On the Implication Problem for Probabilistic Conditional Independency

Shik Kam Michael Wong, Cory James Butz, Dan Wu

Technical Report CS-99-03
September, 1999

Department of Computer Science
University of Regina
Regina, Saskatchewan, CANADA
S4S 0A2

ISSN 0828-3494
ISBN 0-7731-0390-2

On the Implication Problem for Probabilistic Conditional Independency

S.K.M. Wong, C.J. Butz and D. Wu
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada, S4S 0A2
e-mail: {wong,butz,danwu}@cs.uregina.ca
fax: (306)585-4745

Abstract

The *implication problem* is to test whether a given set of independencies logically implies another independency. This problem is *crucial* in the design of a probabilistic reasoning system. We advocate that Bayesian networks are a generalization of standard relational databases. On the contrary, it has been suggested that Bayesian networks are *different* from the relational databases because the implication problem of these two systems does not coincide for *some* classes of probabilistic independencies. This remark, however, does not take into consideration one important issue, namely, the *solvability* of the implication problem.

In this comprehensive study of the implication problem for probabilistic conditional independencies, it is found that Bayesian networks and relational databases coincide on *solvable* classes of independencies. The present study suggests that the implication problem for these two closely related systems differs only in *unsolvable* classes of independencies. This means there is no *real* difference between Bayesian networks and relational databases, in the sense that only *solvable* classes of independencies are useful in the design and implementation of these knowledge systems. More importantly, perhaps, these results suggest that many current attempts to *generalize* Bayesian networks can take full advantage of the generalizations made to standard relational databases.

1 Introduction

Probability theory provides a rigorous foundation for the management of uncertain knowledge [17, 28, 31]. In this approach, it is assumed that knowledge can be represented as a joint probability distribution. The probability of an event can be obtained (in principle) by an appropriate marginalization of the joint distribution. Obviously, it is impractical to obtain the joint distribution directly: for example, one would have to specify 2^n entries for a distribution over n binary variables. *Bayesian networks* [31] provide a semantic modeling tool which greatly facilitate the acquisition of probabilistic knowledge. A Bayesian network

consists of a *directed acyclic graph* (DAG) and a corresponding set of conditional probability distributions. The DAG encodes all the probabilistic conditional independencies satisfied by a particular joint distribution. The set of conditional independencies that can be inferred from a DAG is called *conflict free*. Every independency logically implied by a conflict-free set of conditional independencies can be inferred from the given DAG. In other words, a DAG is a *perfect-map* [31] of the conflict-free set of independencies. It is important to realize that conflict-free sets are a special class within the general class of probabilistic conditional independency. This special class of independencies is important, since it allows a human expert to indirectly specify a joint distribution as a product of conditional probability distributions. To facilitate the computation of marginal distributions, it is useful in practice to transform a Bayesian network into a (decomposable) *Markov network* [17] by sacrificing all the *embedded* independency information. In fact, a Markov network is defined only by a subclass *nonembedded* independencies.

Before Bayesian networks was proposed, the *relational database model* [10, 23] already established itself as the basis for designing and implementing database systems. Data dependencies [1], such as *embedded multivalued dependency* (EMVD), (nonembedded) *multivalued dependency* (MVD) and *join dependency* (JD), are used to provide an economical representation of a universal relation. As in the study of Bayesian networks, two of the most important results are the ability to specify the universal relation as a *lossless* join of several smaller relations, and the development of efficient methods to only access the relevant portions of the database in query processing. A culminating result [4] is that *acyclic join dependency* (AJD) provides a basis for schema design as it possesses many desirable properties in database applications.

Several researchers including [14, 22, 25, 39] have noticed similarities between relational databases and Bayesian networks. However, we advocate that a Bayesian network is indeed a generalized relational database. Our *unified* approach [41, 44] is to express the concepts used in Bayesian networks by generalizing the familiar relational database terminology. This *probabilistic* relational database model, called the *Bayesian database model*, demonstrates that there is a direct correspondence between the operations and dependencies (independencies) used in these two knowledge systems. More specifically, a joint probability distribution can be viewed as a probabilistic (generalized) *relation*. The *projection* and *natural join* operations in relational databases are special cases of the *marginalization* and *multiplication* operations. Embedded multivalued dependency (EMVD) in the relational database model is a special case of probabilistic conditional independency in the Bayesian database model. More importantly, a Markov network is in fact a generalization of an acyclic join dependency.

In the design and implementation of probabilistic reasoning or database systems, a *crucial* issue to consider is the implication problem. The *implication problem* has been extensively studied in both relational databases, including [2, 3, 24, 26, 27], and in Bayesian networks [14, 15, 16, 30, 33, 36, 37, 40, 45]. The implication problem is to test whether a given input set $\sum$ of independencies logically implies another independency σ. We say $\sum$ *logically implies* σ and write $\sum \models \sigma$, if whenever any distribution (relation) that satisfies all the independencies

[1] Constraints are traditionally called *dependencies* in relational databases, but are referred to as *independencies* in Bayesian networks. Henceforth, we will use the terms *dependency* and *independency* interchangeably.

in $\sum$, then the distribution also satisfies σ. That is, there is no counter-example distribution such that $\sum$ is satisfied while σ is not. Traditionally, *axiomatization* was studied in an attempt to solve the implication problem for probabilistic conditional independencies. In this approach, a finite set of inference axioms are used to generate symbolic proofs for a particular probabilistic conditional independency in a manner analogous to the proof procedures in mathematical logics.

In this paper, we use our unified terminology to present a comprehensive study of the implication problem for probabilistic conditional independencies. In particular, we examine four classes of independencies in the Bayesian database model, namely:

(**1a**) BEMVD,

(**1b**) Conflict-free BEMVD,

(**2a**) BMVD,

(**2b**) Conflict-free BMVD.

