VLDB Journal, **2**, 489-512 (1993), Fred J. Maryanski, Editor

Searching a Minimal Semantically-Equivalent Subset of a Set of Partial Values

Frank S.C. Tseng, Arbee L.P. Chen, and Wei-Pang Yang

Received August 6, 1991; revised version received, July 30, 1992; accepted January 17, 1993.

Abstract. Imprecise data exist in databases due to their unavailability or to data/schema incompatibilities in a multidatabase system. Partial values have been used to represent imprecise data. Manipulation of partial values is therefore necessary to process queries involving imprecise data. In this article, we study the problem of eliminating redundant partial values that 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. This work is also extended to the case of multi-attribute projections.

Key Words. Imprecise data, minimal elements, multidatabase systems, partial values, bipartite graph, graph matching.

1. Introduction

Imprecise data, or *null values*, in database systems reflect the real world phenomenon. Null values were originally adopted to represent "values unknown at present" in database systems. Codd (1979) pioneered the work on extended relational algebra to manipulate null values. Since then, incomplete information in relational databases

Frank S.C. Tseng, Ph.D., is Professor, and Wei-Pang Yang, Ph.D., is Professor, Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan 30050, ROC; Arbee L.P. Chen, Ph.D., is Professor, Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan 30043, ROC.

has been extensively studied (Grant, 1977; Lipski, 1979; Imieliński and Lipski, 1981; Biskup, 1983; Liu and Sunderraman, 1990, 1991). Update semantics of null values in relational databases have been discussed (Bancilhon and Spyratos, 1981; Abiteboul and Grahne, 1985), as well as the relationship between null values and functional dependencies (Lien, 1979; Vassiliou, 1979, 1980; Imieliński and Lipski, 1983). Codd (1986, 1987) divided null values into *applicable* and *inapplicable* null values. An inapplicable null value denotes an attribute that is not applicable to a given object (e.g., 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, see Maier (1983).

The concept of applicable null values has been generalized to the concept of *partial values* by Grant (1979). Instead of being treated as an atomic value, an attribute value in a table is considered a nonempty subset of the corresponding domain. A partial value 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 that corresponds to the whole domain of the corresponding attribute (e.g., if we do not know Mary's age, then it can be recorded as an applicable null value, which can be regarded as a partial value [0,$\cdots$,120] if the domain of age is $\{0,\cdots,120\}$). However, if we know Mary's age is either 25 or 28, then it can be recorded as a partial value (Motro, 1990; Tsai and Chen, 1993). Lipski (1979) presented a general discussion for manipulating imprecise information, including partial values. We discussed the implementation of a division operation over partial values (Tseng et al., 1993*b*) and we studied some aggregate operations over partial values (Tseng et al., 1993*c*).

In addition to manipulating incomplete data, partial values are also important in resolving the semantic discrepancies in multidatabase systems. DeMichiel (1989) employed partial values to resolve domain mismatch problems in multidatabase systems, and proposed an algebraic approach to operate on partial values. In this approach, data imprecision comes from data incompatibilities in a multidatabase system.

Suppose we want to integrate the following relations located in different sites in a multidatabase system.

CS-Researchers

name	*specialty*	*age*
Frank	DB	26
Jesse	AI	30
Annie	SE	28

Site 1

Taiwan-Scientists

name	*specialty*	*age*
Frank	CS	26
Jesse	CS	30
Andy	CS	25

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

Taiwan-Scientists'

name	*specialty*	*age*
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.

Taiwan-CS-Scientists

name	*specialty*	*age*
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 (Tseng et al., 1993*a*) to resolve more interoperability problems, and to join relations on incompatible keys (Tsai and Chen, 1993) in multidatabase systems.

In this article, we study the problem of eliminating redundant partial values that may result from a projection on an attribute with partial values. The redundancy of partial values is defined by interpreting 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 to search such a minimal subset and therefore eliminate 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*.

Employees

...	*salary*	...
	30k	
·	30k	·
·	35k	·
·	20k	·
	35k	

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

π_{salary}(**Employees**)

salary
30k
35k
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?

Let the relation *Employees* contain partial values in the attribute *salary* as follows.

Employees

...	*salary*	...
	20k	
.	30k	.
.	[20k, 30k]	.
.	[20k, 35k]	.
	[30k, 35k]	

If we issue the command π_{salary}(Employees), according to our algorithm, the answer can be one of the followings.

π_{salary}(**Employees**)

salary
20k
30k
[20k, 35k]

π_{salary}(**Employees**)

salary
20k
30k
[30k, 35k]

These two answers contain the same information as the original attribute *salary*. More precisely, because 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.

π_{salary}(**Employees**)

salary
20k
30k

π_{salary}(**Employees**)

salary
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, and we reduce the communication costs of data transmission requests in a distributed environment (i.e., our work can be used for query optimization in a distributed database system). Moreover, we simplify further manipulation of the partial values (i.e., processing an operation involving sets of partial values with redundancies is cumbersome).

This article 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 to eliminate redundant partial values is presented in Section 4. Section 5 provides a generalization of this work for the case of multi-attribute projections. In Section 6, we conclude and discuss relevant work.

2. Basic Concepts and Definitions

Partial values model data imprecision in databases in the sense that, the *true* value of an imprecise datum can be restricted in a specific set of possible values (DeMichiel, 1989), or an interval of values (Grant, 1979). In our work, a partial value is represented by a set of *possible* values, in which exactly one of the values is *true*. These kinds of partial values are also known as *disjunctive data* (Motro, 1990). In the following, we follow the definition of a partial value given by DeMichiel (1989), which is formally stated as follows.

