained and the corresponding family of compatibility classes enumerated.

For an illustration, let $M_1 = \{x_1, x_2, x_4\}$ and $M_2 = \{x_2, x_3, x_4, x_5\}$ be the only maximal compatibility classes of a finite set S. By property (ii), $S = M_1 \cup M_2 = \{x_1, x_2, x_3, x_4, x_5\}$. The maximal compatibility classes can be represented in a compatibility table as follows:

Table A1: Compatibility table (CT_a).

	x_1	x_2	x_3	x_4	x_5
x_1		1	0	1	0
x_2	1		1	1	1
x_3	0	1		1	1
x_4	1	1	1		1
x_5	0	1	1	1	

Adding a loop to each row or column in the compatibility table A (CT_a) gives rise to the following table:

Table B1: Compatibility table (CT_b)

	x_1	x_2	x_3	x_4	x_5
x_1	1	1	0	1	0
x_2	1	1	1	1	1
x_3	0	1	1	1	1
x_4	1	1	1	1	1
x_5	0	1	1	1	1

What remains to enumerate the family of compatibility classes of S is to find candidate subclasses of $M_1 and\ M_2$. To do this, we list the candidate subclasses and check exhaustively for those subclasses which are compatibility classes of S. We obtain the following subclasses:

$\{x_1, x_2\}, \{x_1, x_4\}, \{x_2, x_4\}, \{x_2, x_3\}, \{x_2, x_5\}, \{x_3, x_4\}, \{x_3, x_5\}\ and\ \{x_4, x_5\}$.

It is easy to observe that these candidate compatibility classes exist in CT_b. However, it is important to note that covering problem is concerned with selection of as *few* subclasses as possible from a family of subclasses of S such that their union covers S.

Accordingly, we obtain the second candidate compatibility subclasses:

$\{x_2, x_3, x_4\},\ \{x_2, x_3, x_5\}, \{x_2, x_4, x_5\}\ and\ \{x_3, x_4, x_5\}$.

It may be verified graphically that $\{x_2, x_3, x_4\},\ \{x_2, x_3, x_5\}, \{x_3, x_4, x_5\}\ and\ \{x_2, x_4, x_5\}$ are our desired subclasses generated by a suitable compatibility relation Rdefined on S, whenever $M_1 = \{x_1, x_2, x_4\}\ and\ M_2 = \{x_2, x_3, x_4, x_5\}$ are the only maximal compatibility classes for S.

Thus, the corresponding family of compatibility classes is exactly:

$$\{\{x_2, x_3, x_4\},\ \{x_2, x_3, x_5\}, \{x_2, x_4, x_5\}, \{x_1, x_2, x_4\}, \{x_2, x_3, x_4, x_5\}\}.$$

CHAPTER FOUR

APPLICATION OF COMPATIBILITY RELATION TO SWITCHED NETWORKS

4.1 Introduction

In Copeland *et al*. (1999), some concepts of *Packet- switched* versus *Circuit- Switched* Network is discussed. However, in this chapter, we shall only be concerned with *Packet- switched* Networks.

In this chapter we present applications of compatibility relation (discussed in the previous chapter) to switched networks and obtain some new results

In the following, the term *Network* would stand for *Switched Network*.

4.2 Some Definitions

Let S be a finite set upon which a compatibility relation is defined. We say that a node x_k is a *link* from compatibility class C_i to compatibility class $C_j (i \neq j)\ if\ C_i \cap C_j = \{x_k\}\ for\ x_k \in S$.

Packet queues are *buckets* that hold data for a network device when it cannot be immediately transmitted. Most packet queues use FIFO (first in, first out), unless they have been specially configured to do otherwise. In other words, when the packet queue for a device is completely full, the packet most recently placed in the queue will be sent over the device only after all the other packets in the queue at that time have been sent.

The Open System Interconnection (OSI) model (ISO/IEC 7498-1) is formulated, monitored and augmented by the International Organization for Standardization. It is concerned with characterizing and standardizing the functions of communication system in terms of abstraction layers. Similar communication functions are grouped into logical layers. Each layer serves the layer immediately above it and is served by the layer immediately below it. For example, a layer that provides error-free communication across a network provides the path needed by applications above it, while it calls

immediate lower layer to send and receive packets that make up the contents of that path. Two instances at one layer are connected by a horizontal connection on that layer (see Wikipedia (2013), for some details).

The following is a simplified diagram for the OSI model.