Class (**1a**) is the *general* class of probabilistic conditional independencies called *Bayesian embedded multivalued dependency* (BEMVD) in our approach. Classes (**1b**), (**2a**) and (**2b**) are *special* classes of (**1a**). Dependencies in class (**1b**) are called *conflict-free BEMVDs* [2] which can be *faithfully* represented by a *single* DAG. This subclass of dependencies is used to construct a Bayesian network. Dependencies in class (**2a**) are called (*nonembedded*) *Bayesian multivalued dependency* (BMVD). *Nonembedded* probabilistic dependencies, called *fixed context* [14] or *full* [26], are those involving *all* variables. Dependencies in class (**2b**) are called *conflict-free BEMVDs*. In fact, class (**2b**) is a subclass of (**2a**). A set of conflict-free BMVDs is used to construct a Markov network. This class of dependencies can be *faithfully* represented by a single *acyclic hypergraph* [4, 6].

Let $\mathbf{C}$ denote an arbitrary set of probabilistic *dependencies* (see Footnote 1) belonging to one of the above four classes, and $\mathbf{c}$ denote a singleton set from the same class. We desire a means to test whether $\mathbf{C}$ logically implies $\mathbf{c}$, namely:

$$\mathbf{C} \models \mathbf{c}. \tag{1}$$

In our approach, for any arbitrary sets $\mathbf{C}$ and $\mathbf{c}$ of *probabilistic* dependencies, there are *corresponding* sets C and c of *data* dependencies. More specifically, for each of the above four classes of probabilistic dependencies, there is a corresponding class of data dependencies in the relational database model:

[2] A *causal input list* [32] (a *stratified protocol* [38]) is a *minimum cover* [23] of a conflict-free set of BEMVDs.

(1a) EMVD,

(1b) Conflict-free EMVD,

(2a) MVD,

(2b) Conflict-free MVD,

as depicted in Figure 1. Since we advocate that the Bayesian network model is a generalization of the relational database model, an immediate question to answer is:

Do the implication problems coincide in these two database models?

That is, we would like to know whether the proposition:

$$\mathbf{C} \models \mathbf{c} \iff C \models c, \tag{2}$$

holds for the following pairs (**1a**, 1a), (**1b**, 1b), (**2a**, 2a), and (**2b**, 2b). For example, we would like to know whether Proposition (2) holds for the pair (BEMVD, EMVD), where $\mathbf{C}$ is an arbitrary set of BEMVDs, $\mathbf{c}$ is a singleton set of BEMVDs, and C and c are the *corresponding* sets of EMVDs.

Proposition (2) is true for the pair (BMVD, BMVD). That is,

$$\{ \text{ BMVDs } \} \models \mathbf{c} \iff \{ \text{ MVDs } \} \models c.$$

Since the classes (**2b**) and (2b) are special cases with the classes (**2a**) and (2a), respectively, Proposition (2) is obviously true for the pair (conflict-free BMVD, conflict-free MVD):

$$\{ \text{ conflict-free BMVDs } \} \models \mathbf{c} \iff \{ \text{ conflict-free MVDs } \} \models c.$$

It is also true for the pair (conflict-free BEMVD, conflict-free EMVD), namely:

$$\{ \text{ conflict-free BEMVDs } \} \models \mathbf{c} \iff \{ \text{ conflict-free EMVDs } \} \models c.$$

However, it is important to note that Proposition (2) is *not* true for the pair (BEMVD, EMVD). That is, the implication problem does not coincide for the general classes of probabilistic conditional independencies and embedded multivalued dependency. In [37], it was pointed out that:

$$\{ \text{ BEMVDs } \} \models \mathbf{c} \not\Longleftarrow \{ \text{ EMVDs } \} \models c, \tag{3}$$

and

$$\{ \text{ BEMVDs } \} \models \mathbf{c} \not\Longrightarrow \{ \text{ EMVDs } \} \models c. \tag{4}$$

(A solid arrow in Figure 1 represents the fact that Proposition (2) holds, while a dashed arrow indicates that Proposition (2) does not hold.) For this reason, it was suggested in [37] that Bayesian networks are intrinsically *different* from relational databases. This remark,

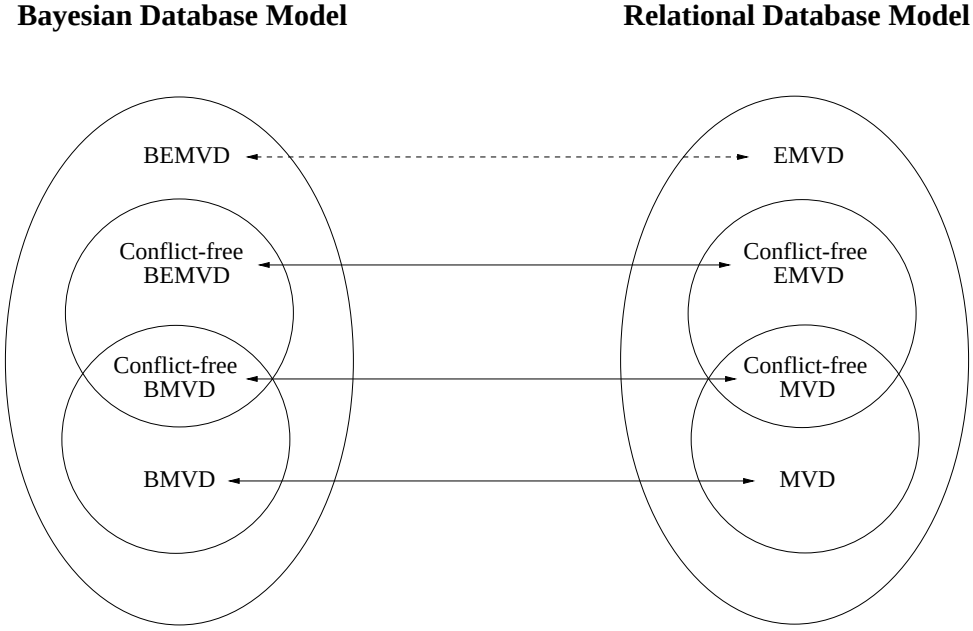

Figure 1: The four classes of *probabilistic* dependencies (BEMVD, conflict-free BEMVD, BMVD, conflict-free BMVD) traditionally found in the Bayesian database model are depicted on the left. The corresponding class of *data* dependencies (EMVD, conflict-free EMVD, MVD, conflict-free MVD) in the standard relational database model are depicted on the right.

however, does not take into consideration one important issue, namely, the *solvability* of the implication problem for a particular class of dependencies.

