

# Searching a Minimal Semantically-Equivalent Subset of a Set of Partial Values\*

FRANK S.C. TSENG

Department of Computer Science and  
Information Engineering  
National Chiao Tung University  
Hsinchu, Taiwan 30050  
Republic of China

*Email: dcp77806@csunix.csie.nctu.edu.tw*

ARBEE L.P. CHEN<sup>†</sup>

Department of Computer Science  
National Tsing Hua University  
Hsinchu, Taiwan 30043  
Republic of China

*Email: alpchen@cs.nthu.edu.tw*

*Fax: 886-35-723694*

WEI-PANG YANG

Department of Computer Science and  
Information Engineering  
National Chiao Tung University  
Hsinchu, Taiwan 30050  
Republic of China

## ABSTRACT

Imprecise data exist in databases due to their unavailability or data/schema incompatibilities in a multidatabase system. The notion of partial values has been employed for representing imprecise data. Manipulation of partial values is therefore needed for processing queries involving imprecise data. In this paper, we study the problem of eliminating redundant partial values which may result from a projection on an attribute with partial values. The redundancy of partial values is defined through the interpretation of a set of partial values. This problem is equivalent to searching a minimal semantically-equivalent subset of a set of partial values. A semantically-equivalent subset contains exactly the same information as the original set. We derive a set of useful properties and apply a graph matching technique to develop an efficient algorithm for searching such a minimal subset and therefore eliminating redundant partial values. By this process, we not only provide a concise answer to the user, but also reduce the communication cost when partial values are requested to be transmitted from one site to another site in a distributed environment. Moreover, further manipulation of the partial values can be simplified. Finally, this work is also extended for the case of multi-attribute projections.

---

\*This research was partially supported by the Republic of China National Science Council under Contract No. NSC 81-0408-E-007-12.

<sup>†</sup>To whom all correspondence should be sent.

*keywords: imprecise data, multidatabase systems, partial values, graph matching.*

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Basic Concepts and Definitions</b>	<b>4</b>
<b>3</b>	<b>Our Approach for Eliminating Redundant Partial Values</b>	<b>7</b>
3.1	Finding All Minimal Elements of $\Phi$ . . . . .	7
3.2	Finding the Upper Bound of $\mathcal{F}(\Phi)$ . . . . .	10
3.3	Properties of a Set of Partial Values . . . . .	12
3.4	Matching in a Graph . . . . .	15
<b>4</b>	<b>Finding a Minimum Sufficient Subset</b>	<b>20</b>
<b>5</b>	<b>An Extension on Multi-Attribute Projections</b>	<b>22</b>
<b>6</b>	<b>Conclusions and Discussions</b>	<b>23</b>

# List of Figures

1	The Deterministic Graph of Example 3.1. . . . .	12
2	The Relationship among $\mathcal{F}(\mathcal{M}(\Phi))$ , $\mathcal{F}(\Phi')$ , $\mathcal{F}(\Phi)$ , and $\mathcal{F}^*(\Phi)$ . . . . .	13
3	The Partitions of $\Sigma$ and $\Phi$ . . . . .	18
4	If $\eta_y \in N(\{x\})$ in $G$ , then $M^*$ can be augmented into $M^{**}$ , which is impossible. . . . .	19
5	(a) The Maximum Matching $M^*$ (b) The Maximum Matching $M^{**}$ . . . . .	22
6	Two Equivalent Relations (a) $\pi_{A_1, A_2, \dots, A_m}(R)$ and (b) $\pi_{A_1 A_2 \dots A_m}(R)$ . . . . .	23

# 1 Introduction

Handling imprecise data in database systems is an important work for reflecting the real world phenomenon. *Null values* were originally adopted to represent the meaning of “values unknown at present” in database systems. Codd [5] pioneers the work on extended relational algebra to manipulate null values. From then on, incomplete information in relational databases have been extensively studied [3][11][15][21][22][23]. The update semantics with null values in relational databases have been discussed in [1][2]. Besides, the relationship between null values and functional dependencies has been studied in [16][20][33][34]. Codd [6][7] distinguishes null values into *applicable* and *inapplicable* null values. An inapplicable null value is used to denote the case that an attribute is not applicable to a given object. For example, if Mary has not married yet, then Mary’s spouse can be recorded as an inapplicable null value. For a concise review of handling null values by algebraic approaches, refer to [24].

The concept of applicable null values has been generalized to the concept of *partial values* by Grant [12]. Instead of being treated as an atomic value, an attribute value in a table is considered as a nonempty subset of the corresponding domain. A partial value in Grant [12] is represented as an interval such that exactly one of the values in the interval is the “true” value of the partial value. In our work, however, a partial value is considered to correspond to a finite set of *possible* values such that exactly one of the values in that set is the “true” value of the partial value. Therefore, an applicable null value is a partial value which corresponds to the whole domain of the corresponding attribute. For example, if we do not know Mary’s age then Mary’s age can be recorded as an applicable null value, which can be regarded as a partial value  $[0, \dots, 120]$  if the domain of age is  $\{0, \dots, 120\}$ . However, if we know Mary’s age is either 25 or 28, then it can be recorded as a partial value  $[25, 28]$ . In [21], Lipski presents a general discussion for manipulating imprecise information including partial values. Besides, we discuss the implementation of a division operation over partial values in [30] and study some aggregate operations over partial values in [31].

In addition to incomplete data manipulation, partial values are also important in re-

solving the semantic discrepancies in multidatabase systems. For example, DeMichiel [8] employs partial values for resolving domain mismatch problems in multidatabase systems, and proposes an algebraic approach for operating on partial values. In DeMichiel’s approach, data imprecision comes from data incompatibilities in a multidatabase system. For example, suppose we want to integrate the following relations located in different sites in a multidatabase system.

<b>CS-Researchers</b>			<b>Taiwan-Scientists</b>		
<i>name</i>	<i>specialty</i>	<i>age</i>	<i>name</i>	<i>specialty</i>	<i>age</i>
Frank	DB	26	Frank	CS	26
Jesse	AI	30	Jesse	CS	30
Annie	SE	28	Andy	CS	25

Site 1 Site 2

Assuming that Computer Science (CS) consists of three subareas, i.e., database (DB), artificial intelligence (AI), and software engineering (SE), we can use partial values to resolve the mismatched domain, *specialty*. That is, the relation **Taiwan-Scientists** can be transformed into

<b>Taiwan-Scientists'</b>		
<i>name</i>	<i>specialty</i>	<i>age</i>
Frank	[DB, AI, SE]	26
Jesse	[DB, AI, SE]	30
Andy	[DB, AI, SE]	25

We can now “integrate” **CS-Researchers** and **Taiwan-Scientists'** into the following relation **Taiwan-CS-Scientists** for global multidatabase queries.

<b>Taiwan-CS-Scientists</b>		
<i>name</i>	<i>specialty</i>	<i>age</i>
Frank	DB	26
Jesse	AI	30
Annie	SE	28
Andy	[DB, AI, SE]	25

We further generalize the concept of partial values into probabilistic partial values [29] for resolving more interoperability problems, and for joining relations on incompatible keys [28] in multidatabase systems.

In this paper, we study the problem of eliminating redundant partial values which may result from a projection on an attribute with partial values. The redundancy of

partial values is defined through the interpretation of a set of partial values. This problem is equivalent to searching a minimal semantically-equivalent subset of a set of partial values. A semantically-equivalent subset contains exactly the same information as the original set. We derive a set of useful properties and apply a graph matching technique to develop an efficient algorithm for searching such a minimal subset and therefore eliminating redundant partial values.

The motivation of this work is as follows. When a non-key attribute is projected, the set of values in that attribute will be obtained. For example, consider the following relation, **Employees**.

<b>Employees</b>		
...	<i>salary</i>	...
	30k	
.	30k	.
.	35k	.
.	20k	.
	35k	

If we issue the command  $\pi_{salary}(\mathbf{Employees})$ , then the answer is

$\pi_{salary}(\mathbf{Employees})$
<i>salary</i>
30k
35k
20k

This answer shows that the salaries of the employees are 30k, 35k, and 20k. Note that duplicate values have been eliminated. However, when partial values are allowed to appear in the projected attribute, how can we determine redundant partial values such that they can be eliminated? For example, let the relation **Employees** contain partial values in attribute *salary* as follows.

<b>Employees</b>		
...	<i>salary</i>	...
	20k	
.	30k	.
.	[20k, 30k]	.
.	[20k, 35k]	.
	[30k, 35k]	

If we issue the command  $\pi_{salary}(\text{Employees})$ , according to our algorithm to be introduced, the answer can be one of the followings.

$\pi_{salary}(\text{Employees})$
<i>salary</i>
20k
30k
[20k, 35k]

$\pi_{salary}(\text{Employees})$
<i>salary</i>
20k
30k
[30k, 35k]

We will claim that these two answers contain the same information as the original attribute *salary*. More precisely, since they each correspond to the following two possible sets of definite data (exactly one of the sets is correct), and so does the original attribute, they are both *semantically-equivalent* to the original attribute.

$\pi_{salary}(\text{Employees})$
<i>salary</i>
20k
30k

$\pi_{salary}(\text{Employees})$
<i>salary</i>
20k
30k
35k

This elimination process has not been studied in previous works concerning partial values. By this process, we provide a concise answer to the user. Besides, it reduces the communication cost when data are requested to be transmitted from one site to another site in a distributed environment. That is, our work can be employed for the query optimization in a distributed database system. Moreover, further manipulation of the partial values can be simplified. For example, the processing of an operation involving sets of partial values with redundancies will be cumbersome.

This paper is organized as follows. In Section 2, basic concepts and some definitions are stated. In Section 3, we first sketch our approach, then elaborate on the properties of a set of partial values. The algorithm developed for eliminating redundant partial values is presented in Section 4. Section 5 provides a generalization of this work for the case of multi-attribute projections. Finally, we conclude and discuss our relevant work in Section 6.

## 2 Basic Concepts and Definitions

Partial values model data imprecision in databases in the sense that, for an imprecise datum, its *true* value can be restricted in a specific set of possible values [8] or an interval

of values [12]. In our work, a partial value is represented by a set of *possible* values, in which exactly one of the value is *true*. This kind of partial values are also known as *disjunctive data* in [25]. It is formally defined as follows.