Definition 2.1 A *partial value*, denoted $\eta = [a_1, a_2, \ldots, a_n]$, associates with n possible values, $a_1, a_2, \ldots, a_n$, n ≥ 1, of the same domain, in which exactly one of the values in η is the "true" value of η.

For a partial value $\eta = [a_1, a_2, \ldots, a_n]$, a function ν is defined by DeMichiel (1989), where ν maps the partial value to its corresponding finite set of *possible* values; i.e., $\nu(\eta) = \{a_1, a_2, \ldots, a_n\}$. Notice that an *applicable null value* (Codd, 1986), $\aleph$, can be considered a partial value with $\nu(\aleph) = D$, where D is the whole domain. We use η 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 η is defined as $\mid \nu(\eta) \mid$ by DeMichiel (1989). When the cardinality of a partial value equals 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* (DeMichiel, 1989).

For any two *proper partial values*, say η_1 and η_2, $\eta_1 \neq \eta_2$ even if $\nu(\eta_1) = \nu(\eta_2)$. This is because the *true* value of η_1 may not be the same as the *true* value of η_2.

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

By Definition 2.2, if $\Phi = \{\overbrace{[a, b]}^{\eta_1}, \overbrace{[a, b]}^{\eta_2}\}$ then η_1 is a quasi-duplicate of η_2, and vice versa.

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

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

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

Definition 2.5 For all interpretations, α_j, $1 \leq j \leq p, p = |\eta_1| \times |\eta_2| \times \cdots \times |\eta_m|$, of a set of partial values $\Phi = \{\eta_1, \eta_2, \ldots, \eta_m\}$, the *family of value sets* of Φ 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 $\Phi = \{\eta_1, \eta_2, \ldots, \eta_m\}$ 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* Φ *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 α_1 and α_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 Φ 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 Φ with respect to $\Phi - \{\eta_3\} = \{\overbrace{[a]}^{\eta_1}, \overbrace{[b]}^{\eta_2}\}$, for we have $\mathcal{F}(\Phi - \{\eta_3\}) = \mathcal{F}(\Phi)$. □

Note that, in Example 2.1, if we delete η_1 (respectively, η_2) instead of η_3 from Φ, 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 η in a set Φ is *necessary* in Φ if the deletion of η from Φ makes $\mathcal{F}(\Phi - \{\eta\}) \neq \mathcal{F}(\Phi)$.

In Example 2.1, η_1 and η_2 are necessary in Φ.

In this article, we derive properties of a set of partial values, Φ, and develop a polynomial time algorithm to find a minimal subset of Φ, Φ^{**}, such that $\mathcal{F}(\Phi^{**}) = \mathcal{F}(\Phi)$. We call Φ^{**} a *minimal sufficient subset* of Φ, because Φ^{**} is sufficient to generate exactly the same family of value sets of Φ. Therefore, Φ^{**} and Φ are *semantically-equivalent.* That is, $\Phi^{*} = \Phi - \Phi^{**}$ is redundant in Φ with respect to Φ^{**}. For a set of partial values Φ, the minimal sufficient subset of Φ may not be unique. For example, suppose $\Phi = \{[a], [a, b], [b, c], [a, c]\}$. There are two minimal sufficient subsets of Φ, namely Φ_1^{**} and Φ_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. Eliminating Redundant Partial Values

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

Our approach can be sketched as follows. We start with finding some necessary elements in Φ that correspond to all *minimal elements* (Suppes, 1960) of Φ. 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 Φ. By $\mathcal{F}^{*}(\Phi)$, we derive some useful properties for searching a minimal sufficient subset. Finally, the matching technique in graph theory (Bondy and Murty, 1976) is employed to develop an efficient algorithm to achieve the goal.

3.1 Finding All Minimal Elements of Φ

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

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

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, η_1 is necessary in Φ.

Case 2: $m > 1$. If $\eta_i \not\subset \eta_k, \forall i \neq k$, then we have $\eta_i - \eta_k \neq \emptyset, \forall i \neq k$. Therefore, there exists an interpretation, $\alpha' = (a'_1, a'_2, \ldots, a'_{k-1}, a'_{k+1}, \ldots, a'_m)$, 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 α' is $S_{\alpha'} = \bigcup_{\forall\, i \neq k} \{a'_i\}$, we have $S_{\alpha'} \cap \eta_k = \emptyset$. But, for all interpretations, $\alpha_j = (a_{1j}, a_{2j}, \ldots, a_{kj}, \ldots, a_{mj})$, $1 \leq j \leq | \eta_1 | \times | \eta_2 | \times \cdots \times | \eta_m |$, of Φ, 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, $a_{kj} \in (S_{\alpha_j} \cap \eta_k) \neq \emptyset$, $1 \leq j \leq | \eta_1 | \times | \eta_2 | \times \cdots \times | \eta_m |$, 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. □

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

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

By Lemma 3.1, the following corollaries can be obtained.

Corollary 3.1 Any partial value of cardinality 1 in a set of partial values Φ is a necessary element of Φ.