The question naturally arises as to why the implication problem coincides for some classes of dependencies but not for others. One important result in relational databases is that the implication problem for the general class of EMVDs is *unsolvable* [18]. (By solvability, we mean there exists a method to decide whether $\sum \models \sigma$ holds for an arbitrary instance of the implication problem.) Therefore, the observation in Equation (3) is not too surprising, since EMVD is an *unsolvable* class of dependencies. Furthermore, the implication problem for the BEMVD class of probabilistic conditional independencies is also *unsolvable*. One immediate consequence of our result is the observation in Equation (4). Therefore, the fact that the implication problem in Bayesian networks and relational databases does not coincide is based on *unsolvable* classes of data dependencies. This supports our argument that there is no *real* difference between Bayesian networks and standard relational databases in a practical sense, since only *solvable* classes of dependencies are useful in the design and implementation of both knowledge systems.

This paper is organized as follows. Section 2 contains background knowledge including the traditional relational database model, and our Bayesian relational model. In Section 3, we introduce the basic notions pertaining to the implication problem. In Section 4, we present an in-depth analysis of the implication problem for the BMVD class of *nonembedded*

probabilistic conditional independencies. In particular, we present the *chase* algorithm as a *nonaxiomatic* method for testing the implication of this special class of independencies. In Section 5, we examine the implication problem for *embedded* dependencies. The conclusion is presented in Section 6, in which we emphasize that Bayesian networks are indeed a general form of relational databases.

2 Background Knowledge

In this section, we review pertinent notions including acyclic hypergraphs, the standard relational database model, Bayesian networks, and our Bayesian relational model.

2.1 Acyclic Hypergraphs and Jointrees

In this subsection, we review two graphical structures, acyclic hypergraph and jointree. Dependencies (independencies) can be conveniently characterized by these graphical structures.

Let $R = \{A_1, A_2, \ldots, A_m\}$ be a finite set of attributes. A *hypergraph* $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$ is a set of subsets $R_i \subseteq R$, namely, $\mathcal{R} \subseteq 2^R$. We say that $\mathcal{R}$ has the *running intersection property* if there is a hypertree construction ordering $R_1, R_2, \ldots, R_n$ of $\mathcal{R}$ such that there exists a branching function $b(i) < i$ such that $R_i \cap (R_1 \cup R_2 \cup \ldots \cup R_{i-1}) \subseteq R_{b(i)}$, for $i = 2, 3, \ldots, n$. We call $\mathcal{R}$ an *acyclic hypergraph* if and only if $\mathcal{R}$ has the running intersection property [4]. Given an ordering $R_1, R_2, \ldots, R_n$ for an acyclic hypergraph $\mathcal{R}$ and a branching function $b(i)$ for this ordering, the set $\mathcal{J}$ of *J-keys* for $\mathcal{R}$ is defined as:

$$\mathcal{J} = \{R_2 \cap R_{b(2)}, R_3 \cap R_{b(3)}, \ldots, R_n \cap R_{b(n)}\}. \tag{5}$$

These J-keys are in fact independent of a particular hypertree construction ordering, an acyclic hypergraph has a unique set of J-keys.

Example 1 Let $R = \{A_1, A_2, A_3, A_4, A_5, A_6\}$ and $\mathcal{R} = \{ R_1 = \{A_1, A_2, A_3\}, R_2 = \{A_2, A_3, A_4\}, R_3 = \{A_2, A_3, A_5\}, R_4 = \{A_5, A_6\} \}$ denote the hypergraph illustrated in Figure 2. It can be easily verified that $\mathcal{R}$ has the following hypertree construction ordering:

$$\begin{array}{rcccll} R_2 \cap R_1 & = & \{A_2, A_3\} & \subseteq & R_1; & b(2) = 1, \\ R_3 \cap (R_1 \cup R_2) & = & \{A_2, A_3\} & \subseteq & R_1; & b(3) = 1, \\ R_4 \cap (R_1 \cup R_2 \cup R_3) & = & \{A_5\} & \subseteq & R_3; & b(4) = 3. \end{array}$$

Thus, $\mathcal{R}$ is an acyclic hypergraph. The set $\mathcal{J}$ of J-keys for this acyclic hypergraph $\mathcal{R}$ is

$$\mathcal{J} = \{R_2 \cap R_1, R_3 \cap R_1, R_4 \cap R_3\} = \{\{A_2, A_3\}, \{A_5\}\}. \ \square$$

In the probabilistic reasoning literature [17, 31], the graphical structure of a probabilistic network is usually a jointree. However, it is important to realize that saying that $\mathcal{R}$ is an acyclic hypergraph is the same as saying that $\mathcal{R}$ has a jointree. Given an acyclic hypergraph $\mathcal{R}$ and a branching function b, a *jointree* for $\mathcal{R}$ is a tree with set $\mathcal{R}$ of nodes, such that:

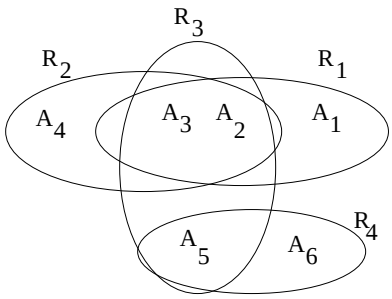

Figure 2: A graphical representation of the acyclic hypergraph $\mathcal{R} = \{ R_1, R_2, R_3, R_4 \}$.