**DEFINITION 2.1** A *partial value*, denoted  $\eta = [a_1, a_2, \dots, a_n]$ , associates with  $n$  possible values,  $a_1, a_2, \dots, a_n$ ,  $n \geq 1$ , of the same domain, in which exactly one of the values in  $\eta$  is the “true” value of  $\eta$ .

For a partial value  $\eta = [a_1, a_2, \dots, a_n]$ , a function  $\nu$  is defined on it by DeMichiel [8], where  $\nu$  maps the partial value to its corresponding finite set of *possible* values; that is,  $\nu(\eta) = \{a_1, a_2, \dots, a_n\}$ . Notice that an *applicable null value* [6],  $\aleph$ , can be considered as a partial value with  $\nu(\aleph) = D$ , where  $D$  is the whole domain. In the following, we will use  $\eta$  and  $\nu(\eta)$  interchangeably when it does not cause confusion. For example,  $v \in \eta$  if  $v \in \nu(\eta)$ .

The *cardinality* of a partial value  $\eta$  is defined as  $|\nu(\eta)|$  by [8]. When the cardinality of a partial value equals to 1, i.e., there exists only one *possible* value, say  $d$ , in the partial value, then the partial value  $[d]$  actually corresponds to the definite value  $d$ . On the other hand, a definite value  $d$  can be represented as a partial value  $[d]$ . Besides, a partial value with cardinality greater than 1 is referred to as a *proper partial value* in [8].

For any two *proper partial values*, say  $\eta_1$  and  $\eta_2$ ,  $\eta_1 \neq \eta_2$  even if  $\nu(\eta_1) = \nu(\eta_2)$ . This is because the *true* value of  $\eta_1$  may not be the same as the *true* value of  $\eta_2$ .

**DEFINITION 2.2** If the proper partial values,  $\eta_1, \eta_2, \dots, \eta_k$ ,  $k \geq 2$ , are elements of a set of partial values,  $\Phi$ , and  $\nu(\eta_1) = \nu(\eta_2) = \dots = \nu(\eta_k)$ , then we say  $\eta_1, \eta_2, \dots, \eta_{i-1}, \eta_{i+1}, \dots, \eta_k$  are *quasi-duplicates* of  $\eta_i$ ,  $1 \leq i \leq k$ .

By Definition 2.2, if  $\Phi = \{\overbrace{[a, b]}^{\eta_1}, \overbrace{[a, b]}^{\eta_2}\}$  then  $\eta_1$  is a quasi-duplicate of  $\eta_2$ , and vice versa.

**DEFINITION 2.3** An *interpretation*,  $\alpha = (a_1, a_2, \dots, a_m)$ , of a set of partial values,  $\Phi = \{\eta_1, \eta_2, \dots, \eta_m\}$ , is an assignment of values from  $\Phi$  such that  $a_i \in \eta_i$ ,  $1 \leq i \leq m$ .

By Definition 2.3, for a set of partial values  $\Phi = \{\eta_1, \eta_2, \dots, \eta_m\}$ ,  $\eta_1 \times \eta_2 \times \dots \times \eta_m$  is the set of all interpretations of  $\Phi$ .

**DEFINITION 2.4** For an interpretation  $\alpha = (a_1, a_2, \dots, a_m)$  of a set of partial values  $\Phi = \{\eta_1, \eta_2, \dots, \eta_m\}$ , the *value set* of  $\alpha$  is denoted  $S_\alpha = \bigcup_{1 \leq i \leq m} \{a_i\}$ .

**DEFINITION 2.5** For all interpretations,  $\alpha_j$ ,  $1 \leq j \leq p$ ,  $p = |\eta_1| \times |\eta_2| \times \dots \times |\eta_m|$ , of a set of partial values  $\Phi = \{\eta_1, \eta_2, \dots, \eta_m\}$ , the *family of value sets* of  $\Phi$  is denoted  $\mathcal{F}(\Phi) = \bigcup_{1 \leq j \leq p} \{S_{\alpha_j}\}$ . If  $\Phi = \emptyset$  then define  $\mathcal{F}(\Phi) = \emptyset$ .

$\mathcal{F}(\Phi)$  is a mapping for characterizing the information content of a set of partial values in terms of the various definite sets it represents. By this, we have the following definition.

**DEFINITION 2.6** For a set of partial values  $\Phi = \{\eta_1, \eta_2, \dots, \eta_m\}$  if we have  $\mathcal{F}(\Phi - \hat{\Phi}) = \mathcal{F}(\Phi)$  for some  $\hat{\Phi} \subset \Phi$ , then those partial values in  $\hat{\Phi}$  are said to be *redundant in  $\Phi$  with respect to  $\Phi - \hat{\Phi}$* .

**EXAMPLE 2.1** Suppose there is a set of partial values  $\Phi = \{\overbrace{[a]}^{\eta_1}, \overbrace{[b]}^{\eta_2}, \overbrace{[a, b]}^{\eta_3}\}$ , then there are two interpretations,  $\alpha_1 = (a, b, a)$  and  $\alpha_2 = (a, b, b)$ , and the value sets of  $\alpha_1$  and  $\alpha_2$  are  $S_{\alpha_1} = \{a\} \cup \{b\} \cup \{a\} = \{a, b\}$  and  $S_{\alpha_2} = \{a\} \cup \{b\} \cup \{b\} = \{a, b\}$ . Therefore, the family of value sets of  $\Phi$  is

$$\mathcal{F}(\Phi) = \{S_{\alpha_1}\} \cup \{S_{\alpha_2}\} = \{\{a, b\}\} \cup \{\{a, b\}\} = \{\{a, b\}\}.$$

$\eta_3 = [a, b]$  is redundant in  $\Phi$  with respect to  $\Phi - \{\eta_3\} = \{\overbrace{[a]}^{\eta_1}, \overbrace{[b]}^{\eta_2}\}$ , for we have  $\mathcal{F}(\Phi - \{\eta_3\}) = \mathcal{F}(\Phi)$ .  $\square$

Note that, in Example 2.1, if we delete  $\eta_1$  (respectively,  $\eta_2$ ) instead of  $\eta_3$  from  $\Phi$ , then the value set  $\{b\}$  (respectively,  $\{a\}$ ), which does not belong to  $\mathcal{F}(\Phi)$ , will be derived in  $\mathcal{F}(\Phi - \{\eta_1\})$  (respectively,  $\mathcal{F}(\Phi - \{\eta_2\})$ ).

**DEFINITION 2.7** A partial value  $\eta$  in a set  $\Phi$  is *necessary* in  $\Phi$  if the deletion of  $\eta$  from  $\Phi$  makes  $\mathcal{F}(\Phi - \{\eta\}) \neq \mathcal{F}(\Phi)$ .

In Example 2.1,  $\eta_1$  and  $\eta_2$  are necessary in  $\Phi$ .

In this paper, we derive properties of a set of partial values,  $\Phi$ , and develop a polynomial time algorithm to find a minimal subset of  $\Phi$ ,  $\Phi^{**}$ , such that  $\mathcal{F}(\Phi^{**}) = \mathcal{F}(\Phi)$ .

We call  $\Phi^{**}$  a *minimal sufficient subset* of  $\Phi$ , since  $\Phi^{**}$  is sufficient to generate exactly the same family of value sets of  $\Phi$ . Therefore,  $\Phi^{**}$  and  $\Phi$  are *semantically-equivalent*. That is,  $\Phi^* = \Phi - \Phi^{**}$  is redundant in  $\Phi$  with respect to  $\Phi^{**}$ . For a set of partial values  $\Phi$ , the minimal sufficient subset of  $\Phi$  may not be unique. For example, suppose  $\Phi = \{[a], [a, b], [b, c], [a, c]\}$ . There are two minimal sufficient subsets of  $\Phi$ , namely  $\Phi_1^{**}$  and  $\Phi_2^{**}$ , where  $\Phi_1^{**} = \{[a], [a, b], [b, c]\}$  and  $\Phi_2^{**} = \{[a], [b, c], [a, c]\}$ , because  $\mathcal{F}(\Phi) = \mathcal{F}(\Phi_1^{**}) = \mathcal{F}(\Phi_2^{**}) = \{\{a, b\}, \{a, c\}, \{a, b, c\}\}$ .

### 3 Our Approach for Eliminating Redundant Partial Values

As we have defined in Definition 2.5, the computational complexity of  $\mathcal{F}(\Phi)$  is exponential. Therefore, a brute force method to compute  $\Phi^{**}$  is also exponential. In the following, we develop a polynomial time algorithm to compute  $\Phi^{**}$  based on some properties.

Our approach can be sketched as follows. We start with finding *some* necessary elements in  $\Phi$ , which correspond to all *minimal elements* [27] of  $\Phi$ . In set theory, we call  $x$  a *minimal element* of a set  $A$  if and only if (1)  $x \in A$ , (2)  $x$  is a set, and (3) for every other  $y \in A$ ,  $y \not\subset x$ . These minimal elements are then used as a kernel to find the *upper bound* of  $\mathcal{F}(\Phi)$ ,  $\mathcal{F}^*(\Phi)$ , through a *deterministic graph* (defined below).  $\mathcal{F}^*(\Phi)$  contains all possible value sets which may be generated from  $\Phi$ . By  $\mathcal{F}^*(\Phi)$ , we derive some useful properties for searching a minimal sufficient subset. Finally, the matching technique in graph theory [4] is employed to develop an efficient algorithm to achieve the goal.

#### 3.1 Finding All Minimal Elements of $\Phi$

Minimal elements are necessary and must be included in  $\Phi^{**}$  to ensure  $\mathcal{F}(\Phi) = \mathcal{F}(\Phi^{**})$ . We prove all minimal elements are necessary in  $\Phi$  by the following lemma. Notice that quasi-duplicates are ignored here. They will be considered in the matching process to be discussed in Section 4.

**LEMMA 3.1** *For a set of partial values  $\Phi = \{\eta_1, \eta_2, \dots, \eta_m\}$  without quasi-duplicates, if  $\eta_k$  is a minimal element of  $\Phi$  (i.e.,  $\eta_i \not\subset \eta_k, \forall i \neq k$ ), then  $\eta_k$  is necessary in  $\Phi$ .*

*Proof:* We distinguish two cases:

*Case 1:*  $m = 1$ . Then we have  $\Phi = \{\eta_1\}$  and  $\mathcal{F}(\Phi) \neq \emptyset$ . But  $\mathcal{F}(\Phi - \{\eta_1\}) = \emptyset$ . Therefore,  $\eta_1$  is necessary in  $\Phi$ .

