

Portland State University
PDXScholar

Systems Science Faculty Publications and
Presentations

Systems Science

2004

Modified Reconstructability Analysis for Many-Valued Functions and Relations

Anas Al-Rabadi
Portland State University

Martin Zwick
Portland State University, zwick@pdx.edu

Let us know how access to this document benefits you.

Follow this and additional works at: http://pdxscholar.library.pdx.edu/sysc_fac

 Part of the [Logic and Foundations Commons](#), and the [Systems Architecture Commons](#)

Citation Details

Al-Rabadi, A., and Zwick, M. (2004). "Modified Reconstructability Analysis for Many-Valued Functions and Relations." *Kybernetes*, vol. 33, No. 5/6, pp. 906-920

This Post-Print is brought to you for free and open access. It has been accepted for inclusion in Systems Science Faculty Publications and Presentations by an authorized administrator of PDXScholar. For more information, please contact pdxscholar@pdx.edu.

MODIFIED RECONSTRUCTABILITY ANALYSIS FOR MANY-VALUED FUNCTIONS AND RELATIONS

Anas N. Al-Rabadi⁽¹⁾, and Martin Zwick⁽²⁾

(1) ECE Department (2) Systems Science Ph.D. Program @Portland State University
[alrabadi@ece.pdx.edu]⁽¹⁾, [zwick@sysc.pdx.edu]⁽²⁾

KEYWORDS: Reconstructability Analysis, Decomposition, Relations, Logic Functions, Many-Valued Logic.

ABSTRACT

A novel many-valued decomposition within the framework of lossless Reconstructability Analysis is presented. In previous work, Modified Reconstructability Analysis (MRA) was applied to Boolean functions, where it was shown that most Boolean functions not decomposable using conventional Reconstructability Analysis (CRA) are decomposable using MRA. Also, it was previously shown that whenever decomposition exists in both MRA and CRA, MRA yields simpler or equal complexity decompositions. In this paper, MRA is extended to many-valued logic functions, and logic structures that correspond to such decomposition are developed. It is shown that many-valued MRA can decompose many-valued functions when CRA fails to do so. Since real-life data are often many-valued, this new decomposition can be useful for machine learning and data mining. Many-valued MRA can also be applied for the decomposition of relations.

1 INTRODUCTION

One general method to understand complex systems is to decompose the system in terms of less complex sub-systems (Klir 1985, Krippendorff 1986). Full decomposition, as opposed to partial decomposition, consists of the determination of the minimal sub-sets of relations that describe the system acceptably. The quality of the decomposition is evaluated by calculating (1) the amount of information (or, conversely, the loss of information, or error) which exists in the decomposed system, and (2) the complexity of the decomposed system. The objective is to decompose the complex system (data) into the least complex and most informative (least error) model.

This paper is organized as follows: section 2 presents a background on Reconstructability Analysis, both conventional (CRA) and modified (MRA). Many-valued MRA decomposition is presented in section 3. Conclusions and future work are included in section 4.

2 RECONSTRUCTABILITY ANALYSIS

Reconstructability Analysis (RA) is a technique developed in the systems community to decompose relations or distributions involving qualitative variables (Conant 1981, Klir 1985, Krippendorff 1986, Zwick 1999). We are here concerned with lossless decomposition of completely specified set-theoretic (crisp possibilistic) functions and relations. (We do not address information-theoretic, i.e. probabilistic, distributions). In lossless RA decomposition, the aim is to obtain the simplest model of the data which has zero error. The models representing possible decompositions define a graph-based lattice of structures. A “model” is a structure applied to some data (here a set-theoretic relation). Each model is a set of sub-relations projected from the original relation and represented by look-up tables.

New lossless RA-based decomposition, called Modified RA (MRA) decomposition, has been introduced in (Al-Rabadi 2001, Al-Rabadi et al 2002). While CRA decomposes using all values of the function, MRA decomposes using (1) the minimum set of values from which the function can be reconstructed without error, and (2) the simplest model (at the lowest level in the lattice of structures) for each value in the minimal set.

The first principle is illustrated for Boolean functions as follows: For every structure in the lattice of structures, decompose the Boolean function for one value only, e.g. for value of “1”, into the simplest error-free decomposed structure. One thus obtains the 1-MRA decomposition. This model consists of a set of projections which when intersected yield the original Boolean function. This is illustrated in the following example.

