A Note on Vector Representation of Compatibility Classes

D. Singh, Ph.D. †
T.O. William-west ‡

Abstract

This paper presents the notion of vector representation of compatibility classes of a finite set S upon which a compatibility relation is defined. Some characteristic properties of these vectors are presented. Moreover, using this notion, a procedure for computing all the maximal compatibility classes of S is described and some new results obtained.
Keywords: Compatibility relation, vectors, maximal compatibility classes.

1. Introduction

A relation on a finite set which is reflexive and symmetric is called a *compatibility* relation. Usually, a compatibility relation is denoted $\approx$. If $x, y \in S$, we say x is compatible to y if $x \approx y$. Essentially, a compatibility relation defined on a set decomposes the set into its possibly non- disjoint subsets [3], henceforth called *compatibility classes*. It follows that the elements of a compatibility class are pairwise compatible. Some properties of compatibility classes have been investigated in [4].

A subclass $M \subseteq S$ is called a *maximal compatibility class* if any element of M is compatible to its every other element and no element of $S - M$ is compatible to every element of M. Graphically, a maximal compatibility class (MCC) is a subgraph of S which is not a proper subgraph of any other subgraph of S. Some results and properties of MCCs can be found in [4].

The notion of representing *equivalence* classes with the help of two vectors called FIRST and MEMBER has been investigated in [6] and a similar approach recently in [5]. In [4], Singh and William-west introduced the notion of representing MCCs with the help of the FIRST and MEMBER vectors. However, in this note,weI modify such technique to all compatibility classes of a finite set S endowed with a compatibility relation and, thereby, present a procedure for computing all MCCs of S from these vectors.

2. Example of Compatibility Relation

Let $S = \{x_1, x_2, \dots, x_5\} = \{\{1,2\}, \{2,3\}, \{3,4\}, \{2,3,4\}, \{1,4,5\}\}$, where $x_1 = \{1,2\}, x_2 = \{2,3\}$, etc. Define a compatibility relation R by:

$$R = \{(x_i, x_j) : x_i, x_j \in S \bigwedge x_i \approx x_j \text{ if } x_i \text{ and } x_j \text{ contain some common element}\}$$

Now, we have the following compatibility classes

$\{x_1, x_2, x_4\}, \{x_2, x_3, x_4\}, \{x_3, x_4, x_5\}, \{x_1, x_5\}$. It is not difficult to observe that all these compatibility classes are maximal.

In any simplified graph of R, since R is symmetric, only one of x_iRx_j is drawn. By consequence, the graph is undirected. Also, since loops carry no information, it is insignificant to draw them. The curious researcher may see [4] for a simplified graph of R.

3. Vector Representation of Compatibility Classes

Abstracting from [6], a representation of compatibility classes with the aid of two vectors called FIRST and MEMBER is presented. The following steps are employed:

Step A

Starting with the compatibility class containing the least subscript number i. The ith component of the FIRST vector, for $1 \leq i \leq n$, contains the (subscript) number which is the first element in the compatibility classes to which i belongs. The ith component of the MEMBER vector contains the numbers which follow i in the compatibility classes to which i belongs, unless x_i (for $i = n$) is the last element, in which case MEMBER[i] is zero.

Step B

Proceed to the compatibility class(es) containing elements with subscript number $i + 1$ and repeat the procedure in step A to obtain the $(i + 1)th$ component of the FIRST and MEMBER vectors.

Step C

Continue the process recursively until the $(i + n - 1)th$ component, then stop.

The aforesaid steps could be illustrated with the example in section 2 as follows:

Subscript	FIRST	MEMBER
1	1	2, 4, 5
2	1, 2	3, 4
3	2, 3	4, 5
4	1, 2, 3	0, 5
5	1, 3	0

The FIRST and MEMBER vectors, corresponding to state x_i, are denoted by FIRST[i] and MEMBER[i] respectively.

Remark 1

For any vector representation (ie., FIRST and MEMBER vectors) of compatibility classes of a finite set S, a compatibility relation defined among all elements of S could be obtained which retrieves the set of all compatibility classes defined by the relation. This is obtained by considering the FIRST vectors. That is, if FIRST[i] $\subseteq FIRST$[j], then $x_i \approx x_j$. By a careful check through all the cells of the FIRST vectors in correspondence with the subscript number, all the compatibility classes may be obtained.

It is significant to note that whenever $x_i \approx x_j$, FIRST$[i]$ need not be equal to FIRST$[j]$, unless it satisfies the criteria of being so.

Theorem 1

Let S be a finite set upon which a compatibility relation is defined. Let $F_m = \{MEMBER[i] \mid i = 1, 2, \dots, n\}$ be a set of MEMBER vectors. Then S has at least two maximal compatibility classes if zero is a part of two of the MEMBER vectors in F_m.

Proof: Let S and F_m be as in the statement of theorem 1. If zero is a part of MEMBER$[j]$, then every element $x_j \in S$ belong to a compatibility class which is itself an MCC or a subclass of some MCC, M_j (say). Also, if zero is a part of MEMBER$[k]$, for $k \neq j$. Then there exist a compatibility class C_k containing x_k, which is not a subclass of M_j (otherwise MEMBER$[k]$ will not contain a zero). Since $C_k \not\subset M_j$, hence C_k must either be itself an MCC or a subclass of some other MCC distinct from M_j. Therefore, the result is immediate.

