Symbolic Representation for Rough Set Attribute Reduction Using Ordered Binary Decision Diagrams

Qianjin Wei
School of Electronic Engineering, XiDian University, Xian, P.R.China
School of Computer Science, Guilin University of Electronic Technology, Guilin, P.R.China
Email: wei_qj@guet.edu.cn

Gu Tianlong
School of Electronic Engineering, XiDian University, Xian, P.R.China
Email: tianlonggu@gmail.com

***Abstract*—The theory of rough set is the current research focus for knowledge discovery, attribute reduction is one of crucial problem in rough set theory. Most existing attribute reduction algorithms are based on algebra and information representations, discernibility matrix is a common knowledge representation for attribute reduction. As problem solving under different knowledge representations corresponding to different difficulties, by changing the method of knowledge representation, a novel knowledge representation to represent the discernibility matrix using ordered binary decision diagrams (OBDD) is proposed in this paper, the procedures to translate the discernibility matrix model to the conversion OBDD model is presented, experiment is carried to compare the storage space of discernibility matrix with that of OBDD, results show that OBDD model has better storage performance and improve the attribute reduction for those information systems with more objects and attributes, it provide the foundation for seeking new efficient algorithm of attribute reduction.**

***Index Terms*—rough set, attribute reduction, discernibility matrix, ordered binary decision diagrams**

I. INTRODUCTION

The theory of Rough set, proposed by Z. Pawlak in 1982, is a new mathematical tool to deal with imprecise, incomplete and inconsistent data [1]. The main idea of the rough set theory is obtain knowledge in the case of keeping the same ability for classification through attribution reduction. It can find the hiding and potential rules, that is knowledge, from the data without any preliminary or additional information. The rough set theory has become an attractive field in recent years, and has already been successful applied in many scientific and engineering fields such as machine learning and data mining, it is a key issue in artificial intelligence. In the rough set theory, attribute reduction is an important problem, and is one of the key steps of knowledge acquisition. Most existing attribute reduction algorithms are based on algebra and information representations [9-14]. The problem of attribute reduction is NP-hard has been demonstrated by Wong S K M and Ziarko W [6]. The main reason causing the problem is attribute combinatorial explosion, so far, there is no highly efficient attribute reduction algorithm.

An ordered binary decision diagram (OBDD) is a data structure that is used to effectively represent a Boolean function. Boolean function can use symbols to represent the state space, so the search based on OBDD can explore very large state space. In practice, OBDD has been successfully used in hardware verification, model checking, testing, assembly sequence planning and optimization of circuits. Tianlong Gu and Zhoubo Xu proposed the symbolic OBDD representations for mechanical assembly sequences [8]. A Muir, I Düntsch and G Gediga discussed rough set data representation using binary decision diagrams [5], in which, a new information system representation is presented, called BDDIS. Chen Yuming and Miao Duoqian presented searching Algorithm for Attribute Reduction based on Power Graph [7], a new knowledge representation, called power graph, is presented in those paper, therefore, searching algorithms based on power graph are also proposed. In this paper, OBDD is used to represent discernibility matrix.

This paper is organized as follows. Section II reviews some basic concepts and notations related to the theory of rough set. Section III presents the fundamentals of ordered binary decision diagram. The symbolic OBDD representation of attribute reduction is given in section IV. In section V, A novel knowledge representation to represent the discernibility matrix using ordered binary decision diagrams (OBDD) is proposed, the procedures to translate the discernibility matrix model to the conversion OBDD model is presented, In section VI the initial experimental results is presented and concludes the paper.

II. BASIC CONCEPTS OF ROUGH SET

The basic concepts, notations and results related to the theory of rough set are briefly reviewed in this section, others can be found in [1-4].

A. Information Systems for Rough Set

An information system T can be represented as a 4-tuple as follows,

T=<U, Q, V, f>

Where U is a finite nonempty set of m objects U

doi:10.4304/jsw.6.6.977-984

$=\{s_1,s_2,\ldots,s_m\}$, Q is a finite nonempty set of n attributes $Q=\{q_1,q_2,\ldots,q_n\}$, $V=\cup_{q\in Q}V_q$, where V_q is a domain of the attribute q, f is the information function , which is defined as follows:

$$f: U\times Q\rightarrow V \quad (1)$$

$$f(s, q) \in V_q \quad \forall q\in Q, \forall s\in U \quad (2)$$

Such that $f(s, q)=v$ means that the object s has the value v on attribute q. An information table is illustrated in Table I, which has five attributes and seven objects, with rows representing objects and columns representing attributes.

TABLE I. AN INFORMATION TABLE

U	a	b	c	d	e
s1	0	0	0	1	1
s2	0	1	2	0	0
s3	0	1	1	1	0
s4	1	2	0	0	1
s5	0	2	2	1	0
s6	0	3	1	0	2
s7	0	3	1	1	1

B. Indiscernibility relation and approximation of sets

Any subset P of Q determines a binary relation on U, which will be referred to as an indiscernibility relation denoted by IND (P), it is defined as the following way, two objects s_i and s_j are indiscernible by the set of attributes P in Q, if $f(s_i,q)=f(s_j,q)$ for every $q\in P$, More formally:

$$IND(P)=\{(s_i,s_j)\in U\times U \mid \forall q\in P, f(s_i,q)=f(s_j,q)\} \quad (3)$$

Obviously IND (P) is an equivalence relation. The family of all equivalence classes of IND (P) will be denoted by U/ IND (P) or simply U/P, an equivalence class of IND(P) containing s will be denoted by P(s) or $[s]_P$.