Example 1. For Boolean function:
 $F = x_1x_2 + x_1x_3$

Figure 1 illustrates the simplest models obtained using CRA and MRA decompositions. While CRA decomposes for both “0” and “1” values of the Boolean function, MRA decomposes only for value “1”, since $F(x_1,x_2,x_3)$ can be completely retrieved if one knows the (x_1,x_2,x_3) values for which $F=1$.

For *Boolean functions* there are two advantages of MRA over CRA: (1) MRA decomposition is simpler than CRA decomposition, so the MRA algorithm needs less time and space for its computation, and (2) MRA directly implements the intersection operation with an AND gate in binary logic; consequently MRA decomposition leads directly to a binary circuit and thus can be applied to both machine learning and binary circuit design. On the other hand, the intersection operation in CRA requires ternary logic to accommodate ‘don’t cares’ which are represented in top middle of Figure 1 by ‘-’. Therefore, CRA has no simple application in binary circuit design.

3 MANY-VALUED MODIFIED RECONSTRUCTABILITY ANALYSIS

This section presents MRA for many-valued functions and relations.

3.1 GENERAL APPROACH

Real-life data are in general many-valued. Consequently, if MRA can decompose relations between many-valued variables it can have practical applications in machine learning and data mining. Many-valued MRA is made up of two main steps which are common to two equivalent (intersection-based and union-based) algorithms: (1) partition the many-valued truth table into sub-tables, each contain only single functional value, and (2) Perform CRA on all sub-tables. Figure 2 illustrates the general pre-processing procedure for the two many-valued MRA algorithms, which will be explained in more detail below.

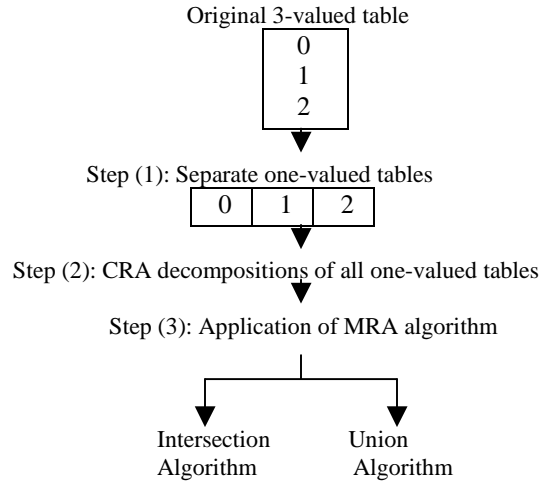


Figure 2. Steps for many-valued MRA.

Original Function				Simplest CRA Model						Simplest CRA Circuit Model				
x_1	x_2	x_3	F	α		β		γ						
				x_1	x_2	f_1	x_1	x_3	f_2	x_2	x_3	f_3		
0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	1	0	0	1	0	0	1	0	0	1	-		
0	1	0	0	1	0	-	1	0	-	1	0	-		
1	0	0	0	1	1	1	1	1	1	1	1	-		
1	0	1	1											
1	1	0	1											
1	1	1	1											
1-Modified Function				Simplest 1-MRA Model						Simplest 1-MRA Circuit Model				
x_1	x_2	x_3	F1	δ'		γ'								
				x_1	f_2'	x_2	x_3	f_3'						
1	0	1	1	0	0	0	0	0						
1	1	0	1	1	1	0	1	1						
1	1	1	1			1	0	1						
						1	1	1						
				$x_1 : x_2 x_3$										

Figure 1. Conventional versus Modified RA decompositions for the Boolean function: $F = x_1x_2 + x_1x_3$.

For an “n”-valued completely specified function one needs (n-1) values to define the function. We thus do all n decompositions and use for our MRA model the (n-1) simplest of these. For example, using the lattice-of-structures, decompose the 3-valued function for each individual value. One then obtains the simplest lossless MRA decomposition for value “0” of the function (denoted as the 0-MRA decomposition), for value “1” (1-MRA decomposition), and for value “2” (2-MRA decomposition). By selecting the simplest two models from these 0-MRA, 1-MRA, and 2-MRA decompositions, one can generate the complete function.