Proof: Directly from Lemma 3.1. □

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

Proof: Directly from Lemma 3.1. □

Corollary 3.3 For all $\eta_i \in \Phi - \mathcal{M}(\Phi)$, Φ 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 η_x to have the minimum cardinality in Φ, say η_j, such that $\eta_j \subset \eta_i$. That is, there is no element in Φ which is a proper subset of η_j. Therefore, by Lemma 3.1, η_j must be an element of $\mathcal{M}(\Phi)$. This completes the proof. □

Because minimal elements of Φ cannot be eliminated, they are used as a kernel for finding the upper bound of $\mathcal{F}(\Phi)$, $\mathcal{F}^*(\Phi)$. First, we identify $\mathcal{M}(\Phi)$ by applying Lemma 3.1 to Φ. 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 Φ.)
Input: A set of partial values, Φ, which contains no quasi-duplicates.
Output: $\mathcal{M}(\Phi)$.

```
1.  M(Φ) = ∅;
2.  for each η_i ∈ Φ do {
3.      if (| η_i | == 1) then M(Φ) = M(Φ) ∪ {η_i};/*Corollary 3.1*/
4.      else {
5.          minimal = true; /* a flag */
6.          for each η_j ∈ Φ, η_j ≠ η_i, do
7.              if (η_j − η_i = ∅) then {
8.                  minimal = false;
9.                  break; /* exit the inner for loop */
10.             }
11.         if (minimal) then M(Φ) = M(Φ) ∪ {η_i};
12.     }
13. }
14. Output(M(Φ));
```

Recall that $\mathcal{M}(\Phi)$ is defined on the set Φ which contains no quasi-duplicates. In other words, if we want to apply *Find_All_Minimal_Elements* to find a subset of necessary partial values for an arbitrary Φ, we need to eliminate all the quasi-duplicates in Φ 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: η_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: η_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. □

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, Σ, δ), where

Q is a finite set of *states*,
Σ is a finite *input alphabet*, and
δ 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.

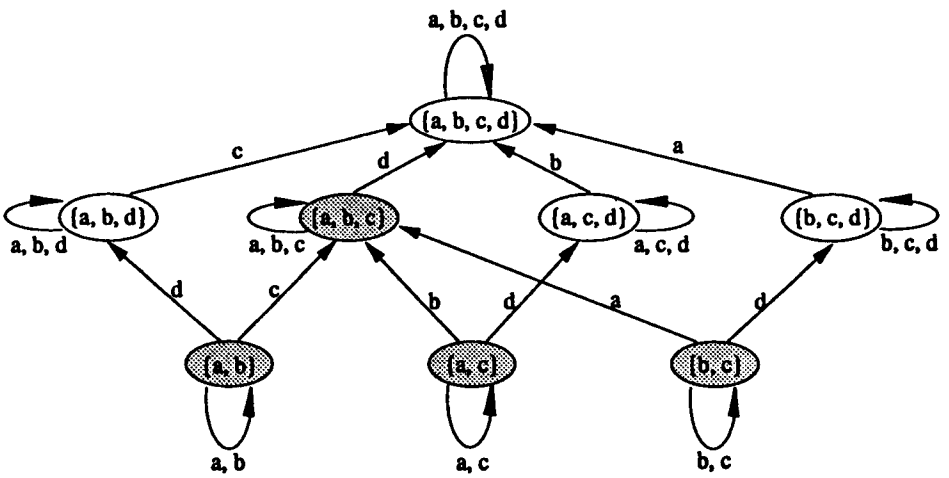
To derive the upper bound of $\mathcal{F}(\Phi)$, we employ a DG (Q, Σ, δ), with $Q = \mathcal{F}^*(\Phi)$, $\Sigma = \bigcup_{\forall \eta_i \in \Phi} \eta_i$; δ defined as $\delta(S_i, a_j) = S_k \in \mathcal{F}^*(\Phi)$, where $S_k = S_i \cup \{a_j\}$, $\forall\, S_i \in \mathcal{F}^*(\Phi)$ and $a_j \in \Sigma$. Initially, we compute $\mathcal{F}(\mathcal{M}(\Phi))$ by Definition 2.5 and then work toward $\mathcal{F}^*(\Phi)$ by applying δ 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, repeating this process will monotonically increase the number of states. However, as Φ and Σ are all finite, there exists a *least fixed point* (Ullman, 1988) such that at that point no more new states can be generated. As a matter of fact, the least fixed point is reached after Σ 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 Φ. 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 /* $\Sigma = \bigcup_{\forall \eta_i \in \Phi} \eta_i$ */
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

Figure 1. The deterministic graph of Example 3.1



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

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

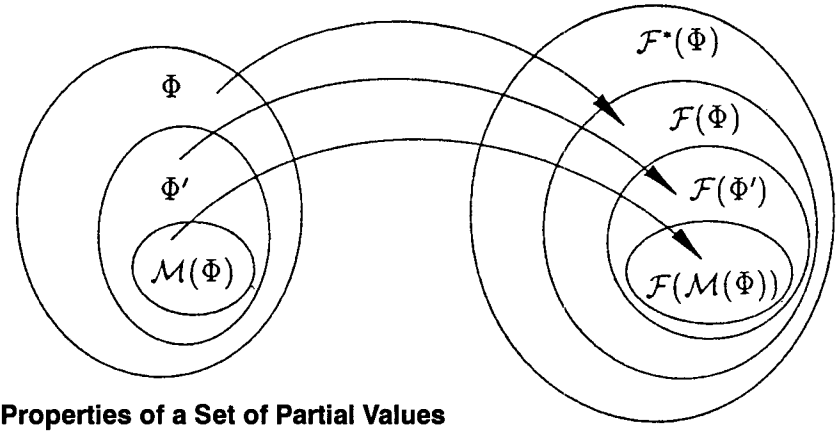
Lemma 3.2 For all Φ', $\mathcal{M}(\Phi) \subseteq \Phi' \subseteq \Phi$, $\mathcal{F}(\Phi') \subseteq \mathcal{F}^*(\Phi)$.