Given any subset of attributes P, any concept $X\subseteq U$ can be precisely characterized in terms of the two precise sets called the lower and upper approximations. The lower approximation, denoted by $\underline{P}X$, is the set of objects in U, which can be classified with certainty as elements in the concept X using the set of attributes P, and is defined as follows:

$$\underline{P}X=\{s_i \in U \mid [s_i]_P\subseteq X\} \quad (4)$$

The upper approximation, denoted by $\overline{P}X$, is the set of elements in U that can be possibly classified as elements in X, and is defined as follows:

$$\overline{P}X=\{s_i \in U \mid [s_i]_P\cap X\neq\varnothing\} \quad (5)$$

For any object s_i of the lower approximation of X, it is certain that it belongs to X. For any object s_i of the upper approximation of X, we can only say that it may belong to X.

C. Reduct of rough set theory and independence of attributes

Reduct is a fundamental concept of rough set. So-called attribute reduction, it means to delete those dispensable attributes with the same partition of the universe. In another words, the reduct is a minimal subset of attributes, which has the discernible power as using the entire attributes. An important task in rough set based data analysis is computation of the attribute reduction. In order to check whether the set of attributes is independent or not, one checks for every attribute whether its removal increase the number of elementary sets in information system. Given an information table T=<U, Q, V, f>, Let $q\in Q$, attribute q is dispensable in T, if IND(U) =IND(U-{q}) ,otherwise q is indispensable in T. A subset $P\subseteq Q$ is called a reduct, if P satisfies the two conditions:

$$IND(P) = IND(Q) \quad (6)$$

$$\forall q\in P, IND(P) \neq IND(P-\{q\}) \quad (7)$$

The first condition indicates the sufficiency of the attribute set P, the second condition indicates that each attribute in P is indispensable. Given an information table, there may exist many reducts, finding all reducts of information system is combinatorial NP-hard computational problem.

D. Discernibility Matrix

Given an information table T=<U, Q, V, f>, two objects are discernible if their values are different in at least one attribute, the discernibility knowledge of the information system is commonly recorded in a symmetric $|U|\times|U|$ matrix M_T (c_{ij} (s_i, sj)), called the discernibility matrix of T, which stores the sets of attributes that discern pairs of objects and helps us understand several properties to construct efficient algorithm to compute reducts. Each element c_{ij} (s_i, sj) for an object pair (s_i, sj) $\in U\times U$ is defined as follows:

$$c_{ij}(s_i, s_j)=\begin{cases}\{q\in Q \mid f(s_i, q)\neq f(s_j, q)\} & f(s_i) \neq f(s_j) \\ \varnothing & \text{otherwise}\end{cases} \quad (8)$$

Since MT (cij (si,sj)) is symmetric and cii (si, si)=∅ for i=1, 2, …, m, we represent MT (cij (si, sj)) only by elements in the lower triangle of MT (cij (si, sj)), i.e. the cij (si, sj) is with 1<j<i<m.

The physical meaning of the matrix element $c_{ij}(s_i,s_j)$ is that objects s_i and s_j can be distinguished by any attribute in $c_{ij}(s_i, s_j)$, In another words, $c_{ij}(s_i,s_j)$ is defined as the set of all attributes which discern object s_i and s_i .The pair (s_i, s_j) can be discerned if $c_{ij}(s_i, s_j) \neq\varnothing$. The discernibility matrix of Table I is shown Table II, for the underlined object pair (s_1, s_2), the entry {b, c, d, e} indicates that attribute b, c, d or e discerns the two objects.

TABLE II. DISCERNIBILITY MATRIX FOR THE INFORMATION SYSTEM IN TABLE I

	s_1	s_2	s_3	s_4	s_5	s_6	s_7
s_1							
s_2	{b,c,d,e}						
s_3	{b,c,e}	{c,d}					
s_4	{a,b,d}	{a,b,c,e}	{a,b,c,d,e}				
s_5	{b,c,e}	{b,d}	{b,c}	{a,c,d,e}			
s_6	{b,c,d,e}	{b,c,e}	{b,d,e}	{a,b,c,e}	{b,c,d,e}		
s_7	{b,c}	{b,c,d,e}	{b,e}	{a,b,c,d}	{b,c,e}	{d,e}	

III. ORDERED BINARY DECISION DIAGRAM

Let $x \rightarrow y_0, y_1$ be the *if-then-else* operator defined by

$$x \rightarrow y_0, y_1 = (x \wedge y_0) \vee (\neg x \wedge y_1) \quad (9)$$

Hence, $f \rightarrow f_0, f_1$ is true if f and f_0 are true or if f is false and f_1 is true. We call f the test expression. All operators can easily be expressed using only the *if-then-else* operator and the constants 0 and 1.Moreover, this can be done in such a way that all tests are performed only on variables and variables occur in other places. Hence the operator gives rise to a new kind of normal form. For example, $\neg x$ is $(x \rightarrow 0, 1)$, $x \Leftrightarrow y$ is $x \rightarrow (y \rightarrow 1, 0), (y \rightarrow 0, 1)$. Since variables must only occur in tests the Boolean expression x is represented as $x \rightarrow 1, 0$.