In the intersection method, first the CRA decompositions are expanded to include the full set of variable and function values, and these “expanded” decompositions are then intersected to yield the original table.

Equivalently, one can use a union operation to generate the corresponding many-valued MRA as follows: (1) Decompose the original table (function or relation) into sub-tables for each output value: e.g., $T = T_0 \cup T_1 \cup T_2$ for the corresponding output values O_0 , O_1 , and O_2 respectively, (2) Do the 3-valued CRA decomposition on each sub-table. Let M_j be the decomposition of T_j , (3) The reconstructed function or relation (T^*) is the union of all the sub-table decompositions, $T^* = \bigcup_{j=0}^{n-1} M_j \otimes O_j$,

where \otimes is the set-theoretic Cartesian product. The union procedure can also be done with (n-1) decompositions.

3.2 COMPLETE EXAMPLES

Following are two examples which illustrate many-valued Modified Reconstructability Analysis of 3-valued functions. In the first example MRA can decompose the function for only two values, and one has no choice but to use both in the MRA model. In the second example, the function is decomposable for all three of its values, and the two simplest decompositions are chosen to define the model. In discussing the second example, we show that this approach is generalizable to set-theoretic relations, in addition to mappings.

Example 2. We will generate the MRA decomposition for the ternary function specified by the following ternary Marquand chart:

	X_3				
		$X_1 X_2$	0	1	2
00			0	0	0
01			1	1	0
02			1	1	1
10			0	0	2
11			0	0	2
12			1	1	1
20			0	2	0
21			1	1	0
22			2	2	0
					F

The following is the intersection algorithm for many-valued MRA for the ternary function in Example 2.

Step 1: decompose the ternary chart of the function into three separate tables each for a single function value. This will produce the following three sub-tables.

Value “0”	Value “1”	Value “2”
000	010	102
001	011	112
002	020	201
012	021	220
100	022	221
101	120	
110	121	
111	122	
200	210	
202	211	
212		
222		
D0	D1	D2

Step 2: Perform CRA for each sub-table.

Step 2a: The simplest error-free 0-MRA decomposition is the original “0”-subtable itself since it is not decomposable.

Step 2b: 1-MRA decomposition of D1 is as follows:

Table 1 Table 2

X_1	X_2	X_3
0	1	1
0	2	1
1	2	0
2	1	2
		2

D11 D12

Step 2c: The 2-MRA decomposition of D2 is as follows:

Table 3 Table 4

X_1	X_3	X_2
1	2	0
2	1	2
2	0	0
		2
		2

D21 D22

THE INTERSECTION ALGORITHM

Step 3.1: Select the (3-1=2) simplest error-free decomposed models. In this example these are 1-MRA and 2-MRA decompositions. MRA thus gives the decomposition model of D11:D12:D21:D22 from which the original function can be reconstructed as follows.

Step 3.2: Note that, for Tables 1 and 2, the MRA decomposition is for the value “1” of the logic function. Therefore, the existence of the tuples in the decomposed model implies that the function has value “1” for those tuples, and the non-existence of the tuples in the decomposed model implies that the function does not have value “1” but “0” or “2” for the non-appearing tuples. This is shown in Tables 1’ and 2’, respectively. Similarly note that, for Tables 3 and 4, the MRA decomposition is for the value “2” of the logic function. Therefore, the existence of the tuples in the decomposed model implies that the function has value “2” for those tuples, and the non-existence of the tuples in the decomposed model implies that the function does not have value “2” but “0” or “1” for the non-appearing tuples. This is shown in Tables 3’ and 4’, respectively.

Table 1’ Table 2’

X_1	X_2	F_1	X_2	X_3	F_2
0	0	0,2	0	0	0,2
0	1	1,0,2	0	1	0,2
0	2	1,0,2	0	2	0,2
1	0	0,2	1	0	1,0,2
1	1	0,2	1	1	1,0,2
1	2	1,0,2	1	2	0,2
2	0	0,2	2	0	1,0,2
2	1	1,0,2	2	1	1,0,2
2	2	0,2	2	2	1,0,2

Table 3’ Table 4’