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 $\Sigma' = \cup_{\forall \eta_i \in \Phi'} \eta_i \subseteq \Sigma$. Therefore, by the procedure *Find_Upper_Bound_of* $\mathcal{F}(\Phi)$, for any $S' \in \mathcal{F}(\Phi')$ we obtain $S' \in \mathcal{F}^*(\Phi)$. That completes the proof. □

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

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

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



3.3 Properties of a Set of Partial Values

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

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

Proof: Because $\mathcal{F}(\Phi) \subseteq \mathcal{F}^*(\Phi)$, 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 $\mathcal{M}(\Phi) = \{\eta'_1, \eta'_2, \ldots, \eta'_k\}$ and $\Phi = \{\underbrace{\eta'_1, \eta'_2, \ldots, \eta'_k}_{\mathcal{M}(\Phi)}, \underbrace{\eta_{k+1}, \eta_{k+2}, \ldots, \eta_m}_{\Phi - \mathcal{M}(\Phi)}\}$, where $k = \mid \mathcal{M}(\Phi) \mid$ and $m = \mid \Phi \mid$. By Corollary 3.4, for all η_i, $k+1 \leq i \leq m$, there exists an η'_j, $1 \leq j \leq k$, such that $\eta'_j \subseteq \eta_i$. Therefore, for any interpretation $\alpha' = (a'_1, a'_2, \ldots, a'_k)$ of $\mathcal{M}(\Phi)$, we can find a corresponding interpretation $\alpha = (a'_1, a'_2, \ldots, a'_k, a_{k+1}, a_{k+2}, \ldots, a_m)$ for Φ such that $a_i = a'_j$ if $\eta'_j \subseteq \eta_i$. Then the value set of α' is equal to that of α. That is, for any $S \in \mathcal{F}(\mathcal{M}(\Phi))$, S is also in $\mathcal{F}(\Phi)$. This completes the proof. □

By Lemma 3.3, we conclude the following corollary.

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

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 Φ is replaced by Φ' and $\Phi' = \{\underbrace{\eta'_1, \eta'_2, \ldots, \eta'_k}_{\mathcal{M}(\Phi)}, \underbrace{\eta_{k+1}, \eta_{k+2}, \ldots, \eta_l}_{\Phi' - \mathcal{M}(\Phi)}\}$, where $k = \mid \mathcal{M}(\Phi) \mid$ and $l = \mid \Phi' \mid$.

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 Φ'. □

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

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 $S \notin \mathcal{F}^*(\Phi)$. 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 Σ be the value set of an interpretation $\alpha = (a_1, a_2, \ldots, a_k)$ of $\mathcal{M}(\Phi)$, where $k = \mid \mathcal{M}(\Phi) \mid$, we can obtain another interpretation, $\alpha' = (a'_1, a'_2, \ldots, a'_k)$, of $\mathcal{M}(\Phi)$ by letting

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

Then, the value set of α' is $S \in \mathcal{F}(\mathcal{M}(\Phi))$—a contradiction. Hence, the claim follows. That is, for all the interpretations of $\mathcal{M}(\Phi)$, $\alpha_j = (a_{1j}, a_{2j}, \ldots, a_{ij}, \ldots, a_{kj})$, where $k = \mid \mathcal{M}(\Phi) \mid$, 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 Σ - S. Recall that $\mathcal{F}^*(\Phi)$ is generated from $\mathcal{F}(\mathcal{M}(\Phi))$ by the transition function δ, 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 δ, 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 δ is 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. □

The following theorem provides a more general property for a minimal sufficient subset of Φ.

Theorem 3.2 For all Φ', $\mathcal{M}(\Phi) \subseteq \Phi' \subseteq \Phi$, $\Sigma \in \mathcal{F}(\Phi')$ 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 have to show only 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 $S \notin \mathcal{F}^*(\Phi)$. 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 Φ', $\alpha_j = (a_{1j}, a_{2j}, \ldots, a_{ij}, \ldots, a_{kj})$, where $k = \mid \Phi' \mid$, 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. □

From Theorem 3.1, if $\Sigma \in \mathcal{F}(\mathcal{M}(\Phi))$, then the minimal sufficient subset of Φ, Φ^{**}, is $\mathcal{M}(\Phi)$. Theorem 3.2 provides another property to determine Φ^{**} when $\Sigma \notin \mathcal{F}(\mathcal{M}(\Phi))$; i.e., for a minimal subset of Φ, Φ', where $\mathcal{M}(\Phi) \subset \Phi'$, and $\Sigma \in \mathcal{F}(\Phi')$ then $\Phi^{**} = \Phi'$. Later, we will discuss how to find Φ^{**} 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 (Bondy and Murty, 1976).

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

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

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 Σ, $G = (\Sigma \cup \mathcal{M}(\Phi), E)$, there is a complete matching from Σ to $\mathcal{M}(\Phi)$.