Note that it is not difficult to see that every compatibility class of a finite set is either itself an MCC or is a subclass of some MCC.

Theorem 2

Let S be a finite set upon which a (non- discrete) compatibility relation is defined. Then there does not exist any vector representation of the compatibility classes of S whose set of MEMBER vectors contain zero in each of its members.

Proof: Let S be as in the statement of theorem 2. Let $F_m = \{MEMBER[1], MEMBER[2], \dots, MEMBER[n]\}$ be the set of MEMBER vectors corresponding to $x_{i=1,2,\dots,n}$. Suppose for contradiction $0 \in MEMBER[i], \textit{for each } i$, then any compatibility class, C_k contains an element x_k which is its last element. This means that the compatibility classes $C_{k_1}, C_{k_2}, C_{k_3}, \dots, C_{k_n}$ have $x_{k_1}, x_{k_2}, x_{k_3}, \dots, x_{k_n}$ as their last elements respectively. Since the compatibility relation is non- discrete, for some $r, s \in N$, $x_{k_s} \approx x_{k_r}$, with $k_r < k_s$. This contradicts the fact that x_{k_r} is a last element of C_{k_r}. Hence the result holds.

Remark 2

If zero is a part of k −number of the MEMBER vectors in F_m, S may not have up to $(k + r)$ −number of MCCs, for $k \leq 2$ *and* $k + r \leq n, r \in N$.

Further, it is important to note that it is not always the case that the number of MCCs of S will be equal to the number of MEMBER vectors containing element of which 0 is one.

4. Characteristic Properties of FIRST and MEMBER Vectors

Let S be a finite set upon which a compatibility relation is defined. The following results hold.

i. For $i \neq j$, FIRST$[i] \neq$ FIRST$[j]$ and MEMBER$[i] \neq$ MEMBER$[j]$.

ii. For some i, there exists an element in $F_m = \{MEMBER[i] \mid i = 1, 2, \dots, n\}$ and $F_f = \{FIRST[i] \mid i = 1, 2, \dots, n\}$ with 0 and 1 as its parts, respectively.

iii. Corresponding to each element of S there is a unique vector which is a combination of FIRST and MEMBER vectors. In particular, $S = \bigcup_{i=1}^{n}\{x_i\} = \{(F_i, M_i)\}, 1 \leq i \leq n$, for $F_i \in FIRST[i]$ and $M_i \in MEMBER[i]$.

5. Procedure for Deriving MCCs from FIRST and MEMBER Vectors

In [1], [2] and [4], an algorithm for computing MCCs have been investigated. However, in what follows, we present a procedure for deriving MCCs from FIRST and MEMBER vectors.

Level 1
Pair up i, j, k if their FIRST vectors form a chain of subsets. That is, FIRST$[i] \subset$ FIRST$[j] \subset$ FIRST$[k]$, for all i, j, k.

Level 2
If FIRST$[i] \cap$ FIRST$[j] \neq \emptyset$, then pair up $i \; and \; j$.

Level 3
If $(i, j), (j, k), (i, k)$, then pair up (i, j, k).
Delete (i, j), if (i, j, k) and (i, j) belong to the list of pairs enumerated in level 1, 2 or 3.

Accordingly, the procedure will now be illustrated using the example presented above, as follows.

Level 1
(1, 2, 4), (1, 4, 5), (3, 4)

Level 2
(2, 3), (3, 5), (2, 5)

Level 3
(2, 3, 5). But (3, 4), (2, 4), (2, 3) $\Longrightarrow$ (2, 3, 4) $[by \; level \; 1 \,\&\, 2]$

Therefore, we obtain (1, 2, 4), *(1, 4, 5)*, (2, 3, 4) and (2, 3, 5).
The procedure presented in this paper may generate a pairwise incompatible classes and thereby, non- maximal compatibility class. Therefore, a second check for *maximality* is required to obtain all MCCs. Consequently, the pair *(1, 4, 5)* when spilt into compatibles, will produce (1, 5) and (1, 4). From level 3, we delete (1, 4). Hence, the MCCs are: (1, 2, 4), (1, 5), (2, 3, 4) and (2, 3, 5).

6. Concluding Remarks

In view of the fact that compatibility classes of a finite set endowed with a compatibility relation can be represented as vectors, it is promising to examine all those properties of vector spaces which may apply to the set. For example, the set of vectors of each set of compatibility classes of a finite set S, which forms *minimal covering* of S, forms a linear span of S.

† D. Singh, Ph.D. Ahmadu Bello University, Nigeria.
‡ T.O. William-west Ahmadu Bello University, Nigeria.

References

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

2. 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.

3. 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.

4. Singh, D. and William-west, T.O. (2013), Compatibility Relation to Network Segmentation and Decentralization, In Proc. of Int. Conf. on Info. Sci. and Tech. Application, Atlantic Press, 114- 119.

5. Sikander M. H, M. Z. Arshad, R. Khadija and K. Najia (2003), An Algorithm to Find Equivalence classes, Vol. 2(3), 240 - 249.

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