X_1	X_3	F_3	X_2	X_3	F_4
0	0	0,1	0	0	0,1
0	1	0,1	0	1	2,0,1
0	2	0,1	0	2	2,0,1
1	0	0,1	1	0	0,1
1	1	0,1	1	1	0,1
1	2	2,0,1	1	2	2,0,1
2	0	2,0,1	2	0	2,0,1
2	1	2,0,1	2	1	2,0,1
2	2	0,1	2	2	0,1

In Tables 1’ and 2’ (i.e., the decomposition for value “1” of the function), the existence of value “1” (of sub-relations F_1 and F_2) means that the value “1” appeared in the original non-decomposed function for the corresponding tuples that appear in each table, but *does not imply* that the values “0” or “2” (of sub-relations F_1 and F_2) did not exist in the original non-decomposed function for the same tuples. Therefore “0” and “2” are added to “1” as allowed values. In the remaining tuples, however, only “0” and “2” are allowed since the value “1” did not occur. Similarly, in Tables 3’ and 4’, the existence of the value “2” (of sub-relations F_3 and F_4) means that the value “2” appeared in the original non-decomposed function for the corresponding tuples that appear in each table, but *does not imply* that values “0” or “1” did not exist in the original non-decomposed function for the same tuples. Therefore “0” and “1” are added to “2” as allowed values. In the remaining tuples, however, only “0” and “1” are allowed since the value “2” did not occur. Set-theoretically, obtaining tables 1’, 2’, 3’, and 4’ from tables 1, 2, 3, and 4 is described as follows:

$$\text{Table 1'}: (D11 \otimes (0,1,2)) \cup (D11' \otimes (0,2))$$

$$\text{Table 2'}: (D12 \otimes (0,1,2)) \cup (D12' \otimes (0,2))$$

$$\text{Table 3'}: (D21 \otimes (0,1,2)) \cup (D21' \otimes (0,1))$$

$$\text{Table 4'}: (D22 \otimes (0,1,2)) \cup (D22' \otimes (0,1))$$

where ‘ here means complement.

Step 3.3: Tables 1’, 2’, 3’, and 4’ are used to obtain the block diagram in Figure 3, where the following set-theoretic equations govern the outputs of the levels in the circuit shown in the figure:

$$F = F5 \cap F6$$

$$F5 = F1 \cap F2$$

$$F6 = F3 \cap F4$$

where F1 is given by Table 1', F2 by Table 2', F3 by Table 3', and F4 by Table 4', respectively.

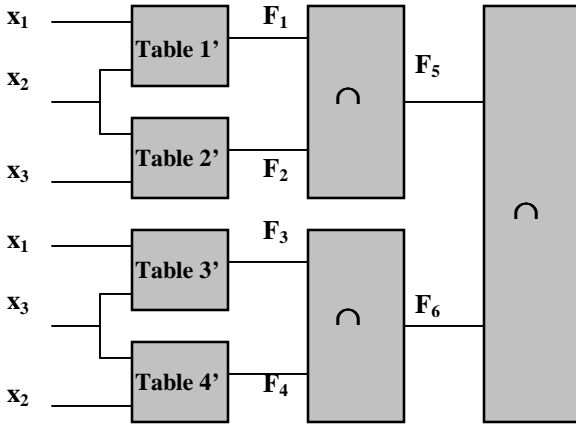


Figure 3. The decomposed structure resulting from the many-valued MRA decomposition.

The intermediate sub-functions, F5 and F6 are shown in the following maps, respectively.

x_2x_3	x_1	00	01	02	10	11	12	20	21	22
0	0	0,2	0,2	0,2	1	1	0,2	1	1	1
0	1	0,2	0,2	0,2	0,2	0,2	0,2	1	1	1
0	2	0,2	0,2	0,2	1	1	0,2	0,2	0,2	0,2

x_2x_3	x_1	00	01	02	10	11	12	20	21	22
0	0	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
0	1	0,1	0,1	2	0,1	0,1	2	0,1	0,1	0,1
0	2	0,1	2	0,1	0,1	0,1	0,1	2	2	0,1

$$F_6 = F_3 \cap F_4$$

Note that in Figure 3 the intersection blocks in the second level and the intersection block at the third (output) level, are general and do not depend on the function being decomposed. Only the tables at the first level depend upon this function.

THE UNION ALGORITHM

Steps 1 and 2 are the same as in the intersection algorithm.