If we by $f\ [0/x]$ denote the Boolean expression obtained by replacing x with 0 in f then it is not hard to see that the following equivalence holds:

$$f = x \rightarrow f\,[1/x], f\,[0/x] \quad (10)$$

This is known as the Shannon expansion of f with respect to x.This simple equation has a lot of useful applications. The first is to generate an INF from any expression f. If f contains no variables it is either equivalent to 0 or 1 which is an INF. Otherwise we form the Shannon expansion of f with respect to one of the variables x in f. Thus since $f\,[0/x]$ and $f\,[0/x]$ both contain one less variable than f, we can recursively find INFs for both of these; call them f_0 and f_1.An INF f is now simply $x \rightarrow f_1, f_0$.

We have proved:

Proposition 1 Any Boolean expression is equivalent to an expression in INF.

Example 1 Consider the Boolean expression $f(x_1, x2, x3) = (x_1+x_2).x_3$. If we find an INF of f by selecting in order the variables $x_3<x_2<x_1$ on which to perform Shannon expansions, we get the expression

$$f = x_3 \rightarrow f_1, f_0$$
$$f_1 = x_2 \rightarrow f_{11}, f_{10}$$
$$f_0 = x_2 \rightarrow f_{01}, f_{00}$$
$$f_{11} = x_1 \rightarrow 1, 1$$
$$f_{10} = x_1 \rightarrow 1, 0$$
$$f_{01} = x_1 \rightarrow 0, 0$$
$$f_{00} = x_1 \rightarrow 0, 0$$

Figure I show the expression as a tree. Such a tree is also called a decision tree.

A lot of the expressions are easily seen to be identical, so it is tempting to identify them. For example, instead of f_{01} we can use f_{00}.

If we in fact identify all equal subexpressions we end up with what is known as a binary decision diagram. It is no longer a tree of Boolean expressions but a directed acyclic graph.

Given an n-ary Boolean function $f(x_1, x2 \ldots x_n)$, an ordered binary decision diagram (OBDD) is a finite directed acyclic graph with one root, n+1 levels, and exactly two branches at each non-terminal node. One of these is the 0 case, denoted by low(x) and drawn as a dashed line, the other the 1 case, denoted by high(x) and drawn as a solid line. The levels are determined by the fixed ordering of the variables $x_i<x_j<\ldots<x_k$. Each traversal through the tree corresponds to an assignment to the variables, and the nodes at level n+1 give the evaluation of f corresponding through this traversal. For example, Figure I shows a binary decision tree (the reduction rules are not applied) for the function $(x_1+x_2).x_3$

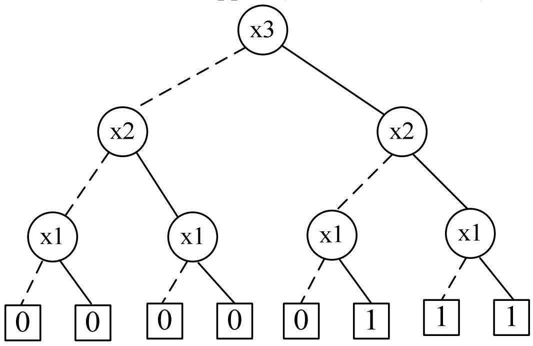

Figure I. Binary decision tree for the function (x1+x2).x3

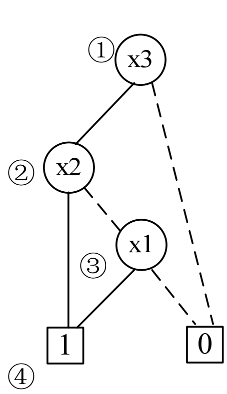

Figure II. OBDD for (x1+x2).x3

The following reduction rules do not change the value of the function:

1) Delete redundant terminal nodes and only one of them is reserved, one terminal label for 0 and one terminal label for 1, redirect all lines from level n to the respective node.
2) For non-terminal nodes *u* and *v*, if *u* and *v* are on the same level and low (*u*) = low (*v*), high (*u*) = high (*v*), then delete one of them, and all entry edges of the deleted node should point to the reserved node.
3) For non-terminal node *u*, if low (*u*) = high (*u*), then delete *u*, and all the entry edges of *u* should point to low (*u*).

The binary decision tree of the Figure I can be transformed into an ordered binary decision diagram (OBDD) by maximally reducing it according to the above reduction rules. The advantage of an OBDD is that it is unique for a particular function and variable order. This property makes it useful in functional equivalence checking and other operations like functional technology mapping. A path from the root node to the 1-terminal represents a variable assignment for which the represented Boolean function is true. Figure II shows an OBDD for the Boolean function $f(x_1,x_2,x_3)=(x_1+x_2).x_3$.We trace the path ①→②→③→④, and reach the terminal node 1. Thus, the value of Boolean function $f(x_1,x_2,x_3)=(x_1+x_2).x_3$ of variable assignment(0,1,0) is 1.

OBDD have some interesting properties. They provide compact representations of Boolean expressions, and there are efficient algorithms for performing all kinds of logical operations on OBDD. They are all based on the crucial fact that for any Boolean function *f* there is exactly one OBDD representing it. This means, in particular, that there is exactly one OBDD for the constant true function. Hence, it is possible to test in constant time whether an OBDD is constantly true or false.