*Case 2:*  $m > 1$ . If  $\eta_i \not\subseteq \eta_k, \forall i \neq k$ , then we have  $\eta_i - \eta_k \neq \emptyset, \forall i \neq k$ . Therefore, there exists an interpretation,  $\alpha' = (a'_1, a'_2, \dots, a'_{k-1}, a'_{k+1}, \dots, a'_m)$ , of  $\Phi - \{\eta_k\}$ , such that  $a'_i \in \eta_i - \eta_k, \forall i \neq k$ . That is,  $a'_i \notin \eta_k, \forall i \neq k$ . Because the value set of  $\alpha'$  is  $S_{\alpha'} = \bigcup_{i \neq k} \{a'_i\}$ , we have  $S_{\alpha'} \cap \eta_k = \emptyset$ . But, for all interpretations,  $\alpha_j = (a_{1j}, a_{2j}, \dots, a_{kj}, \dots, a_{mj}), 1 \leq j \leq |\eta_1| \times |\eta_2| \times \dots \times |\eta_m|$ , of  $\Phi$ , we have  $a_{kj} \in \eta_k$  and the corresponding value sets  $S_{\alpha_j} = \bigcup_{1 \leq i \leq m} \{a_{ij}\}$ . That is,  $a_{kj} \in (S_{\alpha_j} \cap \eta_k) \neq \emptyset, 1 \leq j \leq |\eta_1| \times |\eta_2| \times \dots \times |\eta_m|$ , which implies  $S_{\alpha_j} \neq S_{\alpha'}, \forall j$ . Therefore,  $S_{\alpha'} \in (\mathcal{F}(\Phi - \{\eta_k\}) - \mathcal{F}(\Phi)) \neq \emptyset$ , which completes the proof.  $\square$

We denote  $\mathcal{M}(\Phi) = \{\eta_k \mid \eta_i \not\subseteq \eta_k, \eta_i, \eta_k \in \Phi, \forall i \neq k\}$  to be the set of all *minimal elements* of  $\Phi$  which contains no quasi-duplicates. Note that  $\mathcal{M}(\Phi)$  may be just a subset of the set of *all* the necessary elements of  $\Phi$ . For example, if  $\Phi = \{[a], [b], [a, b], [b, c]\}$  then  $\mathcal{M}(\Phi) = \{[a], [b]\}$ . However, by Definition 2.7,  $[b, c]$  is also necessary in  $\Phi$ . In some cases,  $\mathcal{M}(\Phi)$  contains *all* necessary elements of  $\Phi$ . For example, if  $\Phi = \{[a], [b], [a, b]\}$ , then  $\mathcal{M}(\Phi) = \{[a], [b]\}$  contains all the necessary elements of  $\Phi$ . Besides,  $\mathcal{M}(\Phi) \neq \emptyset, \forall \Phi \neq \emptyset$ .

If we consider a partial value  $\eta_i$  to be *subsumed* by another partial value  $\eta_j$  if  $\eta_j \subseteq \eta_i$ ,  $\mathcal{M}(\Phi)$  can be obtained from  $\Phi$  by eliminating all subsumed partial values. In fact, all *minimal elements* of  $\Phi$  subsume the other non-minimal elements.

By Lemma 3.1, the following corollaries can be obtained.

**COROLLARY 3.1** *Any partial value of cardinality one in a set of partial values  $\Phi$  is a necessary element of  $\Phi$ .*

*Proof:* Directly from Lemma 3.1.  $\square$

**COROLLARY 3.2** *If all the partial values in a set of partial values  $\Phi$  have the same cardinality and there is no quasi-duplicate in  $\Phi$ , then  $\mathcal{M}(\Phi) = \Phi$ .*

*Proof:* Directly from Lemma 3.1.  $\square$

**COROLLARY 3.3** For all  $\eta_i \in \Phi - \mathcal{M}(\Phi)$ ,  $\Phi$  contains no quasi-duplicates, there exists an element  $\eta_j \in \mathcal{M}(\Phi)$  such that  $\eta_j \subset \eta_i$ .

*Proof:* Since  $\eta_i \in \Phi - \mathcal{M}(\Phi)$ , by Lemma 3.1 there exists at least an  $\eta_x \in \Phi$ , such that  $\eta_x \subset \eta_i$ . Now we choose  $\eta_x$  to have the minimum cardinality in  $\Phi$ , say  $\eta_j$ , such that  $\eta_j \subset \eta_i$ . That is, there is no element in  $\Phi$  which is a proper subset of  $\eta_j$ . Therefore, by Lemma 3.1,  $\eta_j$  must be an element of  $\mathcal{M}(\Phi)$ . That completes the proof.  $\square$

Since minimal elements of  $\Phi$  cannot be eliminated, they will be used as a kernel for finding the upper bound of  $\mathcal{F}(\Phi)$ ,  $\mathcal{F}^*(\Phi)$ . First of all, we identify  $\mathcal{M}(\Phi)$  by applying Lemma 3.1 to  $\Phi$ . We summarize the procedure of finding  $\mathcal{M}(\Phi)$  by the following procedure *Find\_All\_Minimal\_Elements*.

**Procedure** *Find\_All\_Minimal\_Elements*: (Finding  $\mathcal{M}(\Phi)$  of  $\Phi$ .)

Input: A set of partial values,  $\Phi$ , which contains no quasi-duplicates.

Output:  $\mathcal{M}(\Phi)$ .

1.  $\mathcal{M}(\Phi) = \emptyset$ ;
2. for each  $\eta_i \in \Phi$  do {
3.     if ( $|\eta_i| == 1$ ) then  $\mathcal{M}(\Phi) = \mathcal{M}(\Phi) \cup \{\eta_i\}$ ; /\* Corollary 3.1 \*/
4.     else {
5.         minimal = true; /\* a flag \*/
6.         for each  $\eta_j \in \Phi$ ,  $\eta_j \neq \eta_i$ , do
7.             if ( $\eta_j - \eta_i = \emptyset$ ) then {
8.                 minimal = false;
9.                 break; /\* exit the inner for loop \*/
10.             }
11.             if (minimal) then  $\mathcal{M}(\Phi) = \mathcal{M}(\Phi) \cup \{\eta_i\}$ ;
12.             }
13.     }
14. Output( $\mathcal{M}(\Phi)$ );

Recall that  $\mathcal{M}(\Phi)$  is defined on the set  $\Phi$  which contains no quasi-duplicates. In other words, if we want to apply *Find\_All\_Minimal\_Elements* to find a subset of necessary par-

tial values for an arbitrary  $\Phi$ , we need to eliminate all the quasi-duplicates in  $\Phi$  first. Therefore, Corollary 3.3 can be stated in a more general form as follows.

**COROLLARY 3.4** *For all  $\eta_i \in \Phi - \mathcal{M}(\Phi)$ , there exists an element  $\eta_j \in \mathcal{M}(\Phi)$ , such that  $\eta_j \subseteq \eta_i$ .*

*Proof:* For an  $\eta_i \in \Phi - \mathcal{M}(\Phi)$ , we distinguish two cases:

*Case 1:*  $\eta_i$  is a quasi-duplicate of an  $\eta_j \in \mathcal{M}(\Phi)$ . Then  $\nu(\eta_i) = \nu(\eta_j)$  and  $\eta_j \subseteq \eta_i$  holds.

*Case 2:*  $\eta_i$  is not a quasi-duplicate of any  $\eta_j \in \mathcal{M}(\Phi)$ . Then, by following the same proof in Corollary 3.3, we have  $\eta_j \subset \eta_i$  and  $\eta_j \subseteq \eta_i$  holds.  $\square$

### 3.2 Finding the Upper Bound of $\mathcal{F}(\Phi)$

Based on  $\mathcal{M}(\Phi)$ , the upper bound of  $\mathcal{F}(\Phi)$ ,  $\mathcal{F}^*(\Phi)$ , can be derived by a *deterministic graph* defined as follows.

**DEFINITION 3.1** A *deterministic graph* (DG) is denoted by a 3-tuple  $(Q, \Sigma, \delta)$ , where

$Q$  is a finite set of *states*,

$\Sigma$  is a finite *input alphabet*, and

$\delta$  is a *transition function* mapping  $Q \times \Sigma$  to  $Q$ . That is,  $\delta(q, a)$  is a state

for each state  $q$  and input symbol  $a$ .

A DG can be represented by a directed graph with the vertices of the graph corresponding to the states of the DG. If there is a transition from state  $q$  to state  $p$  on input  $a$ , then there is an arc labelled  $a$  from state  $q$  to state  $p$  in the directed graph.

To derive the upper bound of  $\mathcal{F}(\Phi)$ , we employ a DG  $(Q, \Sigma, \delta)$ , with  $Q = \mathcal{F}^*(\Phi)$ ,  $\Sigma = \bigcup_{\eta_i \in \Phi} \eta_i$ , and  $\delta$  being defined as  $\delta(S_i, a_j) = S_k \in \mathcal{F}^*(\Phi)$ , where  $S_k = S_i \cup \{a_j\}$ ,  $\forall S_i \in \mathcal{F}^*(\Phi)$  and  $a_j \in \Sigma$ . Initially, we compute  $\mathcal{F}(\mathcal{M}(\Phi))$  by Definition 2.5 and then work toward  $\mathcal{F}^*(\Phi)$  by applying  $\delta$  to all the elements of  $\mathcal{F}(\mathcal{M}(\Phi))$ , which iteratively generates new states  $\delta(S_i, a_j)$ ,  $\forall S_i \in \mathcal{F}(\mathcal{M}(\Phi))$  and  $a_j \in \Sigma$ . These new states are used again to generate other new states. Therefore, by repeating this process it will monotonically increase the number of states. However, as  $\Phi$  and  $\Sigma$  are all finite, there exists a *least fixed*

point [32] such that at that point no more new states can be generated. As a matter of fact, the least fixed point is reached after  $\Sigma$  is generated as a new state. When the least fixed point is reached, we have the maximum number of states which may be generated from  $\Phi$ . Procedure *Find\_Upper\_Bound\_of\_ $\mathcal{F}(\Phi)$*  illustrates this process.

**Procedure** *Find\_Upper\_Bound\_of\_ $\mathcal{F}(\Phi)$* : (Finding the  $\mathcal{F}^*(\Phi)$ .)

Input:  $\mathcal{M}(\Phi)$ .

Output:  $\mathcal{F}^*(\Phi)$ .