Step 3.1: Using the decomposition model D11:D12:D21:D22 obtain D1 and D2 by standard methods as follows:

$$\begin{aligned} D1 &= (D11 \otimes x3) \cap (D12 \otimes x1) \\ D2 &= (D21 \otimes x2) \cap (D22 \otimes x1) \\ D0 &= (D1 \cup D2)' \end{aligned}$$

where D1 is the decomposition for function value "1", D2 for function value "2", and $x1, x2, x3 \in \{0,1,2\}$.

Step 3.2: Perform the set-theoretic operations to obtain the total function from the decomposed sub-functions.

$$\begin{aligned} x1x2x3F &= (D1 \otimes 1) \cup (D2 \otimes 2) \cup ((D1 \cup D2)' \otimes (1 \cup 2)') \\ &= (D1 \otimes 1) \cup (D2 \otimes 2) \cup ((D1 \cup D2)' \otimes 0) \end{aligned}$$

Alternatively, one can use all three decompositions:

$$x1x2x3F = (D0 \otimes 0) \cup (D1 \otimes 1) \cup (D2 \otimes 2)$$

The function value of $(x1, x2, x3)$ is determined by the block diagram of Figure 4, where G performs the following operation:

$$\begin{aligned} F &= 0 \text{ if } (x1x2x3) \in D0 \\ F &= 1 \text{ if } (x1x2x3) \in D1 \\ F &= 2 \text{ if } (x1x2x3) \in D2 \end{aligned}$$

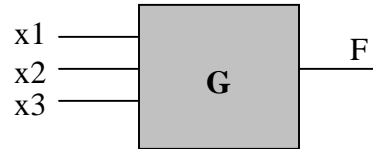


Figure 4. Block diagram for the union algorithm of MRA of Example 2.

Note that the logic function in Example 2 is non-decomposable using CRA. Consequently, as can be seen from this example and analogously to the binary case, the new many-valued MRA is superior to CRA.

We now consider an example where CRA does decompose, and also where MRA decomposes for all three values.

Example 3. Let us generate the MRA decomposition for the ternary Marquand chart by the following ternary Marquand chart:

	X_3			
$X_1 X_2$		0	1	2
00		0	0	0
01		1	1	1
02		1	1	1
10		0	0	2
11		0	0	2
12		1	1	1
20		0	2	0
21		1	1	1
22		2	2	0

F

Utilizing the intersection-based algorithm, one obtains the following results for MRA for the ternary function in Example 3.

Step 1: decompose the ternary chart of the function into three separate tables each for a single function value. This will produce the following three sub-tables.

Value "0"	Value "1"	Value "2"
000 001 002 100 101 110 111 200 202 222	010 011 012 020 021 022 120 121 122 210 211 212	102 112 201 220 221
D0	D1	D2

Step 2: Perform CRA for each sub-table.

Step 2a: The 0-MRA decomposition of D0 is as follows:

Table 1	Table 2	Table 3
$X_1 X_2$	$X_2 X_3$	$X_1 X_3$
00	00	00
10	01	01
11	02	02
20	10	10
22	11	11
	22	20
		22
D01	D02	D03

Step 2b: The 1-MRA decomposition of D1 is as follows:

Table 4	Table 5
$X_1 X_2$	X_3
0	1
0	2
1	2
2	1
D11	D12

Step 2c: The 2-MRA decomposition of D2 is as follows:

Table 6	Table 7
$X_1 X_3$	$X_2 X_3$
1	2
2	1
2	0
	2
	2
D21	D22

THE INTERSECTION ALGORITHM

Step 3.1: Select the two simplest decomposed models, namely the 1-MRA and 2-MRA decompositions. These are at a lower level in the lattice of structures than 0-MRA.