IV. SYMBOLIC OBDD REPRESENTATION OF ATTRIBUTE REDUCTION

Let T=<U, Q, V, f> is an information system which have *n* attributes, there may exist many reducts. Any reduct can be characterized by its set of attributes, and can be represented by an *n*-dimensional binary vector $[x_0x_1\ldots x_n]$, in which 1 component (x_i) indicates that the corresponding attribute is indispensable and 0component (x_i') indicates that the corresponding attribute is dispensable, all the reducts is denoted by *m*-variables Boolean function $\Psi(x_1,x_2, \ldots x_n)$. For example, consider the attribute reduction of the information table in Table I, any reduct can denoted by 5-dimensional binary vector $[x_0x_1\ x_2\ x_3x_4]$,the component from left to right correspond to attribute a, b, c, d and e respectively. The initial state is 11111 or $x_1x_2x_3x_4x_5$, for the two reducts {c, d, e} and {b, d}, the reduct {c, d, e} corresponds to 00111 or $x_0'x_1'x_2x_3x_4$ and other corresponds to 01010 or $x_0'x_1x_2'x_3x_4'$. All the reducts is denoted by 5-variables Boolean function $\Psi(x_0,x_1,x_2,x_3,x_4)= x_0'x_1'x_2x_3x_4+x_0'x_1x_2'x_3x_4'$,Figure III shows the OBDD representation of attribute reduction related to information table in Table I.

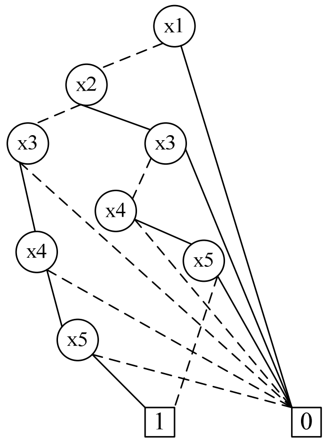

Figure III OBDD of reducts

V. BINARY DISCERNIBILITY FUNCTION AND OBDD REPRESENTATION OF BINARY DISCERNIBILITY FUNCTION

A. Binary discernibility function

Two objects are discernible if their values are different in at least one attribute, A.Skowron suggested a matrix representation for storing the sets of attributes that discern pairs of objects, called a discernibility matrix, which help us to construct efficient algorithm to compute reducts. According to calculate the discernibility function relating to the discernibility matrix, reducts can be obtained.

From a discernibility matrix, we can define a Boolean function, called discernibility function, denoted byΩ. We assign a Boolean variable "*q*" to any attribute *q*. If c_{ij} (s_i, s_j) = {q_1, $q_2\ldots$ q_k}≠∅, then allocate an Boolean function $q_1\vee q_2\vee\ldots\vee q_k$, denoted by $\Sigma c_{ij}(s_i, s_j)$, otherwise allocate Boolean constant 1. Discernibility function can be defined as follows:

$$\Omega= \prod_{(s_i,\, s_j)\,\in U\times U} \Sigma c_{ij}(s_i, s_j) \tag{11}$$

Proposition 2 Let T=<U, Q, V, f> is an information table, Ω is the discernibility function of T, attribute reduction problem is equivalent to the problem of transforming the discernibility function to a minimal disjunctive form. Each conjunctive term of the minimal disjunctive form is called a prime implicant. An attribute set P= {*q1, q2 …q*k} is a reduct if and only if the conjunction of all attributes in P, denoted by $q_1\wedge q_2\wedge q_k$, is a prime implicant ofΩ.

An important method is presented by proposition 2, in order to find the set of reducts, the discernibility function can be transformed into the minimal disjunctive form by using the absorption and distribution, accordingly, attribute reduction can be modeled based on the manipulation of a Boolean function. The procedure to compute reducts based on discernibility matrix is presented as follows:

Input: the information table T=<U, Q, V, f>

Output: the attribute reduct P

Step 1.Build the discernibility matrix M_T (c_{ij} (s_i, *sj*)) of information T.

Step 2.Compute the discernibility function Ω relating to the discernibility matrix M_T (c_{ij} (s_i, sj)).
Step 3. Calculate the minimal disjunctive normal form of Ω, All conjunctive term of the minimal disjunctive form is reduct of Q.

In this paper, an equivalent definition of a binary table to represent discernibility between pairs of objects, called binary discernibility function, and the OBDD is introduced to represent the binary discernibility function which is expressed as a discernibility function *f*, the procedures to translate the discernibility matrix model to the conversion OBDD model is presented. The binary discernibility function *f* is defined as follows:

$$U\times U\rightarrow \{0, 1\}^{|Q|} \quad (12)$$

Where U is a finite nonempty set of *m* objects U $=\{s_1,s_2,\ldots,s_m\}$, Q is a finite nonempty set of *n* attributes Q=$\{ q_1,q_2,\ldots,q_n \}$,$\{0,1\}^{|Q|}$ is *n*-dimensional binary vector space, each component corresponds to the n attributes $q_1,q_2, \ldots q_n$ respectively. Attributes q_1, q_2 …q_n is denoted by 1, 2 …*n* to facilitate the description.

Suppose object pair (s_i, s_j) ∈U×U, the value f (s_i, s_j) of discernibility function *f* is *n*-dimensional binary vector. In which, the *k*th component f_k (s_i, s_j) =0(k=1, 2 …n) indicates that s_i and s_j is indistinguishable on the attribute *k*, in another words, the object s_i has the same value *v* on attribute *k* with the object s_j. The *k*th component f_k (s_i, s_j) =1(k=1,2, …n) indicates that s_i and s_j is discernible on the attribute *k*, in another words, the object s_i has the different value *v* on attribute *k* with the object s_j. For example, Table III shows the binary discernibility function *f* for the information system in Table I, which consists of seven objects U $=\{s_1,s_2,s_3,s_4,s_5,s_6,s_7\}$, five attributes Q=$\{q_1,q_2,q_3,q_4,q_5\}$for example, $f(s_1, s_2)$ is [01111],it means that the object s_1 is distinguishable on the attribute sets {b, c, d, e} with the object s_2.