1. Compute  $\mathcal{F}(\mathcal{M}(\Phi))$  by Definition 2.5;
2.  $Q = \mathcal{F}(\mathcal{M}(\Phi))$ ;
3. repeat {
4.      $c = |Q|$ ;
5.     for each  $S_i \in Q$  do
6.         for each  $a_j \in \Sigma$  do /\*  $\Sigma = \bigcup_{\eta_i \in \Phi} \eta_i$  \*/
7.              $Q = Q \cup \{\delta(S_i, a_j)\}$ ;
8.      $c' = |Q|$ ;
9. } until ( $c == c'$ ); /\* the least fixed point is reached \*/
10.  $\mathcal{F}^*(\Phi) = Q$ ;
11. Output( $\mathcal{F}^*(\Phi)$ );

The following example illustrates this process.

**EXAMPLE 3.1** Let  $\Phi = \{\overbrace{[a, b]}^{\eta_1}, \overbrace{[a, c]}^{\eta_2}, \overbrace{[b, c]}^{\eta_3}, \overbrace{[a, b, d]}^{\eta_4}, \overbrace{[a, c, d]}^{\eta_5}\}$  then we have

$$\mathcal{M}(\Phi) = \{\overbrace{[a, b]}^{\eta_1}, \overbrace{[a, c]}^{\eta_2}, \overbrace{[b, c]}^{\eta_3}\} \text{ and } \mathcal{F}(\mathcal{M}(\Phi)) = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}.$$

By the deterministic graph model, we can derive

$$\mathcal{F}^*(\Phi) = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}, \{a, b, c, d\}\}.$$

Figure 1 depicts the DG  $(Q, \Sigma, \delta)$ , where  $Q = \mathcal{F}^*(\Phi)$ ,  $\Sigma = \{a, b, c, d\}$ , and  $\delta$  is as shown in the directed graph. Note that the shaded nodes are elements in  $\mathcal{F}(\mathcal{M}(\Phi))$ .  $\square$

**LEMMA 3.2** For all  $\Phi'$ ,  $\mathcal{M}(\Phi) \subseteq \Phi' \subseteq \Phi$ ,  $\mathcal{F}(\Phi') \subseteq \mathcal{F}^*(\Phi)$ .

Figure 1: The Deterministic Graph of Example 3.1.

*Proof:* Since  $\mathcal{M}(\Phi) \subseteq \Phi'$ , for any value set  $S' \in \mathcal{F}(\Phi')$ , there exists an  $S \in \mathcal{F}(\mathcal{M}(\Phi))$  such that  $S \subseteq S'$ . By  $\Phi' \subseteq \Phi$ , we have  $\Sigma' = \cup_{\eta_i \in \Phi'} \eta_i \subseteq \Sigma$ . Therefore, by the procedure *Find\_Upper\_Bound\_of\_F*( $\Phi$ ), for any  $S' \in \mathcal{F}(\Phi')$  we obtain  $S' \in \mathcal{F}^*(\Phi)$ . That completes the proof.  $\square$

**COROLLARY 3.5**  $\mathcal{F}(\Phi) \subseteq \mathcal{F}^*(\Phi)$ .

*Proof:* Directly from Lemma 3.2 when  $\Phi' = \Phi$ .  $\square$

### 3.3 Properties of a Set of Partial Values

In the following, we will show that for any  $\Phi'$ ,  $\mathcal{M}(\Phi) \subseteq \Phi' \subseteq \Phi$ ,  $\mathcal{F}(\mathcal{M}(\Phi)) \subseteq \mathcal{F}(\Phi') \subseteq \mathcal{F}(\Phi) \subseteq \mathcal{F}^*(\Phi)$ . Figure 2 illustrates this.

**LEMMA 3.3**  $\mathcal{F}(\mathcal{M}(\Phi)) \subseteq \mathcal{F}(\Phi) \subseteq \mathcal{F}^*(\Phi)$ .

*Proof:* Since  $\mathcal{F}(\Phi) \subseteq \mathcal{F}^*(\Phi)$ , we only have to show that for any value set  $S \in \mathcal{F}(\mathcal{M}(\Phi))$ ,  $S$  is also in  $\mathcal{F}(\Phi)$ . Let  $\mathcal{M}(\Phi) = \{\eta'_1, \eta'_2, \dots, \eta'_k\}$  and  $\Phi = \underbrace{\{\eta'_1, \eta'_2, \dots, \eta'_k\}}_{\mathcal{M}(\Phi)}, \underbrace{\{\eta_{k+1}, \eta_{k+2}, \dots, \eta_m\}}_{\Phi - \mathcal{M}(\Phi)}$ , where  $k = |\mathcal{M}(\Phi)|$  and  $m = |\Phi|$ . By Corollary 3.4, for all  $\eta_i$ ,  $k+1 \leq i \leq m$ , there exists an  $\eta'_j$ ,  $1 \leq j \leq k$ , such that  $\eta'_j \subseteq \eta_i$ . Therefore, for any interpretation  $\alpha' = (a'_1, a'_2, \dots, a'_k)$  of  $\mathcal{M}(\Phi)$ , we can find a corresponding interpretation  $\alpha = (a'_1, a'_2, \dots, a'_k, a_{k+1}, a_{k+2}, \dots, a_m)$  for  $\Phi$  such that  $a_i = a'_j$  if  $\eta'_j \subseteq \eta_i$ . Then the value

Figure 2: The Relationship among  $\mathcal{F}(\mathcal{M}(\Phi))$ ,  $\mathcal{F}(\Phi')$ ,  $\mathcal{F}(\Phi)$ , and  $\mathcal{F}^*(\Phi)$ .

set of  $\alpha'$  will be equal to that of  $\alpha$ . That is, for any  $S \in \mathcal{F}(\mathcal{M}(\Phi))$ ,  $S$  is also in  $\mathcal{F}(\Phi)$ . That completes the proof.  $\square$

By Lemma 3.3, we conclude the following corollary.

**COROLLARY 3.6** *For all  $\Phi'$ ,  $\mathcal{M}(\Phi) \subseteq \Phi' \subseteq \Phi$ ,  $\mathcal{F}(\mathcal{M}(\Phi)) \subseteq \mathcal{F}(\Phi') \subseteq \mathcal{F}(\Phi) \subseteq \mathcal{F}^*(\Phi)$ .*

*Proof:* We need to prove  $\mathcal{F}(\mathcal{M}(\Phi)) \subseteq \mathcal{F}(\Phi')$  and  $\mathcal{F}(\Phi') \subseteq \mathcal{F}(\Phi)$ . The proof of  $\mathcal{F}(\mathcal{M}(\Phi)) \subseteq \mathcal{F}(\Phi')$  is similar to that of Lemma 3.3, except that  $\Phi$  is replaced by  $\Phi'$  and  $\Phi' = \{\underbrace{\eta'_1, \eta'_2, \dots, \eta'_k}_{\mathcal{M}(\Phi)}, \underbrace{\eta_{k+1}, \eta_{k+2}, \dots, \eta_l}_{\Phi' - \mathcal{M}(\Phi)}\}$ , where  $k = |\mathcal{M}(\Phi)|$  and  $l = |\Phi'|$ .

Also, the proof of  $\mathcal{F}(\Phi') \subseteq \mathcal{F}(\Phi)$  is similar to that of Lemma 3.3, except that  $\mathcal{M}(\Phi)$  is replaced by  $\Phi'$ .  $\square$

The following theorem states when  $\mathcal{M}(\Phi)$  can be used as a minimal sufficient subset of  $\Phi$ .

**THEOREM 3.1**  *$\Sigma \in \mathcal{F}(\mathcal{M}(\Phi))$  if and only if  $\mathcal{F}(\mathcal{M}(\Phi)) = \mathcal{F}(\Phi) = \mathcal{F}^*(\Phi)$ .*

*Proof:* If  $\mathcal{F}(\mathcal{M}(\Phi)) = \mathcal{F}(\Phi) = \mathcal{F}^*(\Phi)$  then, by  $\Sigma \in \mathcal{F}^*(\Phi)$ , we have  $\Sigma \in \mathcal{F}(\mathcal{M}(\Phi))$ .

Conversely, by Lemma 3.3 we only have to show that  $\mathcal{F}^*(\Phi) \subseteq \mathcal{F}(\mathcal{M}(\Phi))$  if  $\Sigma \in \mathcal{F}(\mathcal{M}(\Phi))$ . That is, for any value set  $S \notin \mathcal{F}(\mathcal{M}(\Phi))$ , we want to show that  $S \notin \mathcal{F}^*(\Phi)$ , neither. By  $\Sigma \in \mathcal{F}(\mathcal{M}(\Phi))$ , we have  $S \neq \Sigma$ . We now claim that there is an  $\eta_i \in \mathcal{M}(\Phi)$ , such that  $\eta_i \subseteq \Sigma - S$ . If this is not true, then  $\eta_j \cap S \neq \emptyset$ ,  $\forall \eta_j \in \mathcal{M}(\Phi)$ . Let  $\Sigma$  be the

value set of an interpretation  $\alpha = (a_1, a_2, \dots, a_k)$  of  $\mathcal{M}(\Phi)$ , where  $k = |\mathcal{M}(\Phi)|$ , we can obtain another interpretation,  $\alpha' = (a'_1, a'_2, \dots, a'_k)$ , of  $\mathcal{M}(\Phi)$  by letting

$$\begin{cases} a'_j = a_j & \text{if } a_j \in S, & \forall j = 1, 2, \dots, k \\ a'_j \in \eta_j \cap S & \text{if } a_j \in \Sigma - S, & \forall j = 1, 2, \dots, k. \end{cases}$$

Then, the value set of  $\alpha'$  is  $S \in \mathcal{F}(\mathcal{M}(\Phi))$  — a contradiction. Hence, the claim follows.