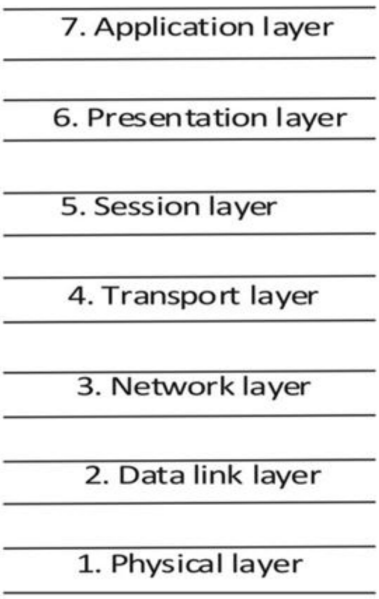


Fig. 4.1: OSI model

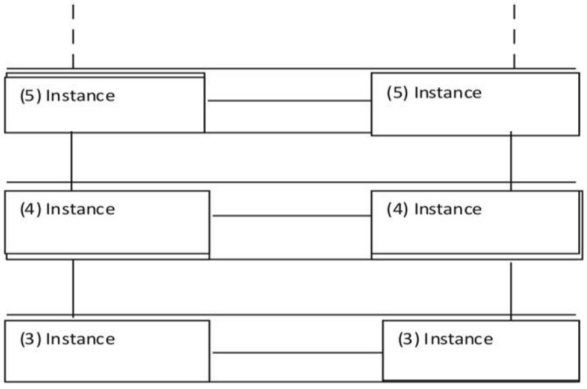


Fig. 4.2: OSI model implementation diagram

4.3 Throughputs

In communication networks, such as Ethernet or packet radio, *throughput* or *network throughput* is the average rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain network node. The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packet per time slot.

The *system throughput* or *aggregate throughput* is the sum of the data rates that are delivered to all terminals in a network. The throughput can be analyzed mathematically by means of *queuing theory*, where the load in packet per-time unit is denoted by *arrival rate* λ, and the throughput in packets per time unit is denoted by *departure rate* μ. Essentially, throughput is synonymous to *digital bandwidth consumption*.

In course of examining throughputs, the term *maximum throughput* is frequently used and end- user maximum throughput tests are considered. Essentially, maximum throughput is synonymous to *digital bandwidth capacity*.

In the context of maximum throughput, it is significant to note that four different *measures*, used for comparing the *upper limit* conceptual performance of multiple systems, are defined: *maximum theoretical throughput*, *maximum achievable throughput*, *peak measured throughput* and *maximum sustained throughput*.

The aforesaid definitions of various concepts are largely abstracted from Wikipedia (2013), and many other works mentioned in the references.

4.3.1 Maximum theoretical throughput

This number is closely related to the channel capacity of the system, and is the maximum possible quantity of data that can be transmitted under ideal circumstances. In some cases the number is

reported as the *channel capacity*, though this can be deceptive as only non- packetized systems (asynchronous) technologies can achieve this without data compression. Maximum theoretical throughput takes into account *format* and *specificationoverhead* with best case assumptions accurately. The number is primarily used as a *rough calculated value*, such as, for determining *bound* on possible performance early phase of a system design.

4.3.2 Peak measured throughput

Peak measured throughput is a throughput measured by a *real, implemented system* or *simulated system*. The value is the throughput measured over a short period of time; mathematically, this is the limit taken with respect to throughput as time approaches zero. This term is synonymouswith *instantaneous throughput*. This number is useful for systems that rely on burst data transmitted; however, for systems with a *high duty cycle*, it is less likely to be a useful measure of system performance.

4.3.3 Maximum sustained throughput

This value is the throughput averaged or integrated over a long period of time (sometimes, considered even infinity). For *high duty cycle networks*, it is likely to be the most accurate indicator of system performance. The maximum throughput is defined as the *asymptotic throughput* when the load (the amount of incoming data) is very large. In packet switched systems, where load and throughput always are equal (packet loss does not occur), the maximum throughput may be defined as the maximum load in bit/s that causes delivery time (the latency) tobecome unstable and increasing towards infinity. This value can also be used inappropriately in relation to Peak measured throughput to conceal packet shaping (see Wikipedia (2013) and the references there, for some details).

Sometimes, throughput is*normalized* and measured in percentage.

In a network system, channel efficiency, also known as *bandwidth utilization efficiency*, measured in percentage is the achieved throughput related to the *net bit-rate* in bit/s of a digital communication channel. For example, if the throughput is 70Mbit/s in a 100 Mbit/s Ethernet connection, the channel efficiency is 70 percent (%). In this example, effective 70Mbits of data are transmitted every second.

Channel utilization is a term related to the use of the channel, disregarding the throughput. It counts not only the data bits but also the overhead that makes use of the channel. In this, the channel is assumed to be noiseless. Otherwise, the throughput would not be only associated to the (efficient) nature of the protocolbut also to the retransmissions resultant from quality of the channel. In a simplistic approach, channel efficiency can be taken as equal to channel utilization assuming that the acknowledged packets are of zero lengthand the communication provider will not find any bandwidth relative to retransmissions or headers.

In a network system, the issues pertaining to *utilization load* versus *network congestion* have been drawing attention since its inception. Collision increases as the network gets on loaded. Introducing retransmissions and resulting increase in load are found toinduce even more collision (Cannon and Caudley (2007), for details). A typical device in the event of having slow traffic resulting from the network overload is to introduce switches. However, it has been found inefficient, apart from being ad hoc.

In this research work, we present a mathematical approach to enhance the performance of a switched network.

Usually, if a network reaches its physical limitations as a result of congestion or traffic, *segmentation* is being used to enhance its performance. However, despite its frequent use, it is known that the segmentation can only approach its upper bound. In this chapter, we describe the limitations of the segmentation method and show that *compatibility* relation based segmentation procedure would

enhance the performance of a network. Essentially, it aims at decentralizing the network leading to its decongestion.