(i) Each edge $(R_i, R_{b(i)})$ is labeled by the set of attributes $R_i \cap R_{b(i)}$, and

(ii) For every pair R_i, R_j $(R_i \neq R_j)$ and every A in $R_i \cap R_j$, each edge along the unique path between R_i and R_j includes A.

Example 2 Consider the acyclic hypergraph $\mathcal{R}$ in Figure 2, where R_1, R_2, R_3, R_4 is a hypertree construction ordering with branching function $b(2) = 1$, $b(3) = 1$ and $b(4) = 3$. A jointree for this $\mathcal{R}$ is shown in Figure 3. □

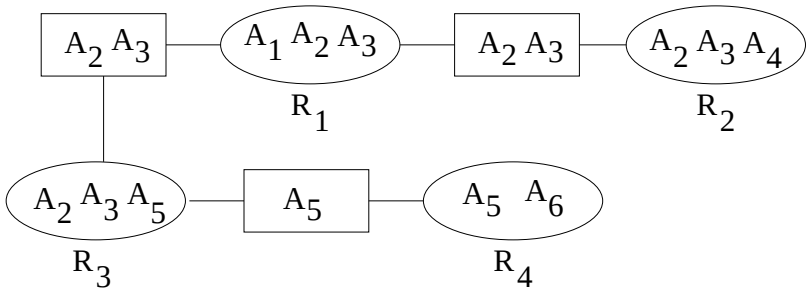

Figure 3: A graphical representation of a jointree of the acyclic hypergraph $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$ in Figure 2.

2.2 Relational Databases

To clarify the notions used, we give a brief review of the standard relational database model [23]. The concepts presented here are generalized in the next section to express corresponding notions in Bayesian networks.

A *relation scheme* $R = \{A_1, A_2, \ldots, A_m\}$ is a finite set of *attributes* (attribute names). Corresponding to each attribute A_i is a nonempty finite set D_{A_i}, $1 \leq i \leq m$, called the *domain* of A_i. Let $D = D_{A_1} \cup D_{A_2} \ldots \cup D_{A_m}$. A *relation* r on the relation scheme R, written $r(R)$, is a finite set of mappings $\{t_1, t_2, \ldots, t_s\}$ from R to D with the restriction that for each mapping $t \in r$, $t(A_i)$ must be in D_{A_i}, $1 \leq i \leq m$, where $t(A_i)$ denotes the value obtained by restricting the mapping to A_i. An example of a relation r on R in general is shown in

r =

A_1	A_2	...	A_m
$t_1(A_1)$	$t_1(A_2)$	...	$t_1(A_m)$
$t_2(A_1)$	$t_2(A_2)$	...	$t_2(A_m)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t_s(A_1)$	$t_s(A_2)$	...	$t_s(A_m)$

Figure 4: A relation r on the scheme $R = \{A_1, A_2, \ldots, A_m\}$.

$r(ABC)$ =

A	B	C
0	0	0
0	0	1
1	0	0

Figure 5: A relation r on the scheme $R = ABC$.

Figure 4. The mappings are called *tuples* and $t(A)$ is called the A-value of t. We use $t(X)$ in the obvious way and call it the X-value of the tuple t.

Mappings are used in our exposition to avoid any explicit ordering of the attributes in the relation scheme. To simplify the notation, however, we will henceforth denote relations by writing the attributes in a certain order and the tuples as lists of values in the same order. The following relational database conventions will be adopted. Uppercase letters A, B, C from the beginning of the alphabet will be used to denote attributes. A relation scheme $R = \{A_1, A_2, \ldots, A_m\}$ is written as simply $A_1A_2 \ldots A_m$. A relation r on scheme R is denoted by either $r(R)$ or $r(A_1A_2 \ldots A_m)$. The singleton set $\{A\}$ is written as A and the concatenation XY is used to denote set union $X \cup Y$. For example, a relation $r(R)$ on $R = ABC$ is shown in Figure 5, where $D_A = D_B = D_C = \{0, 1\}$.

Let r be a relation on R and X a subset of R. The *projection of* r *onto* X, written $\pi_X(r)$, is defined as:

$$\pi_X(r) = \{\ t(X) \mid t \in r\ \}. \tag{6}$$

That is, $\pi_X(r)$ is the set of all tuples $t(X)$ such that t is in r.

The *natural join* of two relations $r_1(X)$ and $r_2(Y)$, written $r_1(X) \bowtie r_2(Y)$, is defined as:

$$r_1(X) \bowtie r_2(Y) = \{\ t(XY) \mid t(X) \in r_1(X) \text{ and } t(Y) \in r_2(Y)\ \}. \tag{7}$$

That is, $r_1(X) \bowtie r_2(Y)$ denotes the set of tuples $t(XY)$ such that $t(X)$ is in r_1 and $t(Y)$ is in r_2.

Let $r_1(R_1)$, $r_2(R_2)$,...,$r_n(R_n)$ be relations and $R = R_1 \cup R_2 \cup \ldots \cup R_n$. Let $t_1, t_2, \ldots, t_n$ be a sequence of tuples (not necessarily distinct) with $t_i \in r_i$, $1 \leq i \leq n$. We say tuples $t_1, t_2, \ldots, t_n$ are *joinable on* $R_1, R_2, \ldots, R_n$ if there is a tuple t on R such that $t_i = t(R)$, $1 \leq i \leq n$. Tuple t is the *result* of joining $t_1, t_2, \ldots, t_n$ on $R_1, R_2, \ldots, R_n$.