Step 3.2: Analogously to Example 2, one obtains the following expanded tables:

Table 4'	Table 5'	Table 6'	Table 7'
$X_1 X_2 F_1$	$X_3 F_2$	$X_1 X_3 F_3$	$X_2 X_3 F_4$
0 0 0,2	0 1,0,2	0 0 0,1	0 0 0,1
0 1 1,0,2	1 1,0,2	0 1 0,1	0 1 2,0,1
0 2 1,0,2	2 1,0,2	0 2 0,1	0 2 2,0,1
1 0 0,2		1 0 0,1	1 0 0,1
1 1 0,2		1 1 0,1	1 1 0,1
1 2 1,0,2		1 2 2,0,1	1 2 2,0,1
2 0 0,2		2 0 2,0,1	2 0 2,0,1
2 1 1,0,2		2 1 2,0,1	2 1 2,0,1
2 2 0,2		2 2 0,1	2 2 0,1

Set-theoretically, obtaining tables 4', 5', 6', and 7' from tables 4, 5, 6, and 7 is described as follows:

- Table 4': $(D11 \otimes (0,1,2)) \cup (D11' \otimes (0,2))$
- Table 5': $(D12 \otimes (0,1,2)) \cup (D12' \otimes (0,2))$
- Table 6': $(D21 \otimes (0,1,2)) \cup (D21' \otimes (0,1))$
- Table 7': $(D22 \otimes (0,1,2)) \cup (D22' \otimes (0,1))$

Step 3.3: Tables 4', 5', 6', and 7' are used to obtain the block diagram in Figure 5, where the following set-theoretic equations govern the outputs of the levels in the circuit shown in the figure:

$$\begin{aligned} F &= F_5 \cap F_6 \\ F_5 &= F_1 \cap F_2 \\ F_6 &= F_3 \cap F_4 \end{aligned}$$

where F_1 is given by Table 4', F_2 by Table 5', F_3 by Table 6', and F_4 by Table 7', respectively.

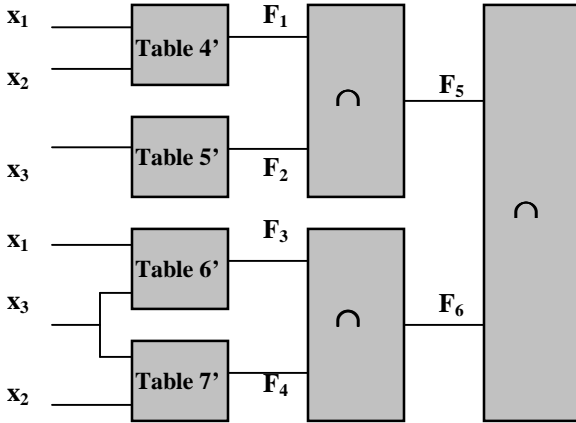


Figure 5. The decomposed structure resulting from the many-valued MRA decomposition.

The intermediate sub-functions, F_5 and F_6 are shown in the following maps, respectively.

$x_1 \backslash x_2x_3$	00	01	02	10	11	12	20	21	22
0	0,2	0,2	0,2	1	1	1	1	1	1
1	0,2	0,2	0,2	0,2	0,2	0,2	1	1	1
2	0,2	0,2	0,2	1	1	1	0,2	0,2	0,2

$x_1 \backslash x_2x_3$	00	01	02	10	11	12	20	21	22
0	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
1	0,1	0,1	2	0,1	0,1	2	0,1	0,1	0,1
2	0,1	2	0,1	0,1	0,1	0,1	2	2	0,1

$$F_6 = F_3 \cap F_4$$

THE UNION ALGORITHM

Steps 1 and 2 are the same as in the intersection algorithm.

Step 3.1: Using the decomposition model $D01:D02:D11:D12:D21:D22$ obtain $D0$, $D1$, and $D2$ by standard methods as follows:

$$\begin{aligned} D0 &= (D01 \otimes x_3) \cap (D02 \otimes x_1) \cap (D03 \otimes x_2) \\ D1 &= (D11 \otimes x_3) \cap (D12 \otimes x_1 x_2) \\ D2 &= (D21 \otimes x_2) \cap (D22 \otimes x_1) \end{aligned}$$

where $D0$ is the decomposition for function value "0", $D1$ is for function value "1", $D2$ for function value "2", and x_1, x_2 , and $x_3 \in \{0,1,2\}$.