4.4 Concept of Switching

A packet is one unit of binary data capable of being routed through a computer network. In order to improve upon communication performance and reliability, each message sent between two devices is often subdivided into packets by the underlying hardware and software.

A network switch is a small hardware device that joins multiple computers together within one local area network (LAN). Technically, network switches operate at layer two (Data link layer) of the OSI model.

Switches look at each packet for *conformity* and *correctness* and process it accordingly rather than merely repeating the *signals* to all *ports*. Switches assign Ethernet addresses of nodes residing on each part of the network and then allow only necessary traffic to pass through. Switches hinder *misaligned packets* from spreading by not forwarding them.

When a packet is received by a switch, it inspects the *destination* and *source hardware addresses* and compares them to a table of network segments and addresses. If the segments are the same, the packet is dropped; if the segments are different, then the packet is forwarded to the proper segment. In Cannon and Caudley (2007), it was observed that *dropping of packets and regeneration of forwarded packets enable switching technology to split a network into separate collision domains*. In a switched network, each segment is an independent collision domain.

4.4.1 Switches in a Network

In order to properly describe the topology of a network, it needs to be emphasized that *introducing switches* in a congested network serves the purpose different from that of *increasing the number of*

switched ports already installed. A switch installed in a location where it forwards most of the traffic it receives, will be less helpful than the one that filters them.

In particular, the goal of *introducing switches* in a network is to *minimize* its congestion, if any. A network which is not congested needs no introduction of switches (Cannon and Caudley2007).

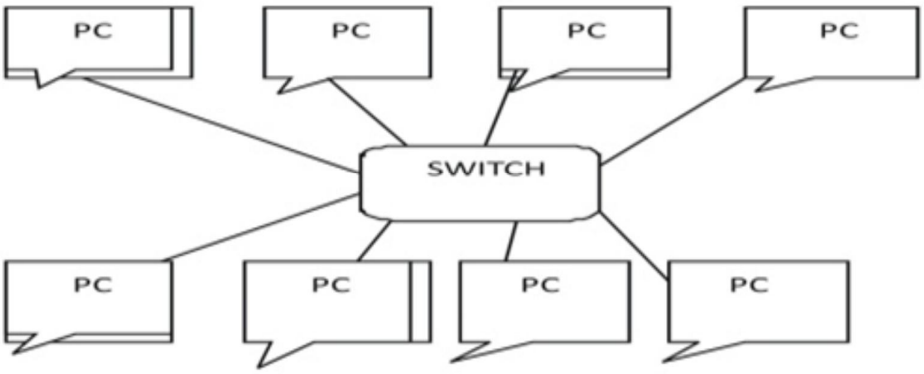


Fig. 4.3: Switches in a network.

4.4.2 Network Congestion

In a network system, issues pertaining to *network congestion* are encountered due to heavy traffic on the network or increase in the number of users sharing it. This can reasonably slow down the performance of the network (Kevin, 1981).

More often than not, more data are required to be added to a shared network. As a result, performance deteriorates due to competition for *Ethernet bus* by users (competitors) of the shared network (Wikipedia, 2012).

Further, nodes on the Ethernet network send packets when they determine that the network is not in use. It is possible that two nodes at different locations could try to send data at the same time, which might result into a collision. In the event of such collisions, both the packets are retransmitted,

adding to the traffic problem. The likelihood of such collision increases as more nodes and/ or traffic are added to the shared collision domain.

4.4.3 Network Segmentation

Network segmentation is an act of splitting a computer network into *subnets* comprehended as *network segments* or *network layers*.In the event of problems occurring with too many nodes in the same collision domain, suitable network segmentation is introduced to reduce collision, retransmission and contention for bandwidth (Cannon and Caudley, 2007). The following diagram is a simple illustration:

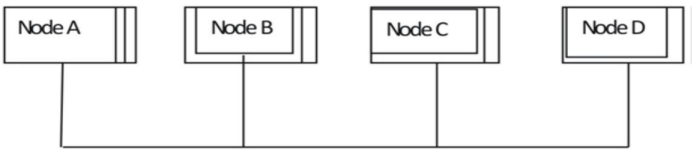


Fig. 4.4: Unsegmented Network

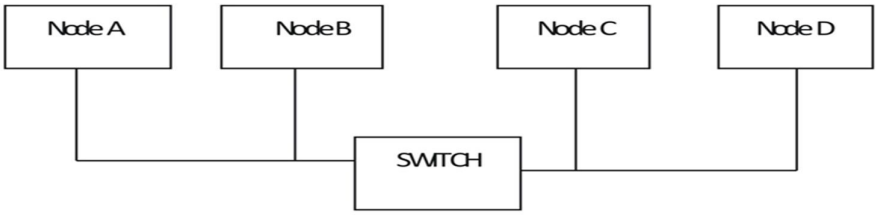


Fig. 4.5: Segmented Network

4.4.4 Limitation of Network Segmentation Method (NSM)

Usually, in a segmented network, most of the *communication activities* are performed by the subnets. At times, the sub-network gets flooded with messages reporting *status* and *throughput statistics* to the *Network Monitoring Centre (NMC) programs*, and as users' traffic increases, larger network

capacity is consumed. Not very frequently, lack of provision for coordinating and analyzing performance measures at terminals in the sub- network as well as at the host tell upon the efficiency of the NSM.