Let R be a relation scheme, X and Y be subsets of R, and $Z = R - XY$. A relation $r(R)$ satisfies the *multivalued dependency* (MVD) $X \rightarrow\rightarrow Y$ if, for any two tuples t_1 and t_2 in r with $t_1(X) = t_2(X)$, there exists a tuple t_3 in r with:

$$t_3(XY) = t_1(XY) \text{ and } t_3(Z) = t_2(Z). \tag{8}$$

It can be shown that:

$$X \rightarrow\rightarrow Y \iff X \rightarrow\rightarrow Y - X.$$

The MVD $X \rightarrow\rightarrow Y$ is a *necessary* and *sufficient* condition for $r(R)$ to be losslessly decomposed, namely:

$$r(R) = \pi_{XY}(r) \bowtie \pi_{XZ}(r). \tag{9}$$

Example 3 The relation $r(ABC)$ in Figure 6 satisfies the MVD $B \rightarrow\rightarrow A$, since $r(ABC) = \pi_{AB}(r) \bowtie \pi_{BC}(r)$. On the other hand, the relation $r'(ABC)$ in Figure 7 does not satisfy the MVD $B \rightarrow\rightarrow A$, since $r'(ABC) \neq \pi_{AB}(r) \bowtie \pi_{BC}(r)$. □

A	B	C
0	0	0
0	0	1
1	0	0
1	0	1
1	1	1

$=$

A	B
0	0
1	0
1	1

$\bowtie$

B	C
0	0
0	1
1	1

Figure 6: Relation $r(ABC)$ satisfies the MVD $B \rightarrow\rightarrow A$.

As indicated in Figure 1, there is subclass of (nonembedded) MVDs called *conflict-free* MVD. Unlike arbitrary sets of MVDs, conflict-free MVDs can be *faithfully* represented in a *unique* acyclic hypergraph. In these situations, the acyclic hypergraph is called a *perfect-map* [4]. That is, every MVD logically implied by the conflict-free set can be inferred from

A	B	C
0	0	0
0	0	1
1	0	0
1	1	1

$\neq$

A	B
0	0
1	0
1	1

$\bowtie$

B	C
0	0
0	1
1	1

$=$

A	B	C
0	0	0
0	0	1
1	0	0
1	0	1
1	1	1

Figure 7: Relation $r'(ABC)$ does not satisfy the MVD $B \rightarrow\rightarrow A$.

A_1	A_2	A_3	A_4	A_5	A_6
0	0	0	0	1	0
0	0	0	1	1	0
1	0	1	1	0	1
1	1	0	1	1	0

Figure 8: Relation $r(R)$ satisfies the AJD, $\bowtie \mathcal{R}$, where $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$ is the acyclic hypergraph depicted in Figure 2.

the acyclic hypergraph, and every MVD inferred from the acyclic hypergraph is logically implied by the conflict-free set.

The conflict-free class of MVDs has many desirable properties in database applications [4]. In fact, a conflict-free set of MVDs is equivalent to an *acyclic join dependency* (AJD) [4]. An AJD can be used to losslessly decompose a relation into two or more projections (smaller relations). Let $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$ be an acyclic hypergraph on the set of attributes $R = R_1 \cup R_2 \cup \ldots \cup R_n$. We say that a relation $r(R)$ satisfies the *acyclic join dependency* (AJD), $\bowtie \{R_1, R_2, \ldots, R_n\}$ if:

$$r(R) = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \bowtie \pi_{R_n}(r). \tag{10}$$

That is, r decomposes losslessly onto $\mathcal{R}$. We also write $\bowtie \{R_1, R_2, \ldots, R_n\}$ as $\bowtie \mathcal{R}$. It follows that a relation $r(R)$ satisfies the acyclic join dependency $\bowtie \mathcal{R} \equiv \bowtie \{R_1, R_2, \ldots, R_n\}$ if and only if $r(R)$ contains the result of joining *all* joinable tuples in $\pi_{R_1}(r), \pi_{R_2}(r), \ldots, \pi_{R_n}(r)$.

Example 4 Relation $r(R)$ in Figure 8 satisfies the AJD, $\bowtie \mathcal{R}$, where $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$ is the acyclic hypergraph in Figure 2. That is,

$$r(A_1A_2A_3A_4A_5A_6) = \pi_{A_1A_2A_3}(r) \bowtie \pi_{A_2A_3A_4}(r) \bowtie \pi_{A_2A_3A_5}(r) \bowtie \pi_{A_5A_6}(r). \ \square$$

The *separation* method [4] is used to infer MVDs from an acyclic hypergraph. Let $\mathcal{R}$ be an acyclic hypergraph on the set R of attributes and $X, Y \subseteq R$. The MVD $X \twoheadrightarrow Y$ is inferred from the acyclic hypergraph $\mathcal{R}$ if and only if Y is the union of some of the disconnected components in the hypergraph $\mathcal{R}$ with the set of nodes X deleted.

Example 5 Consider the following acyclic hypergraph $\mathcal{R}$ on $R = ABCDEFGH$:

$$\mathcal{R} \ = \ \{R_1 = AB, \ R_2 = BCD, \ R_3 = DE, \ R_4 = DFG, \ R_5 = DFH\}.$$

Deleting the node D, we obtain:

$$\mathcal{R}' \ = \ \{R'_1 = AB, \ R'_2 = BC, \ R'_3 = E, \ R'_4 = FG, \ R'_5 = FH\}.$$

The disconnected components in $\mathcal{R}'$ are:

$$S_1 = ABC, \qquad S_2 = E, \qquad S_3 = FGH.$$

By definition, the MVDs $D \twoheadrightarrow ABC$, $D \twoheadrightarrow E$, $D \twoheadrightarrow FGH$, and $D \twoheadrightarrow ABCE$ can be inferred from $\mathcal{R}$. On the other hand, the MVD $D \twoheadrightarrow BC$ is *not* inferred from $\mathcal{R}$ since BC is not equal to the union of some of the sets in $\{S_1, S_2, S_3\}$.