Proof: Let $M = \{(a_1, \eta_1), (a_2, \eta_2), \ldots, (a_s, \eta_s)\}$ be a complete matching from Σ to $\mathcal{M}(\Phi) = \{\eta_1, \eta_2, \ldots, \eta_s, \eta_{s+1}, \ldots, \eta_k\}$ in $G = (\Sigma \cup \mathcal{M}(\Phi), E)$, where $s =$

$| \Sigma |$ and $\Sigma = \{a_1, a_2, \ldots, a_s\}$. Then we have $a_i \in \eta_i, 1 \leq i \leq s$. Therefore, we can find an interpretation of $\mathcal{M}(\Phi)$, $\alpha = (a_1, a_2, \ldots, a_s, a_{s+1}, \ldots, a_k)$, such that its value set $S = \bigcup_{1 \leq i \leq k} \{a_i\} = \Sigma \in \mathcal{F}(\mathcal{M}(\Phi))$.

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

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

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

In some cases, there may not be a complete matching from Σ to Φ in the membership graph $G = (\Sigma \cup \Phi, E)$. That is, $\Sigma \notin \mathcal{F}(\Phi)$. In the following, we explore how to determine Φ^{**} 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 Φ' in the membership graph of Φ' over S, $G = (S \cup \Phi', E)$ or
- there is 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

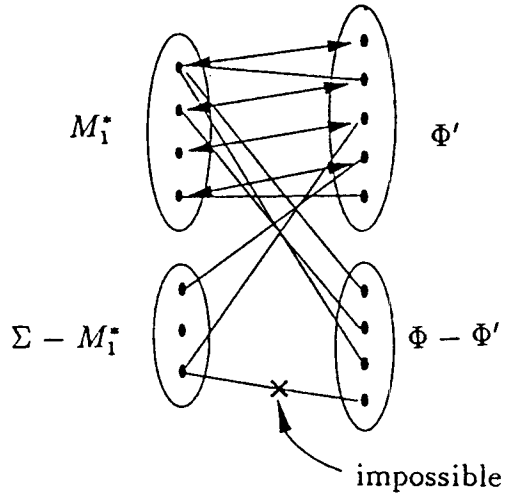
- there is a complete matching from S to Φ' in the membership graph of Φ' over S, $G = (S \cup \Phi', E)$ and
- for all $\eta_i \in \Phi'$, $\eta_i \cap S \neq \emptyset$.

Suppose $M = \{(a_1, \eta_1), (a_2, \eta_2), \ldots, (a_s, \eta_s)\}$, where $s = | S |$ and $S = \{a_1, a_2, \ldots, a_s\}$, 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 $\alpha' = (a_1, a_2, \ldots, a_s, a_{s+1}, \ldots, a_k)$, where $k = | \Phi' |$, of Φ' such that $a_i \in \eta_i \cap S, i = s + 1, \ldots, k$. That is, $S_{\alpha'} = S \in \mathcal{F}(\Phi')$.

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

Hall (1935) gave a necessary and sufficient condition under which there is a complete matching M from X to Y for a bipartite graph $G = (X \cup Y, E)$.

Figure 3. The partitions of Σ and Φ.



Theorem 3.5 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. □

If there is a matching $M = \{(a_1, \eta_1), (a_2, \eta_2), \ldots, (a_s, \eta_s)\}$, 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 Φ^{**} when there is not a complete matching from Σ to Φ.

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 Σ to Φ. Because $\mathcal{M}(\Phi) \subseteq \Phi' \subseteq \Phi$, 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 Σ to Φ. Therefore, according to M^* and Φ', Σ can be partitioned into M_1^* and $\Sigma - M_1^*$ and Φ can be partitioned into Φ' 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, 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 (Figure 3), which violates the condition that M^* is a maximum matching. That is, for all $a_i \in \Sigma - M_1^*$ and $\eta_j \in \Phi - \Phi'$, $a_i \notin \eta_j$.

Because $\mathcal{F}(\Phi') \subseteq \mathcal{F}(\Phi)$, we have to show only $\mathcal{F}(\Phi) \subseteq \mathcal{F}(\Phi')$. That is, for any $S \notin \mathcal{F}(\Phi')$, we want to show that $S \notin \mathcal{F}(\Phi)$. For any $S \notin \mathcal{F}(\Phi')$, we

distinguish three cases as follows. Note that S cannot be M_1^*, for $M_1^* \in \mathcal{F}(\Phi')$.

Case (1): $S \subset M_1^*$. Because 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 is a complete matching $M' \subseteq M^*$ from S to Φ' in the membership graph $G' = (S \cup \Phi', E')$, a subgraph of G^*. Thus, by Lemma 3.4, $S \notin \mathcal{F}(\Phi')$ implies that there must be an $\eta_i \in \Phi'$ such that $\eta_i \cap S = \emptyset$. That is, for all the interpretations of Φ', $\alpha_j = (a_{1j}, a_{2j}, \ldots, a_{ij}, \ldots, a_{kj})$, 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 that 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, $S \notin \mathcal{F}(\Phi)$, because 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 Φ' in the membership graph of Φ' over S, G' = (S $\cup \Phi'$, E') or

(b) There is 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 the *Pigeonhole Principle* (Lewis and Papadimitriou, 1981), 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 $G' = (S \cup \Phi', E')$. Because $S' \not\subseteq M_1^*$, we have either

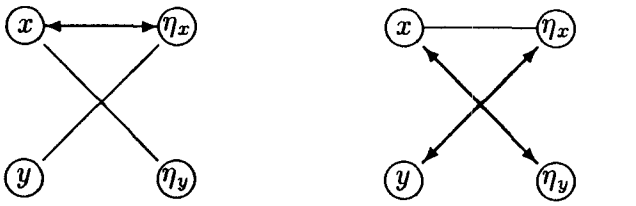
(1) $S'' \subseteq \Sigma - M_1^*$ or

(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 $G'' = (S \cup \Phi, E'')$, because $\Phi - \Phi'$ contains no neighboring vertices of S''. That is, there is no complete matching from S to Φ 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 is more than one element in $S'' \cap M_1^*$ then $|N(S'')| > 1$, which violates $| N(S'') | = 1$. Therefore, $(x, \eta_x) \in M^*$. 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 rti 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 (Figure 4). This contradicts the assumption that M^* is a maximum matching in G.