4.4.5 Application of Compatibility Relation to Switched Network

A segmentation of a congested network consists in its decomposition into smaller segments which give rise to a sort of partition. Essentially, a physically separate path (not intersecting with another path) for each pair of communicating ends needs to be built, which tells upon efficiency as well as costing. We propose that the notion of *covering* instead of *partitioning* a congested network may be applied to obtain a more compact and cost effective network. Thus, in order to achieve a competing segmentation of a congested network, a suitable *compatibility relation,* instead of an *equivalence relation*, needs to be defined. This, in turn, gives rise to a decomposition of the network into (maximal) compatibility blocks (segments) whose union is its minimal covering.

Further, compatibility relation can be applied to switched networks in the area of performance boosting (Cannon and Caudley, 2007). In order to adequately address performance problems in switched networks, the following additional information is in order. If the network demand per node is known, compatibility relation defined on the network would group *users with nodesthey communicate with most frequently on the same segment* in one compatibility block. And, then, user switches could be placed between these segments in an *iterative* process while blocking other types of traffic (incompatibles) until all nodes (devices) are seen to be in optimal performance level.

4.4.6 Compatibility Relation based Segmentation Procedure

Let R be a compatibility relation defined on a given network. Let us assume that the Network Segmentation Manager (NSM) is enabled and the port profile for network segmentation exists. If the network demand per node is known, proceed as follows:

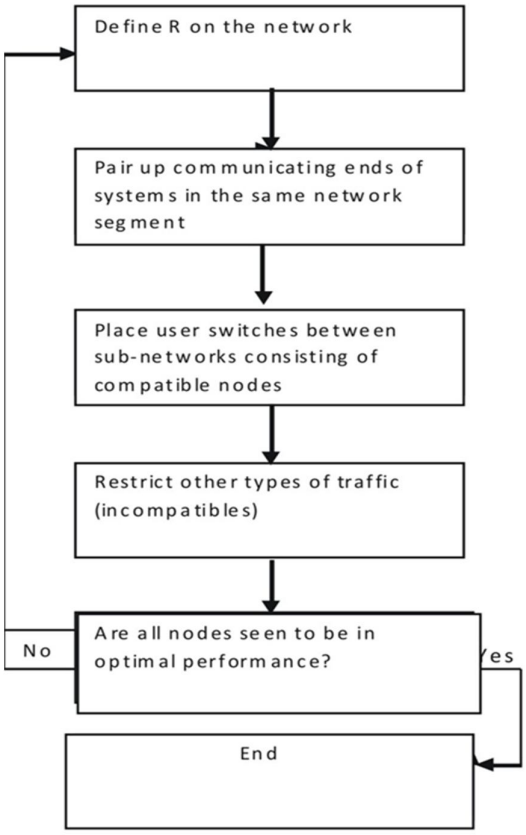


Fig. 4.6: Compatibility relation based segmentation procedure

This process continues until each sub-network attains its optimum in the sense of underlying compatibility relation.

Further, this procedure checkmates the congestion level in the network and thereby, decentralizes the network. It is observed that this procedure does not disconnect any segment or sub-network from the entire network, since the pairwise intersection of these (maximal's) subnets is non- empty.

Essentially, a compatibility relation based segmentation procedure segments a congested network into subnets (maximal compatibility classes). This, in turn, structures the network in such a way that no node of any given subnet communicates to every node of another subnet. As a result, the communicating network is reasonably minimized without adversely affecting its goal. . In fact, when

the network is segmented, there is a tradeoff between time and quality data. That is, if a particular node x_i in a subnet A does not relate to a node x_j in subnet B, but needs information from node x_j, then time is traded off to obtain that information from x_k, whenever x_i relates to node x_k in B. This is because the rate of packet delivery from x_j (source) to x_i (destination) is equal to the rate of packet delivery from x_k to x_i plus some time $t > 0$. However, there is a compensation for this delay. The compensation being that the maximal compatibility class (maximal subnet) through this node x_j (source) lowers its throughput to meet that of a (possible) congested path. This type of congestion control is similar to *rate- based technique.*

Further, it is observed that x_k above is a link from subnet A to subnet B. Therefore, such x_k in any subnet supports high bandwidth and low latency (the amount of time taken for a packet to travel from source to destination) between subnets A and B, since x_k can be reached.

In the following, we describe schematically how compatibility relation based network congestion control procedure does work.

Basically, traffic control can, among other things, decide if packets are queued or dropped (e.g., if the queue has reached some length limit, or if the traffic exceed some rate limit). Besides, it can decide in which order packets are to be delivered. For example, it can delay the sending of some packets to limit the rate of outbound traffic (Almesberger, 1999).

We may recall that when a compatibility relation is defined on a network system N, it decomposes N into (possibly) non-disjoint subnets, which are viewed as compatibility classes. Once the decomposition procedure is completed, each node in the network is associated with a compatibility class (say, C_k, $k = 1, 2, ...$). Also, in order to have a minimal covering of N, compatibility classes are obtained which represent the subnets of the refined form of N. A schematic representation is as follows:

Label x_k as L_k if for some compatibility classes C_i and C_j, we have $C_i \cap C_j = \{x_k\}$, for some i, j, k.