The next example illustrates the notion of perfect-map.

Example 6 Consider the following set C of MVDs on $R = A_1A_2A_3A_4A_5A_6$:

$$\begin{aligned} C \;\; = \;\; & \{A_2A_3 \twoheadrightarrow A_1,\ A_2A_3 \twoheadrightarrow A_4,\ A_2A_3 \twoheadrightarrow A_5A_6, \\ & A_5 \twoheadrightarrow A_1A_2A_3A_4,\ A_5 \twoheadrightarrow A_6,\ A_2A_3A_5 \twoheadrightarrow A_1\}. \end{aligned} \tag{11}$$

This set C of MVDs can be *faithfully* represented by the acyclic hypergraph $\mathcal{R}$ in Figure 2. According to the separation method for inferring MVDs from an acyclic hypergraph, every MVD in C can be inferred from $\mathcal{R}$. Obviously, every MVD logically implied by C can then be inferred from $\mathcal{R}$, and every MVD inferred from $\mathcal{R}$ is logically implied by C. Thus, the acyclic hypergraph $\mathcal{R}$ in Figure 2 is a *perfect-map* of the set C of MVDs in Equation (11).

Example 6 indicates that the set C of MVDs in Equation (11) is *conflict-free*. It is important to realize, however, that there are some sets of MVDs which cannot be faithfully represented by a single acyclic hypergraph.

Example 7 Consider the following set C of MVDs on $R = A_1A_2A_3$:

$$C \;\; = \;\; \{\ A_1 \twoheadrightarrow A_2,\ A_3 \twoheadrightarrow A_2\ \}. \tag{12}$$

There is no *single* acyclic hypergraph that can simultaneously encode both MVDs in C. For example, consider the acyclic hypergraph

$$\mathcal{R} \;\; = \;\; \{R_1 = A_1A_2,\ R_2 = A_1A_3\}.$$

The MVD $A_1 \twoheadrightarrow A_2$ in C can be inferred from $\mathcal{R}$ using the method of separation. However, the MVD $A_3 \twoheadrightarrow A_2$ cannot be inferred from $\mathcal{R}$ using separation. On the other hand, the acyclic hypergraph

$$\mathcal{R}' \;\; = \;\; \{R'_1 = A_2A_3,\ R'_2 = A_1A_3\},$$

represents the MVD $A_3 \twoheadrightarrow A_2$ but not $A_1 \twoheadrightarrow A_2$.

Example 7 indicates that the class of *conflict-free* MVDs is a subclass of the class of MVD. For example, C in Equation (12) is a member in the class of MVDs, but is not a member in class of conflict-free MVDs.

We now turn our attention to the more general class of *embedded* MVDs. Embedded MVDs are those which hold in projections of a relation, but not necessarily the relation itself.

Example 8 Consider the relation $r(ABCD)$ at the top of Figure 9. It can be easily verified that $r(ABCD)$ does not satisfy the MVD $B \twoheadrightarrow A$, namely:

$$r(ABCD) \;\; \neq \;\; \pi_{AB}(r) \bowtie \pi_{BCD}(r).$$

However, the projection $\pi_{ABC}(r)$ does satisfy $B \twoheadrightarrow A$:

$$r(ABC) \;\; = \;\; \pi_{AB}(r) \bowtie \pi_{BC}(r). \;\square$$

$$r(ABCD) \;=\; \begin{array}{|cccc|}\hline A & B & C & D \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ \hline \end{array} \neq \pi_{AB}(r) \bowtie \pi_{BCD}(r)$$

$$r' \;=\; \pi_{ABC}(r) \;=\; \begin{array}{|ccc|}\hline A & B & C \\ \hline 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ \hline \end{array} = \pi_{AB}(r') \bowtie \pi_{BC}(r')$$

Figure 9: At the top of the figure, relation $r(ABCD)$ does *not* satisfy the MVD $B \rightarrow\rightarrow A$. However, its projection $\pi_{ABC}(r)$ does satisfy $B \rightarrow\rightarrow A$ as shown at the bottom of the figure.

Example 8 indicates that MVDs are sensitive to context. It is important to specify the set of attributes over which a particular MVD holds [3]. We say relation $r(R)$ satisfies the *embedded multivalued dependency* (EMVD) $X \rightarrow\rightarrow Y \mid Z$ if $\pi_{XYZ}(r)$ satisfies the MVD $X \rightarrow\rightarrow Y$, where X, Y, Z are subsets of R. For example, the relation $R(ABCD)$ in Figure 9 satisfies the EMVD $B \rightarrow\rightarrow A \mid C$, where the set ABC is the *context*.

For notational convenience we will write the *nonembedded* MVD $X \rightarrow\rightarrow Y \mid (R - XY)$ as $X \rightarrow\rightarrow Y$ if the context R is understood. Likewise, we will use the notation $X \rightarrow\rightarrow Y \mid Z$ for EMVDs to explicitly state that the context is XYZ. The EMVD $X \rightarrow\rightarrow Y \mid Z$ becomes the (nonembedded) MVD $X \rightarrow\rightarrow Y$ in the situation when $XYZ = R$. It is therefore clear that MVD is a *special case* of the more general EMVD class as shown in Figure 1.

2.3 Bayesian Networks

Before we introduce our Bayesian database model, let us first review some pertinent notions used in Bayesian networks [31].