TABLE III. BINARY DISCERNIBILITY FUNCTION *f* FOR THE INFORMATION SYSTEM IN TABLE I

	f_1	f_2	f_3	f_4	f_5		f_1	f_2	f_3	f_4	f_5
(s_1,s_2)	0	1	1	1	1	(s_3,s_4)	1	1	1	1	1
(s_1,s_3)	0	1	1	0	1	(s_3,s_5)	0	1	1	0	0
(s_1,s_4)	1	1	0	1	0	(s_3,s_6)	0	1	0	1	1
(s_1,s_5)	0	1	1	0	1	(s_3,s_7)	0	1	0	0	1
(s_1,s_6)	0	1	1	1	1	(s_4,s_5)	1	0	1	1	1
(s_1,s_7)	0	1	1	0	0	(s_4,s_6)	1	1	1	0	1
(s_2,s_3)	0	0	1	1	0	(s_4,s_7)	1	1	1	1	0
(s_2,s_4)	1	1	1	0	1	(s_5,s_6)	0	1	1	1	1
(s_2,s_5)	0	1	0	1	0	(s_5,s_7)	0	1	1	0	1
(s_2,s_6)	0	1	1	0	1	(s_6,s_7)	0	0	0	1	1
(s_2,s_7)	0	1	1	1	1						

This binary discernibility function has the following properties:

1) Given object pair (si, sj) ∈U×U, if the binary discernibility function f of which is 0 vector, then object si and sj is in discernibility

Intuitively, assume f (si, sj) is 0 vector, that is, for any k (k = 1,2, ... n), with fk (si, sj) = 0, in another words, the object si has the same value on all attributes with the object sj, so objects si and sj are indistinguishable.

2) If there is an attribute q_k ,for any object pairs(s_i, s_j), f_k (s_i, s_j) = 1,then for any object pair(s_i, s_j) ,object s_i is distinguishable on the attribute the attribute q_k with s_j,Since it has only one attribute, it's a reduct of Q.

3) Given object pair (s_i, s_j) ∈U×U, If there is only one component f_k (s_i, s_j) =1, then the object s_i is discernible only on the attribute q_k with s_j, so q_k is indispensable to the reduct.

B.OBDD representation for Binary discernibility function

Suppose a binary discernibility function *f* of an information system with |U| objects and |Q| attributes, we can encode the objects with a *u*-dimensional binary vector $[x_0x_1\ldots x_{u-1}]$, where $u=\lceil\log_2 |U|\rceil$. Therefore, these seven objects of the information system in Table I can be represented by 3-dimensional binary vector $[x_0x_1x_2]$, let s_1= [001], s_2= [010], s_3= [011], s_4= [100], s_5= [101], s_6= [110] and s_7= [111]. For an object pair (s_i, s_j) ∈U×U, in which the first object can be encoded with *u*-dimensional binary vector $[x_0x_1\ldots x_{u-1}]$ and the second object encoded with u-dimensional binary vector $[y_0y_1 \ldots y_{u-1}]$. Similarly, attributes of an information system can be encoded with *v*-dimensional binary vector $[z_0z_1 \ldots z_{v-1}]$, where $v=\lceil\log_2 |Q|\rceil$. Therefore, the characteristic function of binary discernibility function denoted by $\Phi_f(x_0x_1 \ldots x_{u-1} y_0y_1 \ldots y_{n-1}\, z_0z_1 \ldots z_{v-1})$ can be formulated as:

$$\Phi_f(x_0x_1 \ldots x_{u-1}\, y_0y_1 \ldots y_{u-1}\, z_0z_1 \ldots z_{v-1})=\begin{cases} 1 & f(s_i,s_j)=1 \\ 0 & \text{othervise} \end{cases} \quad (13)$$

By encoding a= [001], b= [010], c= [011], d= [100], e= [101], based on the information table in Table I, we obtain the characteristic function of binary discernibility function *f* as follows:

$$\Phi_f(x_0,x_1,x_2,y_0,y_1,y_2,z_0,z_1,z_2)=x_0'x_1'x_2y_0'y_1y_2'z_0'z_1z_2'+x_0'x_1'x_2y_0'y_1y_2'\, z_0'z_1z_2+ x_0'x_1'x_2y_0'y_1y_2'\, z_0z_1'z_2'+x_0'x_1'x_2y_0'y_1y_2'\, z_0z_1'z_2+ x_0'x_1'x_2y_0'y_1y_2z_0'z_1z_2'+x_0'x_1'x_2y_0'y_1y_2\, z_0'z_1z_2+ x_0'x_1'x_2y_0'y_1y_2\, z_0z_1'z_2+x_0'x_1'x_2y_0y_1'y_2'\, z_0'z_1'z_2+ x_0'x_1'x_2y_0y_1'y_2'\, z_0'z_1z_2'+x_0'x_1'x_2y_0y_1'y_2'\, z_0z_1'z_2'+ x_0'x_1'x_2y_0y_1'y_2\, z_0'z_1z_2'+ x_0'x_1'x_2y_0y_1'y_2\, z_0'z_1z_2+ x_0'x_1'x_2y_0y_1'y_2\, z_0z_1'z_2+x_0'x_1'x_2\, y_0y_1y_2'\, z_0'z_1z_2'+ x_0'x_1'x_2\, y_0y_1y_2'\, z_0'z_1z_2+ x_0'x_1'x_2\, y_0y_1y_2'\, z_0z_1'z_2'+x_0'x_1'x_2\, y_0y_1y_2'\, z_0z_1'z_2+x_0'x_1'x_2\, y_0y_1y_2\, z_0'z_1z_2'+ x_0'x_1'x_2\, y_0y_1y_2\, z_0'z_1z_2+x_0'x_1x_2'\, y_0'y_1y_2\, z_0'z_1z_2+ x_0'x_1x_2'\, y_0'y_1y_2\, z_0z_1'z_2'+x_0'x_1x_2'\, y_0y_1'y_2'\, z_0'z_1'z_2+ x_0'x_1x_2'\, y_0y_1'y_2'\, z_0'z_1z_2'+x_0'x_1x_2'\, y_0y_1'y_2'\, z_0'z_1z_2+ x_0'x_1x_2'\, y_0y_1'y_2\, z_0z_1'z_2+ x_0'x_1x_2'\, y_0y_1'y_2\, z_0'z_1z_2'+ x_0'x_1x_2'\, y_0y_1y_2'\, z_0z_1'z_2'+x_0'x_1x_2'\, y_0y_1y_2'\, z_0'z_1z_2'+ x_0'x_1x_2'\, y_0y_1y_2'\, z_0'z_1z_2+ x_0'x_1x_2'\, y_0y_1y_2'\, z_0z_1'z_2+ x_0'x_1x_2'\, y_0y_1y_2$$

$z_0'z_1z_2'+x_0'x_1x_2' \quad y_0y_1y_2 \quad z_0'z_1z_2+ \quad x_0'x_1x_2' \quad y_0y_1y_2$
$z_0z_1'z_2'+ \quad x_0'x_1x_2' \quad y_0y_1y_2 \quad z_0z_1'z_2+ \quad x_0'x_1x_2 \quad y_0y_1'y_2'$
$z_0'z_1'z_2+x_0'x_1x_2 \quad y_0y_1'y_2' \quad z_0'z_1z_2'+ \quad x_0'x_1x_2 \quad y_0y_1'y_2'$
$z_0'z_1z_2+ \quad x_0'x_1x_2 \quad y_0y_1'y_2' \quad z_0z_1'z_2'+ \quad x_0'x_1x_2 \quad y_0y_1'y_2'$
$z_0z_1'z_2+x_0'x_1x_2 \quad y_0y_1'y_2 \quad z_0'z_1z_2'+ \quad x_0'x_1x_2 \quad y_0y_1'y_2$
$z_0'z_1z_2+x_0'x_1x_2 \quad y_0y_1y_2' \quad z_0'z_1z_2'+ \quad x_0'x_1x_2 \quad y_0y_1y_2'$
$z_0z_1'z_2'+x_0'x_1x_2 \quad y_0y_1y_2' \quad z_0z_1'z_2+ \quad x_0'x_1x_2 \quad y_0y_1y_2$
$z_0'z_1z_2'+ \quad x_0'x_1x_2 \quad y_0y_1y_2 \quad z_0z_1'z_2+ \quad x_0x_1'x_2' \quad y_0y_1'y_2$
$z_0'z_1'z_2+x_0x_1'x_2' \quad y_0y_1'y_2 \quad z_0'z_1z_2+ \quad x_0x_1'x_2' \quad y_0y_1'y_2$
$z_0z_1'z_2'+x_0x_1'x_2' \quad y_0y_1'y_2 \quad z_0z_1'z_2+ \quad x_0x_1'x_2' \quad y_0y_1y_2'$
$z_0'z_1'z_2+x_0x_1'x_2' \quad y_0y_1y_2' \quad z_0'z_1z_2'+ \quad x_0x_1'x_2' \quad y_0y_1y_2'$
$z_0'z_1z_2+ \quad x_0x_1'x_2' \quad y_0y_1y_2' \quad z_0z_1'z_2+ \quad x_0x_1'x_2' \quad y_0y_1y_2$
$z_0'z_1'z_2+x_0x_1'x_2' \quad y_0y_1y_2 \quad z_0'z_1z_2'+ \quad x_0x_1'x_2' \quad y_0y_1y_2$
$z_0'z_1z_2+ \quad x_0x_1'x_2' \quad y_0y_1y_2 \quad z_0z_1'z_2'+ \quad x_0x_1'x_2 \quad y_0y_1y_2'$
$z_0'z_1z_2'+x_0x_1'x_2 \quad y_0y_1y_2' \quad z_0'z_1z_2+ \quad x_0x_1'x_2 \quad y_0y_1y_2'$
$z_0z_1'z_2'+ \quad x_0x_1'x_2 \quad y_0y_1y_2' \quad z_0z_1'z_2+ \quad x_0x_1'x_2 \quad y_0y_1y_2$
$z_0'z_1z_2'+x_0x_1'x_2 \; y_0y_1y_2 \; z_0'z_1z_2+ \; x_0x_1'x_2 \; y_0y_1y_2 \; z_0z_1'z_2+$
$x_0x_1x_2' \; y_0y_1y_2 \; z_0z_1'z_2'+ \; x_0x_1x_2' \; y_0y_1y_2 \; z_0z_1'z_2$

The characteristic function is Boolean function, and can be compactly represented by OBDD. The OBDD of the discernibility matrix corresponding to $\Phi_f(x_0,x_1,x_2,y_0,y_1,y_2,z_0,z_1,z_2)$is shown in Figure IV, the path reach the terminal node 0 is omitted. Variables $x_0x_1x_2$ denote the code of the first object in object pair(s_i, s_j) and variables $y_0y_1y_2$ the second object, attribute set {a, b, c, d, e} is encoded with the variables $z_0z_1z_2$. For example, Figure IV, the rightmost path $x_0'x_1'x_2y_0'y_1y_2'z_0'z_1z_2'$ from the root node to the 1-terminal indicates object s1and s2 can be discernibility on the attribute b, that is, object s1 has the different value on attribute b with the object s2.Therefore, we obtain the OBDD of binary discernibility matrix.