These L_k's in N form its backbone.

Now, the packets are routed through the network by means of L_k's. It is important to observe that the performance measure of a refined network will be a function of the cardinality of the set of L_k's.

Further, there exists at least one node in C_i which maintains a separate queue for any other subnet C_j, and a separate queue for nodes in itself.

Denote the first queue as q_1 and the second queue as q_2. Then any node within a compatibility class (subnet) C_i maintains a separate q_2 for each L_k in compatibility class C_i and maintains q_1 for nodes in any other compatibility class C_j. Also, it follows that each $L_i \in N$ maintains q_2 for some L_j (i≠j) in N. Therefore, communication flow can be re-channelized and assigned to the appropriate L_k's with the information about the network traffic and topology. In fact, accounting for the instantaneous queue length between L_k's, controller allocates packets to the least congested link.

The following diagram illustrates the concept described above.

Let a typical portion of the refined (based on a compatibility relation R) network be represented by fig. 7 and 8 below.

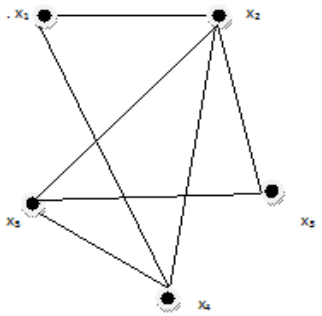


Fig. 4.7: compatibility graph

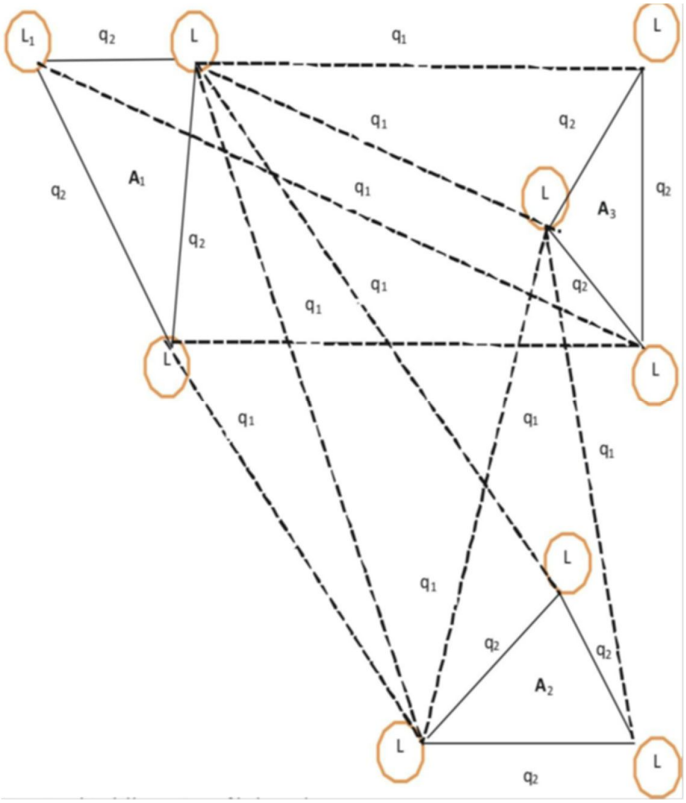


Fig. 4.8: diagram representing links and queues

Accordingly, the diagram (fig. 8) represents the links and queues.

It is conclusive to say that applying the concept of maximal compatibility classes to switched network proves helpful for obtaining the smallest possible set ofcompatible nodes for switch placement. Essentially, application of maximal compatibility classes helps minimizing the number of devices between the source and destination points of each packet. The underlying logic is that a switch installed in a location where it forwards all the traffic (within the maximal compatibility class) it receives will be less helpful than the one that filters most of the traffic.

From operation point of view, a suitable compatibility relation for a particular switched network needs to be directed to the administrative goal. For instance, if the administrative goal is to minimize logical network connectivity of sensitive servers and/ or applications to only those hosts and ports/

protocols that are essential (i.e., white-listing), correspondingly, a suitable compatibility relation needs to be defined.

Finally, we show below that in course of executing segmentation of a congested switched network, applying compatibility relation gives a *finer* as well as competitively reliable network topology.

Let the following diagram (fig. 9) be a typical portion of a star network topology, and fig. 10 a portion of a mesh network topology.



Fig. 4.9: A portion of a typical star topology.



Fig. 4.10: Mesh Network topology derived from Star Network topology.

In order to define a suitable compatibility relation on a network which transforms a star topology into a mesh topology, it is observed that only one node (say) relates to every other node of the network and no node of the network relates to any other node distinct from .

For example, consider the following set of words

, where scrabble, etc., respectively. Define on S by:

The following is a simplified graph for :

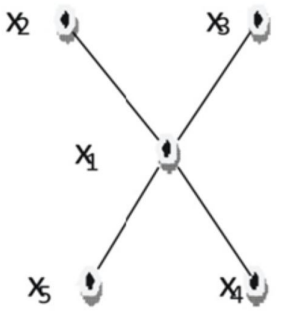


Fig. 4.11: Graph of .