That is, for all the interpretations of  $\mathcal{M}(\Phi)$ ,  $\alpha_j = (a_{1j}, a_{2j}, \dots, a_{ij}, \dots, a_{kj})$ , where  $k = |\mathcal{M}(\Phi)|$ , we have  $a_{ij} \in \eta_i \subseteq (\Sigma - S)$  and the corresponding value set  $S_{\alpha_j} = \bigcup_{1 \leq l \leq k} \{a_{lj}\} \neq S$ . Therefore, all the  $S_{\alpha_j} \in \mathcal{F}(\mathcal{M}(\Phi))$  contain an element of  $\Sigma - S$ . Recall that  $\mathcal{F}^*(\Phi)$  is generated from  $\mathcal{F}(\mathcal{M}(\Phi))$  by the transition function  $\delta$ , which is defined as  $\delta(S_i, a_j) = S_i \cup \{a_j\}$ ,  $S_i \in \mathcal{F}^*(\Phi)$  and  $a_j \in \Sigma$ . Thus, by the definition of  $\delta$ , all the new states generated from any value set in  $\mathcal{F}(\mathcal{M}(\Phi))$  contain an element of  $\Sigma - S$ , no matter how many times the transition function  $\delta$  are applied. That is,  $S$  cannot be an element of  $\mathcal{F}^*(\Phi)$ . Hence,  $\mathcal{F}^*(\Phi) \subseteq \mathcal{F}(\mathcal{M}(\Phi))$  and the theorem follows.  $\square$

Furthermore, we have the following theorem. It provides a more general property for a minimal sufficient subset of  $\Phi$ .

**THEOREM 3.2** *For all  $\Phi'$ ,  $\mathcal{M}(\Phi) \subseteq \Phi' \subseteq \Phi$ ,  $\Sigma \in \mathcal{F}(\Phi')$  if and only if  $\mathcal{F}(\Phi') = \mathcal{F}(\Phi) = \mathcal{F}^*(\Phi)$ .*

*Proof:* If  $\mathcal{F}(\Phi') = \mathcal{F}(\Phi) = \mathcal{F}^*(\Phi)$  then, by  $\Sigma \in \mathcal{F}^*(\Phi)$ , we have  $\Sigma \in \mathcal{F}(\Phi')$ .

Conversely, by Lemma 3.3 we only have to show that  $\mathcal{F}^*(\Phi) \subseteq \mathcal{F}(\Phi')$  if  $\Sigma \in \mathcal{F}(\Phi')$ . That is, for any value set  $S \notin \mathcal{F}(\Phi')$ , we want to show that  $S \notin \mathcal{F}^*(\Phi)$ , neither. By  $\Sigma \in \mathcal{F}(\Phi')$ , we have  $S \neq \Sigma$ . Similar to the proof in Theorem 3.1 we can claim that there is an  $\eta_i \in \Phi'$ , such that  $\eta_i \subseteq \Sigma - S$ .

That is, for all the interpretations of  $\Phi'$ ,  $\alpha_j = (a_{1j}, a_{2j}, \dots, a_{ij}, \dots, a_{kj})$ , where  $k = |\Phi'|$ , we have  $a_{ij} \in \eta_i \subseteq (\Sigma - S)$  and the corresponding value set  $S_{\alpha_j} = \bigcup_{1 \leq l \leq k} \{a_{lj}\} \neq S$ . Therefore, all the  $S_{\alpha_j} \in \mathcal{F}(\Phi')$  contain an element of  $\Sigma - S$ , which implies all the value sets in  $\mathcal{F}(\mathcal{M}(\Phi))$  contain an element of  $\Sigma - S$ . Similar to the proof in Theorem 3.1, we know  $S$  cannot be an element of  $\mathcal{F}^*(\Phi)$ . Hence,  $\mathcal{F}^*(\Phi) \subseteq \mathcal{F}(\Phi')$ , which completes the proof.  $\square$

From Theorem 3.1 if  $\Sigma \in \mathcal{F}(\mathcal{M}(\Phi))$ , then the minimal sufficient subset of  $\Phi$ ,  $\Phi^{**}$ , is  $\mathcal{M}(\Phi)$ . Theorem 3.2 provides another property to determine  $\Phi^{**}$  when  $\Sigma \notin \mathcal{F}(\mathcal{M}(\Phi))$ ;

i.e., for a minimal subset of  $\Phi$ ,  $\Phi'$ , where  $\mathcal{M}(\Phi) \subset \Phi'$ , and  $\Sigma \in \mathcal{F}(\Phi')$  then  $\Phi^{**} = \Phi'$ . Later, we will discuss how to find  $\Phi^{**}$  when  $\Sigma \notin \mathcal{F}(\Phi)$ .

### 3.4 Matching in a Graph

To efficiently determine if  $\Sigma \in \mathcal{F}(\mathcal{M}(\Phi))$  or  $\Sigma \in \mathcal{F}(\Phi')$ , the bipartite matching technique in graph theory can be used. In the following, some terminologies about a graph are given [4].

A *graph*  $G$  is denoted  $G = (V, E)$ , where  $V$ , also denoted  $V(G)$ , is the set of *vertices* and  $E$ , also denoted  $E(G)$ , is the set of *edges* in the graph. An edge  $(x, y)$  is said to join the vertices  $x$  and  $y$ . If  $(x, y) \in E$  then  $x$  and  $y$  are *adjacent* or *neighboring* vertices of  $G$ . For any set  $S \subseteq V$ , we define the *neighbor set* of  $S$  in  $G$ , denoted  $N(S)$ , to be the set of all vertices adjacent to the vertices in  $S$ . Two edges that do not share a common vertex are said to be *independent*. A set of pairwise independent edges is called a *matching*. A matching of maximum cardinality in a graph  $G$  is called a *maximum matching*. Besides, a *bipartite graph*  $G = (V, E)$  is one whose vertex set  $V$  can be partitioned into two subsets  $X$  and  $Y$ , such that each edge in  $G$  joins a vertex in  $X$  and a vertex in  $Y$ . Finally, a *subgraph* of  $G$  is any graph  $H$  such that  $V(H) \subseteq V(G)$  and  $E(H) \subseteq E(G)$ .

**DEFINITION 3.2** Let  $S = \{S_1, S_2, \dots, S_n\}$  be a family of sets and  $s = \{s_1, s_2, \dots, s_m\}$ . The *membership graph of  $S$  over  $s$*  is a bipartite graph  $G = (V, E) = (X \cup Y, E)$ , where

$$\begin{aligned} X = s &= \{s_1, s_2, \dots, s_m\}, \\ Y = S &= \{S_1, S_2, \dots, S_n\}, \text{ and} \\ E &= \{(s_i, S_j) \mid s_i \in S_j, 1 \leq i \leq m, 1 \leq j \leq n\}. \end{aligned}$$

**DEFINITION 3.3** For a bipartite graph  $G = (X \cup Y, E)$ ,  $|X| \leq |Y|$ , we say that there is a *complete matching  $M$  from  $X$  to  $Y$*  if there is a matching of cardinality  $|X|$ ; that is, each vertex in  $X$  is adjacent to a distinct vertex in  $Y$ .

The following two theorems can be used to determine whether  $\Sigma \in \mathcal{F}(\mathcal{M}(\Phi))$  or  $\Sigma \in \mathcal{F}(\Phi')$ , where  $\mathcal{M}(\Phi) \subseteq \Phi' \subseteq \Phi$ .

**THEOREM 3.3**  $\Sigma \in \mathcal{F}(\mathcal{M}(\Phi))$  if and only if, for the membership graph of  $\mathcal{M}(\Phi)$  over  $\Sigma$ ,  $G = (\Sigma \cup \mathcal{M}(\Phi), E)$ , there is a complete matching from  $\Sigma$  to  $\mathcal{M}(\Phi)$ .

*Proof:* Let  $M = \{(a_1, \eta_1), (a_2, \eta_2), \dots, (a_s, \eta_s)\}$  be a complete matching from  $\Sigma$  to  $\mathcal{M}(\Phi) = \{\eta_1, \eta_2, \dots, \eta_s, \eta_{s+1}, \dots, \eta_k\}$  in  $G = (\Sigma \cup \mathcal{M}(\Phi), E)$ , where  $s = |\Sigma|$  and  $\Sigma = \{a_1, a_2, \dots, a_s\}$ . Then we have  $a_i \in \eta_i, 1 \leq i \leq s$ . Therefore, we can find an interpretation of  $\mathcal{M}(\Phi)$ ,  $\alpha = (a_1, a_2, \dots, a_s, a_{s+1}, \dots, a_k)$ , such that its value set  $S = \bigcup_{1 \leq i \leq k} \{a_i\} = \Sigma \in \mathcal{F}(\mathcal{M}(\Phi))$ .

Conversely, if  $\Sigma = \{a_1, a_2, \dots, a_s\} \in \mathcal{F}(\mathcal{M}(\Phi))$  then there exists an interpretation of  $\mathcal{M}(\Phi)$ ,  $\alpha = (a_1, a_2, \dots, a_s, a_{s+1}, \dots, a_k)$ , such that  $a_i \in \eta_i, 1 \leq i \leq s$ . That is,  $M = \{(a_1, \eta_1), (a_2, \eta_2), \dots, (a_s, \eta_s)\}$  is a complete matching from  $\Sigma$  to  $\mathcal{M}(\Phi)$  in the membership graph  $G = (\Sigma \cup \mathcal{M}(\Phi), E)$ .  $\square$

**THEOREM 3.4** *For all  $\Phi', \mathcal{M}(\Phi) \subseteq \Phi' \subseteq \Phi, \Sigma \in \mathcal{F}(\Phi')$  if and only if, for the membership graph of  $\Phi'$  over  $\Sigma, G = (\Sigma \cup \Phi', E)$ , there is a complete matching from  $\Sigma$  to  $\Phi'$ .*

*Proof:* By replacing  $\mathcal{M}(\Phi)$  by  $\Phi'$ , the proof is the same as that of Theorem 3.3.  $\square$

In some cases, there may not exist a complete matching from  $\Sigma$  to  $\Phi$  in the membership graph  $G = (\Sigma \cup \Phi, E)$ . That is,  $\Sigma \notin \mathcal{F}(\Phi)$ . In the following, we explore how to determine  $\Phi^{**}$  in this situation. We start with a useful lemma and an important theorem as follows.