Let $R = \{A_1, A_2, \ldots, A_m\}$ denote a finite set of discrete variables (attributes). Each variable A_i is associated with a finite domain D_{A_i}. Let D be the Cartesian product of the domains D_{A_i}, $1 \leq i \leq m$. A *joint probability distribution* (jpd) [17, 28, 31] on D is a function p on D, $p : D \rightarrow [0, 1]$. That is, this function p assigns to each tuple $t \equiv < t(A_1), t(A_2), \ldots, t(A_m) > \in D$ a real number $0 \leq p(t) \leq 1$ and p is normalized, namely, $\sum_{t \in D} p(t) = 1$. For convenience, we write a joint probability distribution p as $p(A_1, A_2, \ldots, A_m)$ over the set of variables R. In particular, we use $p(a_1, a_2, \ldots, a_m)$ to denote the value $p(t) = p(< t(A_1), t(A_2), \ldots, t(A_m) >)$. That is, $p(a_1, a_2, \ldots, a_m)$ denotes the probability value $p(< t(A_1), t(A_2), \ldots, t(A_m) >)$ of the function p for a particular *instantiation* of the variables $A_1, A_2, \ldots, A_m$. In general, a *potential* [17] is a function q on D such

that $q(t)$ is a nonnegative real number and $\sum_{t \in D} q(t)$ is positive, i.e., at least one $q(t) > 0$. Each potential q can be transformed to a joint probability distribution p by *normalization*, that is, by setting $p(t) = q(t) / \sum_{v \in D} q(v)$.

We now introduce the fundamental notion of *probabilistic conditional independency*. Let X, Y and Z be subsets of variables in R. Let $x = t(X)$, $y = t(Y)$, and $z = t(Z)$ denote arbitrary values of X, Y and Z, respectively. We say Y and Z are *conditionally independent* given X under the joint probability distribution p, denoted $I_p(Y, X, Z)$, if

$$p(y \mid xz) \;=\; p(y \mid x), \tag{13}$$

whenever $p(xz) > 0$. This conditional independency $I_p(Y, X, Z)$ can be equivalently written as

$$p(yxz) \;=\; \frac{p(yx) \cdot p(xz)}{p(x)}. \tag{14}$$

We write $I_p(Y, X, Z)$ as $I(Y, X, Z)$ if the joint probability distribution p is understood. In the special case where $Y \cup X \cup Z = R$, we call the probabilistic conditional independency $I(Y, X, Z)$ *nonembedded*; otherwise $I(Y, X, Z)$ is called *embedded*. (*Nonembedded* probabilistic conditional independency is also called *fixed context* [14] and *full* [26].)

Example 9 Let $R = \{A, B, C, D\}$. Consider the following set $\mathbf{C}$ of probabilistic conditional independencies:

$$\mathbf{C} \;=\; \{\; I(A, B, C),\; I(A, BC, D)\;\}.$$

The first independency $I(A, B, C)$ is *embedded* since $\{A, B, C\} \subset R$. The second independency $I(A, BC, D)$ is *nonembedded* since $\{A, B, C, D\} = R$.

By the chain rule, a joint probability distribution $p(A_1, A_2, \ldots, A_m)$ can always be written as:

$$p(A_1, A_2, \ldots, A_m) \;=\; p(A_1) \cdot p(A_2|A_1) \cdot p(A_3|A_1, A_2) \cdot \ldots \cdot p(A_m|A_1, A_2, \ldots, A_{m-1}).$$

The above equation is an *identity*. However, one can use conditional independencies that are assumed to hold in the problem domain to obtain a simpler representation of a joint distribution.

Example 10 Consider a joint distribution $p(A_1, A_2, A_3, A_4, A_5, A_6)$ and the set $\mathbf{C}$ of probabilistic conditional independencies:

$$\begin{aligned} \mathbf{C} \;=\; \{\; & I(A_1, \emptyset, \emptyset),\; I(A_2, A_1, \emptyset),\; I(A_3, A_1, A_2),\; I(A_4, A_2A_3, A_1),\; I(A_5, A_2A_3, A_1A_4), \\ & I(A_6, A_5, A_1A_2A_3A_4)\;\}, \end{aligned} \tag{15}$$

namely,

$$\begin{aligned} p(A_1) &= p(A_1), \\ p(A_2|A_1) &= p(A_2|A_1), \\ p(A_3|A_1, A_2) &= p(A_3|A_1), \\ p(A_4|A_1, A_2, A_3) &= p(A_4|A_2, A_3), \\ p(A_5|A_1, A_2, A_3, A_4) &= p(A_5|A_2, A_3), \\ p(A_6|A_1, A_2, A_3, A_4, A_5) &= p(A_6|A_5). \end{aligned}$$

By the chain rule, $p(A_1, A_2, A_3, A_4, A_5, A_6)$ can be written as:

$$\begin{aligned} & p(A_1, A_2, A_3, A_4, A_5, A_6) \\ = \; & p(A_1) \cdot p(A_2|A_1) \cdot p(A_3|A_1, A_2) \cdot p(A_4|A_1, A_2, A_3) \cdot p(A_5|A_1, A_2, A_3, A_4) \cdot p(A_6|A_1, A_2, A_3, A_4, A_5). \end{aligned}$$

Utilizing the conditional independencies in $\mathbf{C}$, the joint distribution $p(A_1, A_2, A_3, A_4, A_5, A_6)$ can now be expressed in a simpler form:

$$\begin{aligned} & p(A_1, A_2, A_3, A_4, A_5, A_6) \\ = \; & p(A_1) \cdot p(A_2|A_1) \cdot p(A_3|A_1) \cdot p(A_4|A_2, A_3) \cdot p(A_5|A_2, A_3) \cdot p(A_6|A_5). \end{aligned} \tag{16}$$

We can represent all of the probabilistic conditional independencies satisfied by this joint distribution by the *directed acyclic graph* (DAG) shown in Figure 10. This DAG together with the conditional probability distributions $p(A_1)$, $p(A_2|A_1)$, $p(A_3|A_1)$, $p(A_4|A_2, A_3)$, $p(A_5|A_2, A_3)$, and $p(A_6|A_5)$, define a *Bayesian network* [31].