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



$\{(x, \eta_x)\} \subseteq M^*$ $\{(x, \eta_y), (y, \eta_x)\} \subseteq M^{**}$

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

4. Finding a Minimal Sufficient Subset

Based on the properties discussed above, we develop an efficient algorithm to derive Φ^{**} 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 (1973) developed an $O(n^{5/2})$ algorithm for finding a maximum matching in a bipartite graph, where n is the number of vertices. Due to this algorithm, Papadimitriou and Steiglitz (1982) relate this problem to the max-flow problem (Ford and Fulkerson, 1962) 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 that which is empty), this algorithm gradually augments the matching process until no augmentation can be obtained. Thus, the resultant matching becomes maximum.

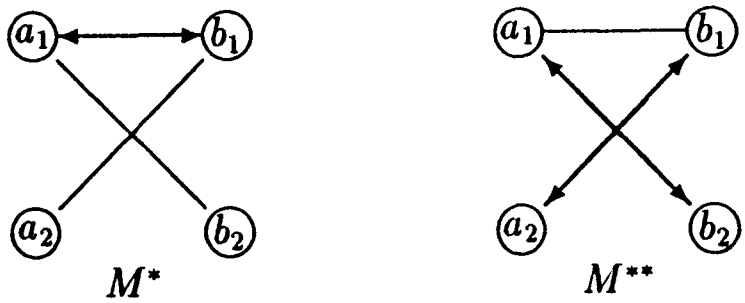
By giving an initial matching, this matching algorithm will be used as a procedure in our algorithm as follows. Notice that a *complete matching* in a bipartite graph G is also a *maximum matching* in G.

Algorithm 4.1 An Algorithm That Derives a Minimum Sufficient Subset of Φ, Φ^{**}.

Input: A Set of Partial Values, Φ.
Output: A Minimum Sufficient Subset of Φ, Φ^{**}.

1. $\Sigma = \bigcup_{\forall \eta_i \in \Phi} \eta_i$;
2. Eliminate all quasi-duplicates of Φ and denote the resultant set Φ';
3. Call *Find_All_Minimal_Elements*(Φ'), which returns $\mathcal{M}(\Phi')$;
4. Find a maximum matching M^* in membership graph G =

Figure 5. Relationship between M^* and $M.^*$*



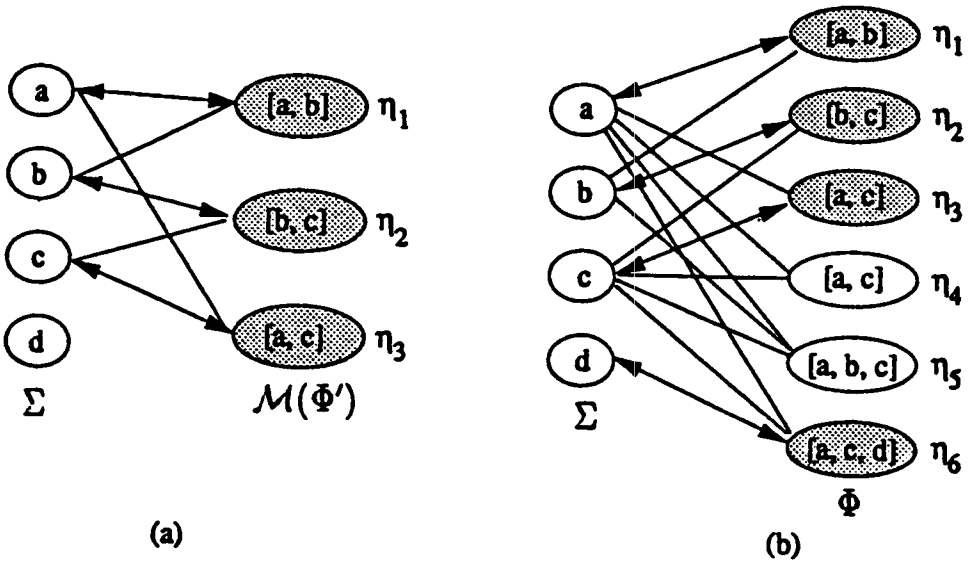
$(\Sigma \cup \mathcal{M}(\Phi'), E)$
by giving an empty matching as the initial matching;

5. If ($| M^* | == | \Sigma |$) then { /* Theorems 3.1 and 3.3 */
6. $\Phi^{**} = \mathcal{M}(\Phi')$;
7. Output(Φ^{**}); Stop;
8. } else {
9. Find a maximum matching M^{**} in membership graph $G^* = (\Sigma \cup \Phi, E)$
by giving M^* as the initial matching to ensure the minimality;
10. $\Phi^{**} = \mathcal{M}(\Phi') \cup M_2^{**}$; /* Theorem 3.6 */
11. Output(Φ^{**}); Stop;
12. }

Note that to ensure Φ^{**} to be minimal, M^* must be given as the initial matching when finding M^{**} in Step 9. That ensures $M_2^* \subseteq M_2^{**}$. Notice that M^* is not necessarily a subset of M^{**}. For example, in Figure 5, $M^* = \{(a_1, b_1)\}$ and $M^{**} = \{(a_1, b_2), (a_2, b_1)\}$. $M^* \not\subseteq M^{**}$ but $M_2^* \subseteq M_2^{**}$. In the following, we 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 Φ^{**} 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 is illustrated by Figure 6(a). The shaded nodes in Figure 6(a) are elements of M_2^*. Because the cardinalities of

Figure 6(a). Maximum matching M^* (b) Maximum matching M^{**}.