**LEMMA 3.4** *For all  $S \subseteq \Sigma$  and  $\mathcal{M}(\Phi) \subseteq \Phi' \subseteq \Phi, S \notin \mathcal{F}(\Phi')$  if and only if*

- *there is no complete matching from  $S$  to  $\Phi'$  in the membership graph of  $\Phi'$  over  $S, G = (S \cup \Phi', E)$  or*
- *there exists an  $\eta_i \in \Phi'$ , such that  $\eta_i \cap S = \emptyset$ .*

*Proof:* We prove the following equivalence statement of this lemma: For all  $S \subseteq \Sigma$  and  $\mathcal{M}(\Phi) \subseteq \Phi' \subseteq \Phi, S \in \mathcal{F}(\Phi')$  if and only if

- *there is a complete matching from  $S$  to  $\Phi'$  in the membership graph of  $\Phi'$  over  $S, G = (S \cup \Phi', E)$  and*
- *for all  $\eta_i \in \Phi', \eta_i \cap S \neq \emptyset$ .*

Suppose  $M = \{(a_1, \eta_1), (a_2, \eta_2), \dots, (a_s, \eta_s)\}$ , where  $s = |S|$  and  $S = \{a_1, a_2, \dots, a_s\}$ , is a complete matching in  $G = (S \cup \Phi', E)$  and  $\eta_i \cap S \neq \emptyset, \forall \eta_i \in \Phi'$ . Then choose the interpretation  $\alpha' = (a_1, a_2, \dots, a_s, a_{s+1}, \dots, a_k)$ , where  $k = |\Phi'|$ , of  $\Phi'$  such that  $a_i \in \eta_i \cap S, i = s + 1, \dots, k$ . That is,  $S_{\alpha'} = S \in \mathcal{F}(\Phi')$ .

Conversely, if  $S \in \mathcal{F}(\Phi')$  then there is an interpretation  $\alpha' = (a_1, a_2, \dots, a_s, a_{s+1}, \dots, a_k)$  of  $\Phi'$ , such that  $a_i \in \eta_i$ ,  $1 \leq i \leq s$ , and  $a_j \in \eta_j \cap S \neq \emptyset$ ,  $s + 1 \leq j \leq k$ . That is,  $M = \{(a_1, \eta_1), (a_2, \eta_2), \dots, (a_s, \eta_s)\}$  is a complete matching in  $G = (S \cup \Phi', E)$  and  $a_i \in (\eta_i \cap S) \neq \emptyset$ ,  $1 \leq i \leq k$ .  $\square$

Hall [13] has given a necessary and sufficient condition under which there exists a complete matching  $M$  from  $X$  to  $Y$  for a bipartite graph  $G = (X \cup Y, E)$ .

**THEOREM 3.5 (Hall [13])** *Let  $G = (X \cup Y, E)$  be a bipartite graph, then there exists a complete matching from  $X$  to  $Y$  if and only if  $|N(S)| \geq |S|$ ,  $\forall S \subseteq X$ , where  $N(S)$  is the neighbor set of  $S$ .*  $\square$

If there is a matching  $M = \{(a_1, \eta_1), (a_2, \eta_2), \dots, (a_s, \eta_s)\}$ , where  $s = |M|$ , in a membership graph  $G = (\Sigma \cup \Phi, E)$ , then denote  $M_1 = \bigcup_{1 \leq i \leq s} \{a_i\}$  and  $M_2 = \bigcup_{1 \leq i \leq s} \{\eta_i\}$ . The following theorem states how to determine  $\Phi^{**}$  when there does not exist a complete matching from  $\Sigma$  to  $\Phi$ .

**THEOREM 3.6** *If  $M^*$  is a maximum matching in the membership graph  $G = (\Sigma \cup \Phi, E)$ , then  $\mathcal{F}(\mathcal{M}(\Phi) \cup M_2^*) = \mathcal{F}(\Phi)$ .*

*Proof:* Denote  $\Phi' = \mathcal{M}(\Phi) \cup M_2^*$ . We distinguish two cases:

*Case 1:*  $|M^*| = |\Sigma|$ . That is,  $M^*$  is a complete matching from  $\Sigma$  to  $\Phi$ . Since  $\mathcal{M}(\Phi) \subseteq \Phi' \subseteq \Phi$ , by Theorem 3.2 and Theorem 3.4, we have  $\mathcal{F}(\Phi') = \mathcal{F}(\Phi) = \mathcal{F}^*(\Phi)$ .

*Case 2:*  $|M^*| < |\Sigma|$ . That is, there is no complete matching from  $\Sigma$  to  $\Phi$ . Therefore, according to  $M^*$  and  $\Phi'$ ,  $\Sigma$  can be partitioned into  $M_1^*$  and  $\Sigma - M_1^*$  and  $\Phi$  can be partitioned into  $\Phi'$  and  $\Phi - \Phi'$ . If  $\Phi - \Phi' = \emptyset$  then  $\Phi' = \Phi$ , which implies  $\mathcal{F}(\Phi') = \mathcal{F}(\Phi)$  and the theorem follows. In the following, we prove the case for  $\Phi - \Phi' \neq \emptyset$ . First of all, we claim that it is impossible for  $G$  to have an edge  $(a, b)$  such that  $a \in \Sigma - M_1^*$  and  $b \in \Phi - \Phi'$ . Otherwise, a larger matching  $M^{**} = M^* \cup \{(a, b)\}$  can be obtained (see Figure 3), which violates that  $M^*$  is a maximum matching. That is, for all  $a_i \in \Sigma - M_1^*$  and  $\eta_j \in \Phi - \Phi'$ ,  $a_i \notin \eta_j$ .

Since  $\mathcal{F}(\Phi') \subseteq \mathcal{F}(\Phi)$ , we only have to show  $\mathcal{F}(\Phi) \subseteq \mathcal{F}(\Phi')$ . That is, for any  $S \notin \mathcal{F}(\Phi')$ , we want to show that  $S \notin \mathcal{F}(\Phi)$ , neither. For any  $S \notin \mathcal{F}(\Phi')$ , we distinguish three cases as follows. Note that  $S$  cannot be  $M_1^*$ , for  $M_1^* \in \mathcal{F}(\Phi')$ .

Figure 3: The Partitions of  $\Sigma$  and  $\Phi$ .

*Case (1):*  $S \subset M_1^*$ . Since  $M^*$  is also a complete matching in  $G^* = (M_1^* \cup \Phi', E^*)$ , by Theorem 3.5 we know that  $|N(S^*)| \geq |S^*|$ , for all  $S^* \subseteq M_1^*$ . That implies  $|N(S')| \geq |S'|$ , for all  $S' \subseteq S$ . Therefore, there exists a complete matching  $M' \subseteq M^*$  from  $S$  to  $\Phi'$  in the membership graph  $G' = (S \cup \Phi', E')$ , a subgraph of  $G^*$ . Thus, by Lemma 3.4,  $S \notin \mathcal{F}(\Phi')$  implies there must exist an  $\eta_i \in \Phi'$  such that  $\eta_i \cap S = \emptyset$ . That is, for all the interpretations of  $\Phi'$ ,  $\alpha_j = (a_{1j}, a_{2j}, \dots, a_{ij}, \dots, a_{kj})$ , where  $k = |\Phi'|$ , we have  $a_{ij} \in \eta_i$ ,  $a_{ij} \notin S$ , and the corresponding value set  $S_{\alpha_j} = \bigcup_{1 \leq l \leq k} \{a_{lj}\} \neq S$ . Therefore, all the  $S_{\alpha_j} \in \mathcal{F}(\Phi')$  contain an element  $a_{ij} \notin S$ , which implies all the value sets in  $\mathcal{F}(\mathcal{M}(\Phi))$  contain an element  $a_{ij} \notin S$ . Similar to the proof in Theorem 3.1, we know  $S$  cannot be an element of  $\mathcal{F}^*(\Phi)$ , which implies  $S \notin \mathcal{F}(\Phi)$ .

*Case (2):*  $S \subseteq \Sigma - M_1^*$ . We know that for all  $a_i \in \Sigma - M_1^*$  and  $\eta_j \in \Phi - \Phi'$ ,  $a_i \notin \eta_j$ . Therefore,  $S \notin \mathcal{F}(\Phi)$ , since elements in  $\Phi - \Phi'$  cannot contribute to any element of  $\Sigma - M_1^*$ .

*Case (3):*  $S \cap M_1^* \neq \emptyset$  and  $S \cap (\Sigma - M_1^*) \neq \emptyset$ . By Lemma 3.4, for  $S \notin \mathcal{F}(\Phi')$ , either

(a) *There is no complete matching from  $S$  to  $\Phi'$  in the membership graph of  $\Phi'$  over  $S$ ,*  
 $G' = (S \cup \Phi', E')$  or

(b) *There exists an  $\eta_i \in \Phi'$  such that  $\eta_i \cap S = \emptyset$ .*

If (a) holds, then  $|N(S')| < |S'|$  in  $G' = (S \cup \Phi', E')$ , for some  $S' \subseteq S$ , which implies  $S' \not\subseteq M_1^*$ . By *Pigeonhole Principle* [19], we can find  $S'' \subseteq S'$  such that  $S''$  contains at least

two elements adjacent to only a common neighboring vertex  $\eta_x \in \Phi'$ . That is,  $|S''| \geq 2$  and  $|N(S'')| = 1$  in  $G' = (S \cup \Phi', E')$ . Since  $S' \not\subseteq M_1^*$ , we have either

(1)  $S'' \subseteq \Sigma - M_1^*$  or

(2)  $S'' \cap M_1^* \neq \emptyset$  and  $S'' \cap (\Sigma - M_1^*) \neq \emptyset$ .

If (1) holds, then we also have  $|N(S'')| < |S''|$  in  $G'' = (S \cup \Phi, E'')$ , since  $\Phi - \Phi'$  contains no neighboring vertices of  $S''$ . That is, there is no complete matching from  $S$  to  $\Phi$  in  $G''$ . By Lemma 3.4,  $S \notin \mathcal{F}(\Phi)$ .

If (2) holds, then we claim that there is only one element  $x$  in  $S''$ , such that  $x \in S'' \cap M_1^*$ . Otherwise, if there are more than one element in  $S'' \cap M_1^*$  then  $|N(S'')| > 1$ , which violates  $|N(S'')| = 1$ . Therefore,  $(x, \eta_x) \in M^*$ . Besides, we also claim that  $x$  has no neighboring vertices in  $\Phi - \Phi'$ . Otherwise, suppose  $\eta_y \in (\Phi - \Phi')$  is a neighboring vertex of  $x$ , and  $y$  is any element in  $S'' \cap (\Sigma - M_1^*)$ , then a larger matching  $M^{**} = (M^* - \{(x, \eta_x)\}) \cup \{(x, \eta_y), (y, \eta_x)\}$ ,  $|M^{**}| = |M^*| + 1$ , can be obtained (see Figure 4). That contradicts the assumption that  $M^*$  is a maximum matching in  $G$ . Therefore,  $|N(S'')| = 1 < |S''|$  is true in the graph  $G'' = (S \cup \Phi, E'')$ , which implies there is no complete matching from  $S$  to  $\Phi$  in  $G''$ . Hence,  $S \notin \mathcal{F}(\Phi)$ .