A suitable compatibility relation which transforms the star topology (fig.4.11) into mesh topology (fig. 12) may thus be defined:

As a result, the following is a simplified graph for defined on S.

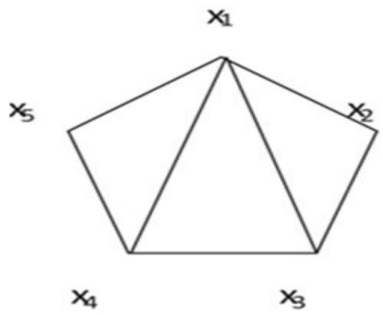


Fig. 4.12: Graph of

4.5 Network decentralizations

In what follows, we give an application of compatibility relation in *network decentralization*.

4.5.1 Some basic definitions

It is known that the technique of *network decentralization* is concerned with distribution of control to a class of nodes of the network. It makes sense to ask which node(s) should be given the control. The answer, of course, depends on the administrative goal for the design of the network. In this subsection, we describe limitations of a decentralized network and attempt to show that the notion of *maximal compatibles* could be gainfully applied for assigning controls to nodes. In other words, it aims at determining the smallest possible set of compatible nodes to which controls could be assigned.

Usually, a computer network connects nodes (*computing devices*) and supports communication between these nodes (devices). The network uses different schemes to create these connections. The scheme used is a function of the size of the network, services offered, etc. A *decentralized* network is such a scheme *commonly* used to *create* and *support* a network.

In a d*ecentralized* network system, issues pertaining to distribution of *control* versus optimum *communication flow* have been drawing attention since long: *manageability* becomes constrained as the system grows large (Jacob and Bullen, 1978).

This research work attempts to show that application of *maximal compatibles* would be helpful in assigning controls in network decentralization.

4.5.2 The Structure of a decentralized Network

Network decentralization is an act of distributing control to many devices of the network. In other words, it is a scheme that allocates resources; both hardware and software, to each individual work

station. The main goal of a decentralized network is to create multiple locations that support different *operating systems* to provide safeguards against complete *neutralization* by any form of attack.

In order to properly describe the topology of a decentralized network, it needs to be emphasized that the main goal of a decentralized network is to connect users and *resources* in a *transparent*, *open* and *scalable* fashion. However, the choice of a decentralized network depends on the need of the application.

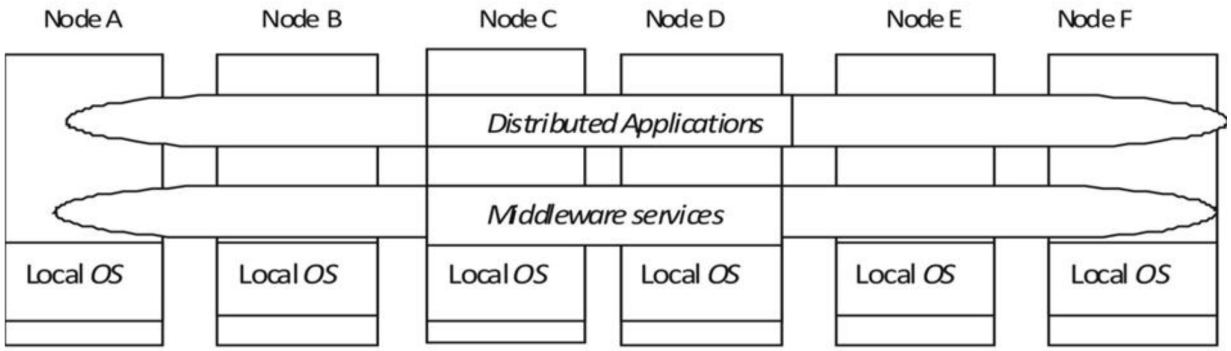


Fig. 4.13: Decentralized network organized as *middleware*.

Note that the *middleware* layer extends over multiple nodes.

We may recall that a *middleware* is computer software that provides services to software applications beyond those available from the Operating System. In distributed applications, middleware enhances communication and management of data.

4.5.3 Limitation of a decentralized network

Usually, in a decentralized network, no node has complete information about the network system state. Essentially, a node makes decision based only on local information it possess. This, in turn, makes diagnosing problems somewhat difficult in the event whereby troubleshooting analysis requires inspecting communication between nodes. Moreso, the *far- flung* distribution of control tends to make decentralized network difficult to manage.

It is possible that two nodes from different ends on the network could try to send data to one another. If no direct link exists between these nodes, the time taken to send one byte of data will depend on the distance between source and destination nodes. Therefore, in a decentralized network, *wire delay* is a bottleneck. See (Wikipedia, 2013), for some disadvantages of decentralized network.

4.5.4 Application of the notion of maximal compatibles in networkdecentralization

As described above, in a decentralized network, different segments may control and maintain their various *activities*. It consists of configured systems such that multiple nodes (servers) in the network provide the services used by other nodes (clients) of the network. A typical example of a decentralized network is one in which nodes (clients) utilize different nodes (servers) to carry out their various *functions* (Lau *et al.* (1972) provides many examples of decentralized network).