Step 3.2: Perform the set-theoretic operations to obtain the total function from the decomposed sub-functions. This can be done using only two of the three decompositions as in Step (3.2) of the union algorithm in Example 2, or alternatively, one can use all three decompositions as follows:

$$x_1x_2x_3F = (D0 \otimes 0) \cup (D1 \otimes 1) \cup (D2 \otimes 2)$$

The function value of (x_1, x_2, x_3) is determined by the block diagram of Figure 6, where G performs the following operation:

$$\begin{aligned} F &= 0 \text{ if } (x_1x_2x_3) \in D0 \\ F &= 1 \text{ if } (x_1x_2x_3) \in D1 \\ F &= 2 \text{ if } (x_1x_2x_3) \in D2 \end{aligned}$$

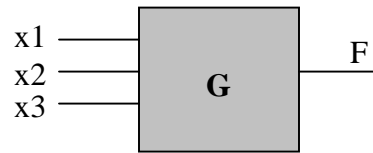


Figure 6. Block diagram for the union algorithm of MRA of Example 3.

The logic function in Example 3 is decomposable using CRA with the lossless CRA model $x_1x_2:x_2x_3:x_1x_3$. Consequently, unlike the previous example, both many-valued MRA and CRA decompose losslessly. Since both CRA and MRA decompose this function, we would like to be able to compare the complexities of the two decompositions. The complexity measure reported in (Al-Rabadi et al 2002) could be used, but needs to be extended to many-valued functions.

From the previous discussion, it follows that the extension of many-valued MRA from functions to relations is trivial. One just performs the union algorithm using all n decompositions, e.g., for three values $(D_0 \otimes 0) \cup (D_1 \otimes 1) \cup (D_2 \otimes 2)$.

4 CONCLUSION

A novel many-valued decomposition within the framework of Reconstructability Analysis is presented. In previous work (Al-Rabadi 2001, Al-Rabadi et al 2002) Modified Reconstructability Analysis (MRA) was applied to Boolean functions. In this paper, MRA is extended to many-valued logic functions and relations. It has been shown that MRA can decompose many-valued functions when CRA fails to do so. Since real-life data are naturally many-valued, future work will apply many-valued MRA to real-life data for machine learning, data mining, and data analysis.

5 REFERENCES

Al-Rabadi, A. N. 2001. "A Novel Reconstructability Analysis For the Decomposition of Boolean Functions". Technical Report #2001/005, Electrical and Computer Engineering Department, Portland State University, Portland, Oregon, 1st July 2001.

Al-Rabadi, A. N., M. Zwick, and M. Perkowski 2002. "A Comparison of Enhanced Reconstructability Analysis and Ashenhurst-Curtis Decomposition of Boolean Functions". Accepted to WOSC/IGSS'2002 Conference.

Conant, R. 1981. "Set-Theoretic Structural Modeling," *Int. J. General Systems*, Vol. 7, pp. 93-107.

Hurst, S. L. 1978. *Logical Processing of Digital Signals*. Crane Russak and Edward Arnold, London and Basel.

Klir, G. 1985. *Architecture of Systems Problem Solving*. Plenum Press, New York.

Klir, G. and M. J. Wierman, 1998. *Uncertainty-Based Information: Variables of Generalized Information Theory*, Physica-Verlag, New York.

Krippendorff, K. 1986. *Information Theory: Structural Models for Qualitative Data*. Sage Publications, Inc.

Zwick, M. 1995. "Control Uniqueness in Reconstructability Analysis". *International Journal of General Systems*, 23(2).

Zwick, M. 2001. Wholes and Parts in General Systems Methodology. In: *The Character Concept in Evolutionary Biology*, edited by G. Wagner, Academic Press.

BIOGRAPHY

Anas N. Al-Rabadi is currently a Ph.D. candidate in the Electrical and Computer Engineering Department at Portland State University, Portland, Oregon. He received his M.S. in Electrical and Computer Engineering from Portland State University in 1998 in the specialty of Power Electronics and Control Systems Design. His current research includes reconstructability analysis, reversible logic, quantum logic, and logic synthesis.

Martin Zwick is a Professor of Systems Science at Portland State University. Prior to taking his current position at PSU, he was a faculty member in the Department of Biophysics and Theoretical Biology at the University of Chicago, where he worked in macromolecular structure and mathematical crystallography. In the 1970's his interests shifted to systems theory and methodology. Since 1976 he has been on the faculty of the PSU Systems Science Ph.D. Program and during the years 1984-1989 he was the program coordinator and then director. His research interests are in discrete multivariate modeling (reconstructability analysis), "artificial life" and theoretical/computational biology, and systems philosophy.