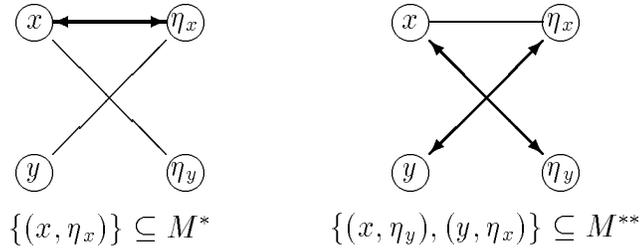


Figure 4: If  $\eta_y \in N(\{x\})$  in  $G$ , then  $M^*$  can be augmented into  $M^{**}$ , which is impossible.

If (b) holds, then similar to the proof in *Case (1)*, we have  $S \notin \mathcal{F}(\Phi)$ .

For all the cases discussed above, we conclude that for any  $S \notin \mathcal{F}(\Phi')$ ,  $S \notin \mathcal{F}(\Phi)$ , neither. That is,  $\mathcal{F}(\Phi) \subseteq (\Phi')$ . That completes the proof.  $\square$

## 4 Finding a Minimum Sufficient Subset

Based on the properties discussed above, we develop an efficient algorithm to derive  $\Phi^{**}$  in this section. As we have shown in the previous section, the bipartite matching technique plays an important role in our algorithm. Hopcroft and Karp [14] have developed an  $O(n^{5/2})$  algorithm, where  $n$  is the number of vertices, for finding a maximum matching in a bipartite graph. Due to this algorithm, Papadimitriou and Steiglitz [26] relate this problem to the max-flow problem [9] for simple networks and prove that the matching problem for bipartite graphs can be solved in  $O(|V|^{1/2} \cdot |E|)$ . Given an initial matching (including the empty matching), this matching algorithm gradually augments this matching until no augmentation can be obtained. Then, the resultant matching is a maximum matching.

By giving an initial matching, this matching algorithm will be used as a procedure in our algorithm as follows. Notice that a *complete matching* in a bipartite graph  $G$  is also a *maximum matching* in  $G$ .

**ALGORITHM 4.1** : An Algorithm That Derives a Minimum Sufficient Subset of  $\Phi$ ,  $\Phi^{**}$ .

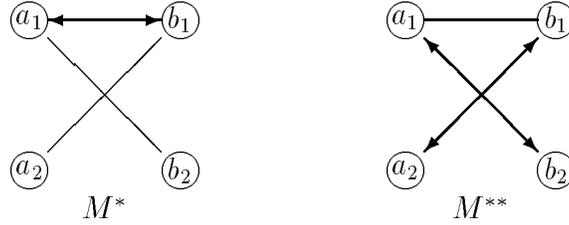
**Input:** A Set of Partial Values,  $\Phi$ .

**Output:** A Minimum Sufficient Subset of  $\Phi$ ,  $\Phi^{**}$ .

1.  $\Sigma = \bigcup_{\eta_i \in \Phi} \eta_i$ ;
2. Eliminate all quasi-duplicates of  $\Phi$  and denote the resultant set  $\Phi'$ ;
3. Call *Find\_All\_Minimal\_Elements*( $\Phi'$ ), which returns  $\mathcal{M}(\Phi')$ ;
4. Find a maximum matching  $M^*$  in the membership graph  $G = (\Sigma \cup \mathcal{M}(\Phi'), E)$   
by giving an empty matching as the initial matching;
5. If  $(|M^*| == |\Sigma|)$  then { /\* Theorems 3.1 and 3.3 \*/
6.      $\Phi^{**} = \mathcal{M}(\Phi')$ ;
7.     Output( $\Phi^{**}$ ); Stop;
8. } else {
9.     Find a maximum matching  $M^{**}$  in the membership graph  $G^* = (\Sigma \cup \Phi, E)$   
by giving  $M^*$  as the initial matching to ensure the minimality;

10.  $\Phi^{**} = \mathcal{M}(\Phi') \cup M_2^{**}; /* \text{Theorem 3.6} */$
11.  $\text{Output}(\Phi^{**}); \text{Stop};$
12. }

Note that to ensure  $\Phi^{**}$  to be minimal,  $M^*$  must be given as the initial matching when finding  $M_2^{**}$  in Step 9. That ensures  $M_2^* \subseteq M_2^{**}$ . Notice that  $M^*$  is not necessary a subset of  $M^{**}$ . For example, in the following figure,  $M^* = \{(a_1, b_1)\}$  and  $M^{**} = \{(a_1, b_2), (a_2, b_1)\}$ . However,  $M^* \not\subseteq M^{**}$  but  $M_2^* \subseteq M_2^{**}$ .



In the following, we present an example to show how the algorithm works.

**EXAMPLE 4.1** Let  $\Phi = \{\overbrace{[a, b]}^{\eta_1}, \overbrace{[b, c]}^{\eta_2}, \overbrace{[a, c]}^{\eta_3}, \overbrace{[a, c]}^{\eta_4}, \overbrace{[a, b, c]}^{\eta_5}, \overbrace{[a, c, d]}^{\eta_6}\}$ . We want to find  $\Phi^{**}$  such that  $\mathcal{F}(\Phi) = \mathcal{F}(\Phi^{**})$ . By the algorithm, we obtain  $\Sigma = \{a, b, c, d\}$  and  $\Phi' = \{\overbrace{[a, b]}^{\eta_1}, \overbrace{[b, c]}^{\eta_2}, \overbrace{[a, c]}^{\eta_3}, \overbrace{[a, b, c]}^{\eta_5}, \overbrace{[a, c, d]}^{\eta_6}\}$  in Steps 1 and 2, respectively. After Step 3, we derive  $\mathcal{M}(\Phi') = \{\overbrace{[a, b]}^{\eta_1}, \overbrace{[b, c]}^{\eta_2}, \overbrace{[a, c]}^{\eta_3}\}$ . After finding a maximum matching in the membership graph  $G = (\Sigma \cup \mathcal{M}(\Phi'), E)$ , we have one of the possible maximum matching  $M^* = \{(a, \overbrace{[a, b]}^{\eta_1}), (b, \overbrace{[b, c]}^{\eta_2}), (c, \overbrace{[a, c]}^{\eta_3})\}$ . This can be illustrated by Figure 5(a). The shaded nodes in Figure 5(a) are elements of  $M_2^*$ . Because the cardinalities of  $M^*$  and  $\Sigma$  are not identical, we continue to find another maximum matching in  $G^* = (\Sigma \cup \Phi, E)$  by giving  $M^*$  as the initial matching. This produces  $M^{**} = \{(a, \overbrace{[a, b]}^{\eta_1}), (b, \overbrace{[b, c]}^{\eta_2}), (c, \overbrace{[a, c]}^{\eta_3}), (d, \overbrace{[a, c, d]}^{\eta_6})\}$ , which implies  $M_2^{**} = \{\overbrace{[a, b]}^{\eta_1}, \overbrace{[b, c]}^{\eta_2}, \overbrace{[a, c]}^{\eta_3}, \overbrace{[a, c, d]}^{\eta_6}\}$ . Figure 5(b) depicts this. The shaded nodes in Figure 5(b) are elements of  $M_2^{**}$ .

Therefore,

$$\Phi^{**} = \mathcal{M}(\Phi') \cup M_2^{**} = \{\overbrace{[a, b]}^{\eta_1}, \overbrace{[b, c]}^{\eta_2}, \overbrace{[a, c]}^{\eta_3}, \overbrace{[a, c, d]}^{\eta_6}\}.$$

A computation of  $\mathcal{F}(\Phi)$  and  $\mathcal{F}(\mathcal{M}(\Phi') \cup M_2^{**})$  verifies the result:

Figure 5: (a) The Maximum Matching  $M^*$  (b) The Maximum Matching  $M^{**}$ .

$$\begin{aligned} \mathcal{F}(\Phi) &= \mathcal{F}(\mathcal{M}(\Phi') \cup M_2^{**}) \\ &= \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}, \{a, b, c, d\}\}. \quad \square \end{aligned}$$

The procedure *Find\_All\_Minimal\_Elements* takes  $O(n^2)$ , where  $n = |\Phi'|$ , to generate  $\mathcal{M}(\Phi')$ . Therefore, the time complexity of the algorithm is dominated by the procedure for finding a maximum matching. That is, the time complexity of the algorithm is  $O(|V|^{1/2} \cdot |E|)$ , where  $|V| = |\Sigma| + |\Phi|$  and  $|E| = \sum_{\forall \eta_i \in \Phi} |\eta_i|$ . In the worst case, this complexity is  $O(n^{5/2})$ , where  $n = |V|$ . Note that in this algorithm we do not need to generate  $\mathcal{F}^*(\Phi)$  by *Find\_Upper\_Bound\_of\_F*( $\Phi$ ).

## 5 An Extension on Multi-Attribute Projections

In general, a projection may involve more than one attribute in a relation. To cope with the redundant tuple elimination under this case, the following definition is given.

**DEFINITION 5.1** The *cartesian product*  $\eta_a \hat{\times} \eta_b$  of the partial values  $\eta_a = [a_1, a_2, \dots, a_m]$  and  $\eta_b = [b_1, b_2, \dots, b_n]$  is the partial value  $\eta_{a \hat{\times} b}$  with  $\nu(\eta_{a \hat{\times} b})$  being a set of the ordered pairs  $(a_i, b_j)$  for every  $a_i \in \eta_a$  and  $b_j \in \eta_b$ .