Schematically, the graph of a compatibility relation R of fig. 7 may be viewed as a decentralized network if, each node is a client and a server as well. It is observed that the symmetric property of a compatibility relation guarantees each node in the graph of R to assume *client – server functions*. Since the topology of R, viewed as a decentralized network, is identical to a mesh topology, there is reasonably enhanced *reliability* and *speed* in a decentralized network. Also, it is observed that this advantage grows significantly when the decentralized network is configured with *redundant* servers (nodes) - multiple nodes providing the same service.

Accordingly, we propose that applying the notion of maximal compatibility classes would help to determine the smallest possible set of compatible nodes to assign control in network decentralization. In order to properly demonstrate the application of maximal compatibility classes in a decentralized network, it needs to be emphasized that in the context of dual- mode decentralized scheme for controlling the flow of *messages* through the network, such as the ones discussed in (Lau *et al.*, 1972), each member of the maximal compatibility class in the graph of R, a given compatibility

relation, may be viewed as a local controller and, therefore, exchange some *simple* messages with their *neighbors* and determine the local optimum flows.

In addition, in order to achieve a competing decentralized network, the following steps are to be executed:

(i) A suitable compatibility relation is to be defined on the network (say N) and all its maximal compatibility classes (subnets) identified.

(ii) Assign *control* to each maximal compatibility class (MCC). Note that each MCC is an autonomous segment (by definition of MCC).

It may happen that some compatibility classes intersect with some MCCs. For example, in chapter 3,$\{x_2,\ x_3,\ x_4\} \subset \{x_2,\ x_3,\ x_4,\ x_5\}$ which is an MCC. However, since any compatibility class of a finite set is either itself a maximal compatibility class or a subclass of a maximal compatibility class. Therefore, (ii) above is sufficient.

It is observed that by applying this technique, the distribution of control to nodes would be reasonably minimized without adversely affecting the goal of network decentralization. In effect, it helps in distributing control to nodes in compatibility blocks whose union is a (minimal) covering of the network.

4.5.5 Some Characteristics of Compatibility based Decentralized Network

As discussed above, introduction of compatibility relation reasonably facilitates the distribution of control to nodes without adversely affecting the goal of decentralizing a network. We identify some characteristics of a compatibility- based decentralized network. The intent is to develop a broad framework for constructing network design that can be applied to some specific domains.

In what follows, considering some typical criteria used for performance, a compatibility- based decentralized network can be seen to perform better in the following respects:

i. Manageability: It becomes easier to control various functions (updating, repairing, logging etc.) of a system.

ii. Fault tolerance: The ability of the system to handle failure gets enhanced. Failure of some particular node(s) does not drastically hinder the functioning of the rest of the network.

iii. Extendibility: Adding new resource to enlarge the system becomes transparent as any node can be introduced into the network to share files with other nodes.

iv. Scalability: How large could the system grow? In reality, this characteristic is quite complex. In practice, any algorithm required to keep a decentralized network coherent is bound to carry a lot of overheads. Consequent to the cumulative growth of such overheads, the network may not scale.

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATION

5.1 Conclusion

In this research work, we have studied the concept of compatibility relation and thereby, that of maximal compatibility classes. In particular, we have pinned down some characteristic properties of compatibility classes. Moreso, we have studied some algorithms for computing maximal compatibility classes described in Paull and Unger (1959), Biswas (1989), etc. and have proposed one such algorithm.

We have presented an application of maximal compatibles to network segmentation and decentralization, in the event of network congestion and assignment of control to nodes, respectively.

Besides presenting an application of maximal compatibles to network segmentation and decentralization, this research work has described some limitations of NSM and decentralized network, and shown that application of compatibility relation can be gainfully made to overcome some of these limitations. In particular, it is emphasized that the concept of maximal compatibility classes is helpful for obtaining the smallest class of compatible nodes to assign control in network decentralization.

5.2 Recommendations

In view of the fact that applications of compatibility relation and thereby, that of the notion of maximal compatibility classes, have been found useful in different fields (Grasselli, 1962), it has increasingly become pertinent to closely investigate issues pertaining to maximal compatibility classes. In fact, the observation that the problem of deriving maximal compatibility classes for a given compatible relation defined on a finite set is similar to the graph theory problem of computing

maximal *complete subgraphs* in a symmetric graph has unfolded a perspective of multi- faceted application of compatibility relation.

Concluding, it can be emphasized that in a foreseeable future, besides in *hard sciences* like Engineering, Computer sciences, etc., in order to model problems in *soft sciences* like biology, economics, sociology, etc., which are invariably supposed to deal with a large number of incompletely specified interactions, applications of compatibility relation would play an important role.

In the light of this, a further deeper study of the mathematics of compatibility relations could be undertaken in readiness to handling any problem that may arise from the use of the current knowledge.

References

Almesberger, W. (1999), Linux Network Traffic Control- Implementation Overview, *In Proc. Of 5th Ann. Linux Expo, Raleigh,* pp. 153-164.

Baccelli, B.F and Muhlethaler, P. (2004), An Aloha Protocol for Multihog mobile Wireless Network. *In Proc.Of 16th ITC Specialist Seminar*. Pp. 6-13.

Biswas, N. N. (1989), Maximum Compatible Classes from Compatibility Matrices, *Sadhana*, vol.14, 213 – 218.

Caldwell S.H. (1958), Switching Circuits and Logic Design, John Wiley, pp.2-14.