M^* and Σ 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 6(b) depicts this. The shaded nodes are elements of M_2^{**}.
Therefore,

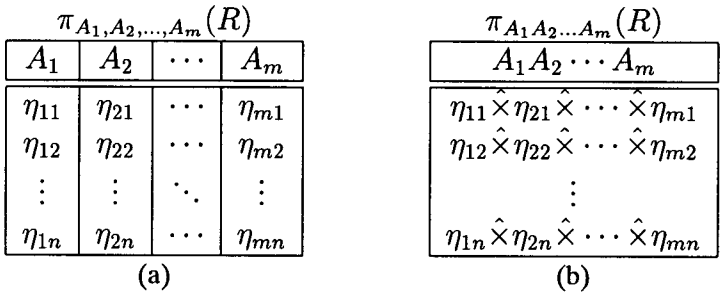
$$\Phi^{**} = \mathcal{M}(\Phi') \cup M_2^{**} = \{\overbrace{[a, b]}^{\eta_1}, \overbrace{[b, c]}^{\eta_2}, \overbrace{[a, c]}^{\eta_3}, \overbrace{[a, c, d]}^{\eta_6}\}.$$

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

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

The procedure *Find_All_Minimal_Elements* takes $O(n^2)$, where $n = \mid \Phi' \mid$, 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 $O(\mid V \mid^{1/2} \cdot \mid E \mid)$, where $\mid V \mid = \mid \Sigma \mid + \mid \Phi \mid$ and $\mid E \mid = \sum_{\forall \eta_i \in \Phi} \mid \eta_i \mid$. In the worst case, this complexity is $O(n^{5/2})$, where $n = \mid V \mid$. Note that in this algorithm we do not need to generate $\mathcal{F}^*(\Phi)$ by *Find_Upper_Bound_of_*$\mathcal{F}(\Phi)$.

Figure 7. Two equivalent relations



5. 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 $\eta_a = [a_1, a_2, \ldots, a_m]$ and $\eta_b = [b_1, b_2, \ldots, b_n]$ 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, $\eta_{a\hat{\times}b} = [(a, x), (a, y), (b, x), (b, y), (c, x), (c, y)]$. □

Consider the result of a projection $\pi_{A_1,A_2,\ldots,A_m}(R), m > 1$, as Figure 7(a) depicts. The relation $\pi_{A_1,A_2,\ldots,A_m}(R)$ then can be regarded as a relation $\pi_{A_1A_2\ldots A_m}(R)$ with the single attribute $A_1 A_2 \ldots A_m$. If the "true" value of a tuple of $\pi_{A_1,A_2,\ldots,A_m}(R)$, $\boxed{\eta_{1j}, \eta_{2j}, \ldots, \eta_{mj}}$, is the m-tuple $(a_1, a_2, \ldots, a_m)$, $a_i \in \eta_{ij}$, where a_i is the "true" value of η_{ij}; then the "true" value of $\eta_{1j\hat{\times}2j\hat{\times}\cdots\hat{\times}mj}$ is also $(a_1, a_2, \ldots, a_m)$, and vice versa. We know that a tuple of $\pi_{A_1,A_2,\ldots,A_m}(R)$, $\boxed{\eta_{1j}, \eta_{2j}, \ldots, \eta_{mj}}$, can be considered as a tuple of $\pi_{A_1A_2\ldots A_m}(R)$ with attribute value $\eta_{1j\hat{\times}2j\hat{\times}\cdots\hat{\times}mj} \equiv \eta_{1j}\hat{\times}\eta_{2j}\hat{\times}\ldots\hat{\times}\eta_{mj}$. That is, the relations $\pi_{A_1,A_2,\ldots,A_m}(R)$ and $\pi_{A_1A_2\ldots A_m}(R)$ are semantically equivalent and can be transformed to each other. Figure 7 illustrates this. By this transformation, a one-attribute relation can always be obtained and Algorithm 4.1 works as before.

6. Conclusion

Partial values have been used to represent imprecise data in databases. In previous work we studied extended algebraic operations on partial values (Tseng et al., 1993*b*,

1993*c*). In this article, 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.

Using 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 Φ, and derive properties for a set of partial values. In addition, the problem of searching a minimal sufficient subset of Φ, Φ^{**}, is converted into a bipartite graph matching problem. Based on the properties of partial values, we develop an efficient algorithm to find Φ^{**} 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.

For the *union* of two sets of partial values, Φ_1 and Φ_2, our work can be employed as follows. First, collect together all members of Φ_1 and Φ_2 to form another set Φ. Then, apply our work to eliminate redundant elements in Φ. Imieliński and Vadaparty (1989) and Imieliński (1991) pointed out that if partial values are allowed to occur in databases, the data complexity of query processing jumps from PTIME to CoNP (Garey and Johnson, 1979). However, there are also some types of queries that have PTIME complexity. Our ongoing studies of query processing over partial values are intended to discover more PTIME algorithms from algebraic point of view. In our recent work (Tseng et al., 1993*b*, 1993*c*), we found that *division* (by restricting the divisor to be definite) and some aggregate operations over partial values—*min*, *max*, and *count*—can be done in PTIME.