**EXAMPLE 5.1** The cartesian product  $\eta_a \hat{\times} \eta_b$  of the partial values  $\eta_a = [a, b, c]$  and  $\eta_b = [x, y]$  is the partial value  $\eta_{a \hat{\times} b}$  with  $\nu(\eta_{a \hat{\times} b}) = \{(a, x), (a, y), (b, x), (b, y), (c, x), (c, y)\}$ . That

is,  $\eta_{a\hat{\times}b} = [(a, x), (a, y), (b, x), (b, y), (c, x), (c, y)]$ .  $\square$

Let the result of a projection  $\pi_{A_1, A_2, \dots, A_m}(R)$ ,  $m > 1$ , be as Figure 6(a) depicts. Then the relation  $\pi_{A_1, A_2, \dots, A_m}(R)$  can be regarded as a relation  $\pi_{A_1 A_2 \dots A_m}(R)$  with the

$\pi_{A_1, A_2, \dots, A_m}(R)$			
$A_1$	$A_2$	$\dots$	$A_m$
$\eta_{11}$	$\eta_{21}$	$\dots$	$\eta_{m1}$
$\eta_{12}$	$\eta_{22}$	$\dots$	$\eta_{m2}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\eta_{1n}$	$\eta_{2n}$	$\dots$	$\eta_{mn}$

(a)

$\pi_{A_1 A_2 \dots A_m}(R)$
$A_1 A_2 \dots A_m$
$\eta_{11} \hat{\times} \eta_{21} \hat{\times} \dots \hat{\times} \eta_{m1}$
$\eta_{12} \hat{\times} \eta_{22} \hat{\times} \dots \hat{\times} \eta_{m2}$
$\vdots$
$\eta_{1n} \hat{\times} \eta_{2n} \hat{\times} \dots \hat{\times} \eta_{mn}$

(b)

Figure 6: Two Equivalent Relations (a) $\pi_{A_1, A_2, \dots, A_m}(R)$  and (b) $\pi_{A_1 A_2 \dots A_m}(R)$ .

single attribute  $A_1 A_2 \dots A_m$ . Since if the “true” value of a tuple of  $\pi_{A_1, A_2, \dots, A_m}(R)$ ,  $\boxed{\eta_{1j}, \eta_{2j}, \dots, \eta_{mj}}$ , is the m-tuple  $(a_1, a_2, \dots, a_m)$ ,  $a_i \in \eta_{ij}$ , where  $a_i$  is the “true” value of  $\eta_{ij}$ ; then the “true” value of  $\eta_{1j} \hat{\times} \eta_{2j} \hat{\times} \dots \hat{\times} \eta_{mj}$  is also  $(a_1, a_2, \dots, a_m)$ , and vice versa. We know that a tuple of  $\pi_{A_1, A_2, \dots, A_m}(R)$ ,  $\boxed{\eta_{1j}, \eta_{2j}, \dots, \eta_{mj}}$ , can be considered as a tuple of  $\pi_{A_1 A_2 \dots A_m}(R)$  with attribute value  $\eta_{1j} \hat{\times} \eta_{2j} \hat{\times} \dots \hat{\times} \eta_{mj} \equiv \eta_{1j} \hat{\times} \eta_{2j} \hat{\times} \dots \hat{\times} \eta_{mj}$ . That is, the relations  $\pi_{A_1, A_2, \dots, A_m}(R)$  and  $\pi_{A_1 A_2 \dots A_m}(R)$  are semantically equivalent and can be transformed to each other. Figure 6 illustrates this. By this transformation, a one-attribute relation can always be obtained and Algorithm 4.1 works as before.

## 6 Conclusions and Discussions

Partial values have been used to represent imprecise data in databases. Data imprecision may come from their unavailability or data/schema incompatibilities in a multidatabase system. We have studied some extended algebraic operations on partial values [30][31]. In this paper, we further consider the problem of eliminating redundant partial values which may result from a projection on an attribute with partial values. Our work provides a more concise answer for users and reduces the communication cost when partial values are requested to be transmitted from one site to another site in a distributed environment.

Therefore, our work also contributes to query optimization in a distributed database system.

By the notion of *interpretations* over a set of partial values, we define necessary and redundant partial values. We then proceed to find a subset of the necessary partial values, which is the set of all minimal elements of  $\Phi$ , and derive properties for a set of partial values. In addition, the problem of searching a minimal sufficient subset of  $\Phi$ ,  $\Phi^{**}$ , is converted into a bipartite graph matching problem. Based on the properties of partial values, we develop an efficient algorithm to find  $\Phi^{**}$  and eliminate the redundant subset  $\Phi - \Phi^{**}$ . A very interesting duality in our algorithm is that searching a *minimal* sufficient subset in a set of partial values can be achieved by finding a *maximum* matching in a bipartite membership graph.

Notice that for the *union* of two sets of partial values,  $\Phi_1$  and  $\Phi_2$ , our work can be employed as follows. Firstly, all members of  $\Phi_1$  and  $\Phi_2$  are collected together to form another set  $\Phi$ . Then, apply our work to eliminate redundant elements in  $\Phi$ .

Imieliński and Vadaparty [17][18] pointed out that if partial values are allowed to occur in databases, the data complexity of query processing jumps from PTIME to CoNP [10]. However, there also exist some types of queries which have PTIME complexity [17]. Our ongoing studies on the query processing over partial values intend to discover more PTIME algorithms from algebraic point of view. In our recent work [30][31], we found the *division* (by restricting the divisor to be definite) and some aggregate operations over partial values—*min*, *max*, and *count*—can be done in PTIME.

## Acknowledgement

The authors wish to thank the anonymous referees whose invaluable comments and suggestions helped to improve this paper substantially.

## References

- [1] S. Abiteboul and G. Grahne, "Update semantics for incomplete information," in *Proc. 11th Int. Conf. Very Large Data Bases*, Stockholm, 1985, pp.1-12.
- [2] F. Bancilhon and N. Spyratos, "Update semantics of relational views," *ACM Trans. Database Systems*, vol.6, no.4, pp.557-575, Dec. 1981.
- [3] J. Biskup, "A foundation of Codd's relational maybe-operations," *ACM Trans. Database Systems*, vol.8, no.4, pp.608-636, Dec. 1983.
- [4] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, Macmillan Press, NY, 1976.
- [5] E.F. Codd, "Extending the database relational model to capture more meaning," *ACM Trans. Database Systems*, vol.4, no.4, pp.397-434, Dec. 1979.
- [6] E.F. Codd, "Missing information (applicable and inapplicable) in relational databases," *SIGMOD RECORD*, vol.15, no.4, pp. 53-78, Dec. 1986.
- [7] E.F. Codd, "More commentary on missing information in relational databases (applicable and inapplicable information)," *SIGMOD RECORD*, vol.16, no.1, pp.42-50, Mar. 1987.
- [8] L.G. DeMichiel, "Resolving database incompatibility: An approach to performing relational operations over mismatched domains," *IEEE Trans. Knowledge and Data Engineering*, vol.1, no.4, pp.485-493, Dec. 1989.
- [9] L.R. Ford and D.R. Fulkerson, *Flows in Networks*, Princeton Univ. Press, Princeton, NJ, 1962.
- [10] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, SF, 1979.
- [11] J. Grant, "Null values in a relational data base," *Information Processing Letters*, vol.6 no.5, pp.156-157, Oct. 1977.
- [12] J. Grant, "Partial values in a tabular database model," *Information Processing Letters*, vol.9, no.2, pp.97-99, Aug. 1979.
- [13] P. Hall, "On representatives of subsets," *J. London Math. Soc.*, vol.10, pp.26-30, 1935.
- [14] J.E. Hopcroft and R.M. Karp, "An  $n^{5/2}$  algorithm for maximum matching in bipartite graphs," *SIAM J. Computing*, vol.2, no.4, pp.225-231, 1973.
- [15] T. Imieliński and W. Lipski, "On representing incomplete information in a relational database," in *Proc. 7th Int. Conf. Very Large Data Bases*, Cannes, France, Sep. 1981, pp.389-397.
- [16] T. Imieliński and W. Lipski, "Incomplete information and dependencies in relational databases," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, San Jose, CA, May 1983, pp.178-184.
- [17] T. Imieliński and K. Vadaparty, "Complexity of Query Processing in Databases with OR-Objects," *Proc. ACM Symp. on Principles of Database Systems*, (1989) 51-65.

- [18] T. Imieliński, Incomplete Deductive Databases, *Annals of Mathematics and Artificial Intelligence*, 3 (2-4) (1984) 259-294.
- [19] H. Lewis and C. Papadimitriou, *Elements of the Theory of Computation*, Prentice-Hall, Englewood Cliffs, NJ, 1981, pp.26-26.
- [20] E. Lien, "Multivalued dependencies with null values in relational databases," in *Proc. 5th Int. Conf. Very Large Data Bases*, 1979, pp.61-66.
- [21] W. Lipski, "On semantic issues connected with incomplete information systems," *ACM Trans. Database Systems*, vol.4, no.3, pp.262-296, Sept. 1979.
- [22] K.-C. Liu and R. Sunderraman, "Indefinite and maybe information in relational databases," *ACM Trans. Database Systems*, vol.15, no.1, pp.1-39, Mar. 1990.
- [23] K.-C. Liu and R. Sunderraman, "A generalized relational model for indefinite and maybe information," *IEEE Trans. Knowledge and Data Engineering*, vol.3, no.1, pp.65-77, Mar. 1991.
- [24] D. Maier, *The Theory of Relational Databases*, Computer Science Press, Rockville, MD, 1983.
- [25] A. Motro, "Accommodating imprecision in database systems: issues and solutions," *ACM SIGMOD RECORD*, Vol.19, No.4, pp.69-74, Dec. 1990.
- [26] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982, pp.221-226.
- [27] P. Suppes, *Axiomatic Set Theory*, D. Van Nostrand Company, 1960, pp.99-100.
- [28] P.S.M. Tsai and A.L.P. Chen, "Querying uncertain data in heterogeneous databases", to appear in *Proc. IEEE Int'l Workshop on Research Issues in Data Engineering (RIDE)*, 1993.
- [29] F.S.C. Tseng, A.L.P. Chen, and W.P. Yang, "A probabilistic approach to query processing in heterogeneous database systems," in *Proc. IEEE Int'l Workshop on Research Issues in Data Engineering: Transaction and Query Processing (RIDE-TQP)*, 1992, pp 176-183.
- [30] F.S.C. Tseng, A.L.P. Chen and W.P. Yang, "Implementing the division operation on a database containing uncertain data," submitted for publication, 1992.
- [31] F.S.C. Tseng, A.L.P. Chen and W.P. Yang, "Evaluating aggregate operators over imprecise data," submitted for publication, 1992.
- [32] J.D. Ullman, *Principles of Database and Knowledge-Base Systems*, Computer Science Press, Vol. 2, Rockville, MD, 1988.
- [33] Y. Vassiliou, "Null values in data base management — A denotational semantics approach," in *Proc. ACM-SIGMOD Int. Conf. Management of Data*, Boston, MA, May-June 1979, pp.162-169.
- [34] Y. Vassiliou, "Functional dependencies and incomplete information," in *Proc. 6th Int. Conf. Very Large Data Bases*, 1980, pp.260-269.