Knowledge representation is necessary for the solution of a problem, problem description is for further problem-solving. As we know, problem-solving under different knowledge representations corresponding to different difficulties. From the problem description to solve the problem, there is a process, the processes involved in attribute reduction based on the OBDD representation is as follows:

Input: the information table T=<U, Q, V, f>

Output: the attribute reduct P

Step 1.Build the binary discernibility matrix of information T.

Step 2.Compute the characteristic function relating to the binary discernibility matrix and represented in OBDD

Step 3. Calculate the minimal disjunctive normal form of by performing logic operation on OBDD, All conjunctive term of the minimal disjunctive form is reduct of Q.

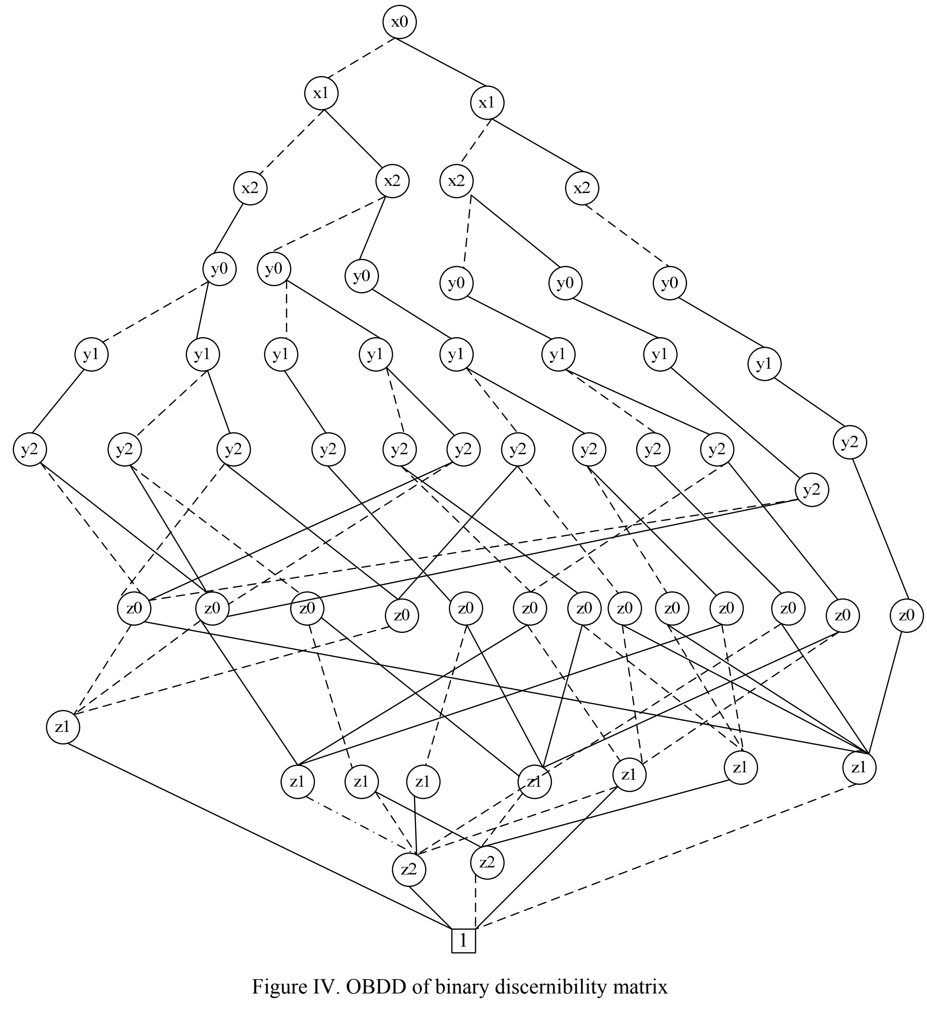

Figure IV. OBDD of binary discernibility matrix

The levels are determined by the fixed ordering of the variables $x_0<x_1<x_2<y_0<y_1<y_2<z_0<z_1<z_2$ with the OBDD showed in Figure IV. The size of the OBDD is determined both by the function being represented and the chosen ordering of the variables, called variable ordering, it is of crucial importance to care about variable ordering when applying this data structure in practice. Hence, we can improve the performance of OBDD representation for attribute reduction in the rough set theory.

VI. EXPERIMENTS AND CONCLUSIONS

A. Experimental Results

The success of the OBDD representation for discernibility matrix is currently defined in terms of its storage efficiency. The experiments are carried in windows XP and software package javaBDD, which is developed by Stanford University. The software can easily generate a variety of required BDD and have a strong capacity for manipulating BDD. The storage efficiency of symbolic OBDD representation has been tested using many random data sets, and compared with discernibility matrix, the results is shown in Table IV.