Acknowledgments

This research was partially supported by the Republic of China National Science Council under Contract No. NSC 81-0408-E-007-12. The authors wish to thank the anonymous referees whose invaluable comments and suggestions helped to improve this paper substantially.

References

Abiteboul, S. and Grahne, G. Update semantics for incomplete information, *Proceedings of the Eleventh International Conference on Very Large Data Bases,* Stockholm, 1985.

Bancilhon, F. and Spyratos, N. Update semantics of relational views, *ACM Transmissions of Database Systems*, 6(4):557-575, 1981.

Biskup, J. A foundation of Codd's relational maybe-operations, *ACM Transmissions of Database Systems*, 8(4):608-636, 1983.

Bondy, J.A. and Murty, U.S.R. *Graph Theory with Applications,* New York: Macmillan Press, 1976.

Codd, E.F. Extending the database relational model to capture more meaning, *ACM Transmissions of Database Systems*, 4(4):397-434, 1979.

Codd, E.F. Missing information (applicable and inapplicable) in relational databases, *SIGMOD Record*, 15(4):53-78, 1986.

Codd, E.F. More commentary on missing information in relational databases (applicable and inapplicable information), *SIGMOD Record*, 16(1):42-50, 1987.

DeMichiel, L.G. Resolving database incompatibility: An approach to performing relational operations over mismatched domains, *IEEE Transmissions on Knowledge and Data Engineering*, 1(4):485-493, 1989.

Ford, L.R. and Fulkerson, D.R. *Flows in Networks*, Princeton, NJ: Princeton University Press, 1962.

Garey, M.R. and Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco: Freeman, 1979.

Grant, J. Null values in a relational data base, *Information Processing Letters*, 6(5):156-157, 1977.

Grant, J. Partial values in a tabular database model, *Information Processing Letters*, 9(2):97-99, 1979.

Hall, P. On representatives of subsets, *J. London Mathematical Society*, 10(26-30), 1935.

Hopcroft, J.E. and Karp, R.M. An $n^{5/2}$ algorithm for maximum matching in bipartite graphs, *SIAM J. Computing*, 2(4):225-231, 1973.

Imieliński, T. and Lipski, W. On representing incomplete information in a relational database. *Proceedings of the Seventh International Conference on Very Large Data Bases*, Cannes, France, 1981.

Imieliński, T. and Lipski, W. Incomplete information and dependencies in relational databases, *Proceedings of the ACM SIGMOD International Conference Management of Data*, San Jose, California, 1983.

Imieliński, T. and Vadaparty, K. Complexity of query processing in databases with or-objects, *Proceedings of the ACM Symposium on Principles of Database Systems,* 1989.

Imieliński, T. Incomplete Deductive Databases. *Annals of Mathematics and Artificial Intelligence,* 3(2-4):259-294, 1991.

Lewis, H. and Papadimitriou, C. *Elements of the Theory of Computation*, Englewood Cliffs, NJ: Prentice-Hall, 1981, pp.26-26.

Lien, E. Multivalued dependencies with null values in relational databases. *Proceedings of the Fifth International Conference on Very Large Data Bases*, Rio de Janeiro, 1979.

Lipski, W. On semantic issues connected with incomplete information systems. *ACM Transmissions on Database Systems*, 4(3):262-296, 1979.

Liu, K.-C. and Sunderraman, R. Indefinite and maybe information in relational databases. *ACM Transmissions on Database Systems*, 15(1):1-39, 1990.

Liu, K.-C. and Sunderraman, R. A generalized relational model for indefinite and maybe information. *IEEE Transmissions on Knowledge and Data Engineering*, 3(1):65-77, 1991.

Maier, D. *The Theory of Relational Databases*, Rockville, MD: Computer Science Press, 1983.

Motro, A. Accommodating imprecision in database systems: issues and solutions. *ACM SIGMOD Record*, 19(4):69-74, 1990.

Papadimitriou, C.H. and Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*, Englewood Cliffs, NJ: Prentice-Hall, 1982, pp.221-226.

Suppes, P. *Axiomatic Set Theory*, Princeton, New Jersey: D. Van Nostrand Company, 1960, pp.99-100.

Tsai, P.S.M. and Chen, A.L.P. Querying uncertain data in heterogeneous databases. *Proceedings of the IEEE International Workshop on Research Issues on Data Engineering (RIDE)*, Vienna, 1993.

Tseng, F.S.C., Chen, A.L.P., and Yang, W.P. Answering heterogeneous database queries with degrees of uncertainty. *Distributed and Parallel Databases: An International Journal*, 1(3):281-302, 1993*a*.

Tseng, F.S.C., Chen, A.L.P., and Yang, W.P. Implementing the division operation on a database containing uncertain data. Submitted, 1993*b*.

Tseng, F.S.C., Chen, A.L.P., and Yang, W.P. Evaluating aggregate operators over imprecise data. Submitted, 1993*c*.

Ullman, J.D. *Principles of Database and Knowledge-Base Systems*, Vol. 2, Rockville, MD: Computer Science Press, 1988.

Vassiliou, Y. Null values in data base management: A denotational semantics approach. *Proceedings of the ACM-SIGMOD International Conference on the Management of Data*, Boston, MA, 1979.

Vassiliou, Y. Functional dependencies and incomplete information. *Proceedings of the Sixth International Conference on Very Large Data Bases*, Montreal, 1980.