Cannon K,and Caudley, K.(2007), CCNA Guide to Cisco Networking, third edition, vol.4, 1-10.

Charkraborty, A. (1987), On State Minimization of Very Large Incompletely Specified Sequential Machines, BE Project Report, Dept. of Electrical Communication Eng. Indian Inst. of Sci.

Copland, J.A, Abler, R and Bernhardt, K.L. (1999), IP Flow Identification for IP Traffic Carried over Switched Networks, *Int. J. of Computer and Telecomm. Networking*, vol. 31, pp.493-504.

Filep, L. (1998), Study of Fuzzy algebras and Relations from a general viewpoint, *Acta Mathematica Academiae Paedagogicae Nyrlregyhaziensis, Tonus* vol. 14, pp.49-55.

Grasselli, A. (1966), A Note on the Derivation of Maximal Compatibility Classes, *Calcalo*, vol. 3, 165- 176.

Grasselli A and Luccio, F. (1965), A Method for Minimizing the number of internal states in Incompletely Specified Sequential Networks, *IEEE Trans. on Electronic Computers*, vol.14, 350- 359.

Grasselli, A.(1962), Control units for Sequencing Complex Asynchronous Operations, *IRE Trans. on Electronic Computers*, vol.11, 483- 493.

Hill F.J and Peterson, G.R. (1968), Introduction to Switching Theory and Logic Design, John Wiley.

Holt, R.C. (1998), Structural Manipulations of Software Architecture Using Tarski Relational Algebra*, IEEE Trans., WCRE,* PP. 1-14.

Jacob, A. and Bullen,C. (1978), Centralization versus Decentralization of Information Systems: A critical survey and an annotated bibliography, Centre for Information Systems Research, vol.11, 112-114.

Karki, T. (2003), Compatibility Relations on Codes and Free Monoids, TUCS Tech. Rep. 772, Turku Centre for Computer Science, pp. 1-22.

Kelvin, L.M. (1981), Performance Measurement Problems in a Packet- Switch Network, *System Development Corporation*, vol.5, 29-40.

Kim, H. and Hou, J.C. (2004), A Fast Simulation Framework for IEEE 802.11- Operated WLANs, *In Proc. of ACM SIGMETRICS*, pp. 21-36.

Kohavi, Z. (1978), Switching and Finite Automata Theory, 2nd edition, McGraw- Hill.

Kramosil, I. (1995), Believeability Functions Induced by Partial Generalized Compatibility Relations, *Tara Mount*, vol.6, 103- 116.

Kurepa, D.J. (1952- 1953), On Reflexive Symmetric Relations and Graphs, RazpraveDissertationes, SlovenskaAkademija in Umetosti, pp.1- 28.

Lau, R. Persiano, R. and Varaiya, P. (1972), Decentralized information and control: A network flow example, *Automatic Control, IEEE Transactions,* vol. 17, 466- 473.

Lipovaski, S. and Marcus, M.P. (1994), Tolerance Rough Sets, Cech Topologies, *Learning process, Bull. Polish Academy of Science, Technical Science*, vol.42, 3, 471 – 487.

McCluskey, E.J. (1962), Minimum- State Sequential Circuit for Restricted Class of Incompletely Specified Flow Tables, *Int. Symposium on Theory of Switching Systems and Finite Automata*, vol.10, 1759- 1767.

Paull, M.C. and Unger, S.H. (1959), Minimizing the number of states in Incompletely Specified Sequential Switching functions, *IRE Trans. on Electronic Computers,*vol.8, 356- 367.

Pognowski, J. (1981), Tolerance Space with Applications in Linguistics. Pozhan Univ. Press, Pozhan.

Rosen, H.K. (1999), Discrete Mathematics and its Applications, 2nd Edition, McGraw Hill Press, New York.

Singh, D. Ibrahim, A.M. and Singh, J.N. (2010), A Note on Symmetrized Max- Plus Algebra, *Math. Sci. Math. Edu*., vol.5, 1 – 5.

Singh, D. Dunari, M.H. and Singh, J.N. (2010), A Note on Covering and Partition of a Finite Set, *Math. Sci. Math. Edu*.Vol.5 (2), 1- 5.

Singh, D. and William-west, T.O. (2012), A Note on Application of Compatibility Relation to Switched Network, *Microwave Int. Journal of Sci. and Tech*. vol.3(1), 74- 81.

Tremblay, J.P. and Manohar, R. (1975), Discrete Mathematical Structures with Applications to Computer Science, McGraw- Hill.

Weissman, J. (1962), Boolean Algebra, Map Coloring, and Interconnections, *American math. Monthly*, vol. 69, 608-614.

Wikipedia: user contributed 11(2012).

William C.W.M. (1962), On the Minimal Set of Compatibles for Closure, Microwave research Institute, pp.1- 46

Yen, J. (1990), Generalizing the Dempster – Shafer Theory to Fuzzy Set, IEEE Trans., Man and Cybernetics, vol. 20(3), pp. 559-570

Zhao, J.K, Rudnick, E. and Patel, F. (1997), Static Logic Implication with Application to Redundancy Identification*, In Proc. IEEE VLSI Test Symposium*, pp. 288-293.