TABLE IV. Storage efficiency comparison of OBDD with discernibility matrix

	|U|	|Q|	Size(KB)	
			discernibility matrix	OBDD
1	7	5	0.19	0.23
2	50	5	9.76	8.39
3	80	6	28.58	21.15
4	100	7	53.67	35.95
5	100	9	71.23	46.87
6	150	10	175.78	120.12
7	150	12	202.56	138.23
8	200	10	299.34	181.52
9	400	7	872.69	460.26
10	500	11	2142.76	996.35

Table IV shows that symbolic OBDD representation outperforms discernibility matrix in storage efficiency, especially on complex information system with more objects and attributes. Because all identical nodes are shared and all redundant tests are eliminated, OBDD have some very convenient properties, therefore, the storage of the OBDD representation required is less than that of discernibility matrix. For example, the data of the second group in Table IV, the storage required by discernibility matrix is 1.16 times the storage required by OBDD and the data or the 10th group achieves 2.16 times. Experimental results show that the OBDD representation has better storage performance and can improve the attribute reduction of complex information system with more objects and attributes.

B. Conclusion

Attribute reduction is fundamental in rough set theory. The concept of a discernibility matrix enables us to establish a logical and theoretical foundation for reducts of an information table. This paper studies the knowledge representation for attribute reduction of the rough set theory. From a point of view to improve the knowledge representation for attribute reduction, a novel knowledge representation to represent the discernibility matrix is proposed, called symbolic OBDD representation, and the procedures to translate the discernibility matrix to the conversion OBDD representation is presented. The representation provides a new way to solve the problem of attribute reduction, which can translate computing problem of attribute reduction into searching problem in OBDD. Experiments give the proof that the OBDD representation has better storage performance. It is desirable that develops optimization symbolic algorithms for attribute reduction and work out a better heuristic function.

ACKNOWLEDGMENT

This work has been supported by National Natural Science Foundation of China (No.60963010, 60903079) and the Natural Science Foundation of Guangxi (No. 0832006Z)

REFERENCES

[1] Pawlak Z. “Rough sets.” *International Journal of Information and Computer Science*, vol.11, no.5, pp.341-356, 1982.

[2] Skowron A, “Rauszer. The discernibility matrices and functions in information systems,” in: *R. Slowinski (Ed.), Intelligent Decision Support–handbook of Applications and Advances of the Rough Sets Theory*. Dordrecht: Kluwer Academic Publishers, pp.331-362, 1992.

[3] Pawlak Z. “Rough set approach to multi-attribute decision analysis.” *European Journal of Operational Research*, vol.72, no.3, pp.443-459, 1994.

[4] Pawlak Z, Skowron A. “Rudiments of rough set.” *Information Sciences*, vol.177, no.1, pp. 3-27, 2007.

[5] A Muir, I Düntsch, G Gediga. “Rough set data representation using binary decision diagrams.” *Computational Sciences*, vol.98, no.1, pp.197-211, 2004.

[6] Wong S K M, Ziarko W. “On Optimal Decision Rules in Decision Tables.” *Bulletin of Polish Academy of Sciences*, vol.33, pp. 693-696, 1985.

[7] Chen Yuming, Miao Duoqian. “Searching Algorithm for Attribute Reduction based on Power Graph.” *Chinese Journal of Computers*, vol.32, no.8, pp.1486-1492, 2009.

[8] Tianlong Gu, Zhoubo Xu, Zhifei Yang. “Symbolic OBDD representations for mechanical assembly sequences.” *Computer-Aided Design*, vol.40, pp.411-421, 2008.

[9] YiYu Yao, Yan Zhao, and Jue Wang, “On Reduct Construction Algorithms,” *Proceedings of the First International Conference on Rough Sets and Knowledge Technologies*, Chongqing, China, pp.297-304, 2006.

[10] Yiyu Yao, Yan Zhao. “Discernibility matrixsimplification for constructing attribute ruducts.” *Information Sciences*, vol.179, no.7, pp. 867-882, 2009.
[11] K.Thangavel, A. “Pethalakshmi. Dimensionality reduction based on rough set theory: A review.” *Applied Soft Computing*, no.9, pp. 1-12, 2009.
[12] Hu X H, Cercone N. “Learning in relational databases: A rough set approach.” *Computational Intelligence*, vol.11, no.2, pp.323-337, 1995.
[13] Ye Dongyi, Chen Zhaojiong. “A new discernibility matrix and the computation of a core.” *Acta Electronica Sinica*, vol.30no.7, pp.1086-1088, 2002.
[14] Wang Guoyin. “Calculation methods for core attributes of decision table.” *Chinese Journal of Computer*, vol.26, no. 5, pp. 611-615, 2003.
[15] Liu Qing. Rough sets and Rough Reasoning. Beijing: Science Press, 2001
[16] Gu Tianlong, Xu Zhoubo. Ordered binary decision diagram and it s application. Beijing: Science Press, pp.22-57, 2009.

Qianjin Wei was born in Hunan, China, on March 1978. He received the Bachelor Degree from Harbin Institute of Technology in 2000, received Master Degree from Kunming University of Science and Technology in 2003. He is currently a Ph.d.candidate in School of Electronic Engineering at XiDian University. His main research interests include formal method, rough set, and knowledge engineering.

Gu Tianlong He received his Ph.D. degree from Zhejiang University in 1996. From 1998 to 2002, he was a postdoctoral research fellow and visiting professor within Murdoch University and Curtin University of Technology, Australia. He has published more than 130 papers, and authored 6 books. His main research interests include formal method, knowledge engineering and mobile computing. He is currently a Professor and PhD supervisor in School of Electronic Engineering at XiDian University.