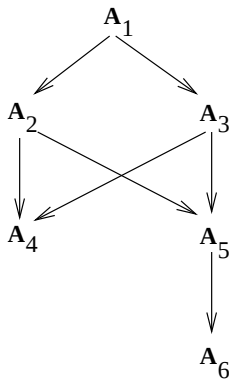

Figure 10: The DAG representing all of the probabilistic conditional independencies satisfied by the joint distribution in Equation (16).

Example 10 demonstrates that Bayesian networks provide a convenient semantic modeling tool which greatly facilitates the *acquisition* of probabilistic knowledge. That is, a human expert can indirectly specify a joint distribution by specifying probability conditional independencies and the corresponding conditional probability distributions.

The set $\mathbf{C}$ of conditional independencies in Equation (15) is an example of a *causal input list* [32] (a *stratified protocol* [38]), since $\mathbf{C}$ precisely defines a *directed acyclic graph* (DAG). Such a DAG encodes all the probabilistic conditional independencies satisfied by a particular joint distribution. The method of *d-separation* [31] is used to infer conditional independencies from a DAG. For example, the conditional independency of A_1 and A_5 given $A_2A_3A_4$, i.e., $I(A_5, A_2A_3A_4, A_1)$, can be inferred from the DAG in Figure 10 using the d-separation method. The set of all independencies that can be inferred from a DAG is called a *conflict-free* set of probabilistic conditional independencies. Given a DAG, the associated

conflict-free set of conditional independencies is:

$$\begin{aligned} & \text{Conflict-free set of probabilistic conditional independencies} \\ = \; & \{\, \mathbf{c} \mid \mathbf{c} \text{ can be inferred from the given DAG by d-separation} \}, \end{aligned} \tag{17}$$

where $\mathbf{c}$ denotes a probabilistic conditional independency. It should be clear from Equation (17) that the conflict-free set of conditional independencies for a given DAG contains all the independencies in the causal input list used to define the DAG, possibly along with other independencies. Conversely, every conditional independency logically implied by the conflict-free set can be inferred from the given DAG. In other words, a DAG is a *perfect-map* [31] of the conflict-free set.

As illustrated in Figure 1 (left), it is important to realize that conflict-free sets of probabilistic conditional independencies are a *special class* within the more general class of probabilistic conditional independencies. There are *arbitrary* sets of probabilistic conditional independencies that are not conflict-free. In other words, there are some sets of conditional independencies that cannot be *faithfully* encoded as a DAG.

Example 11 Consider the following set $\mathbf{C}$ of probabilistic conditional independencies on $\{A, B, C\}$:

$$\mathbf{C} \;=\; \{\; I(A, B, C),\; I(A, C, B)\; \}. \tag{18}$$

There is no *single* DAG that can simultaneously encode both independencies in $\mathbf{C}$.

Example 11 demonstrates that conflict-free sets of independencies are a *special class* of probabilistic conditional independencies as depicted in Figure 1. In this example, the set $\mathbf{C}$ of conditional independencies in Equation (18) belongs to the general class of probabilistic conditional independencies, but does not belong to the class of conflict-free sets.

To facilitate the computation of marginal distributions in practice, it is useful to transform a Bayesian network into a (decomposable) Markov network. A *Markov network* [17] consists of an acyclic hypergraph and a corresponding set of marginal distributions. The DAG of a given Bayesian network can be converted by the *moralization* and *triangulation* procedures [17, 31] into an acyclic hypergraph. (An acyclic hypergraph in fact represents a chordal undirected graph. Each maximal clique in the graph corresponds to a hyperedge in the acyclic hypergraph [4].) For example, by applying these procedures to the DAG in Figure 10, we obtain the acyclic hypergraph depicted in Figure 2. Similarly, *local computation* procedures [44] can be applied to transform the conditional probability distributions into marginal distributions defined over the acyclic hypergraph. The joint probability distribution in Equation (16) can be rewritten, in terms of marginal distributions over the acyclic hypergraph in Figure 2, as:

$$p(A_1, A_2, A_3, A_4, A_5, A_6) \;=\; \frac{p(A_1, A_2, A_3) \cdot p(A_2, A_3, A_4) \cdot p(A_2, A_3, A_5) \cdot p(A_5, A_6)}{p(A_2, A_3) \cdot p(A_2, A_3) \cdot p(A_5)}. \tag{19}$$

The Markov network representation of probabilistic knowledge in Equation (19) is typically used for inference in many practical applications.

The following two examples emphasize the fact that Markov networks only use a subclass of nonembedded independencies. The first example demonstrates that Markov networks only use nonembedded independencies.

Example 12 Consider the marginal distribution $p(A_1, A_2, A_3)$ obtained from the Bayesian network in Equation (16):

$$\begin{aligned}
p(A_1, A_2, A_3) &= \sum_{A_4, A_5, A_6} p(A_1, A_2, A_3, A_4, A_5, A_6) \\
&= \sum_{A_4, A_5, A_6} p(A_1) \cdot p(A_2|A_1) \cdot p(A_3|A_1) \cdot p(A_4|A_2, A_3) \cdot p(A_5|A_2, A_3) \cdot p(A_6|A_5) \\
&= p(A_1) \cdot p(A_2|A_1) \cdot p(A_3|A_1) \cdot \sum_{A_4, A_5, A_6} p(A_4|A_2, A_3) \cdot p(A_5|A_2, A_3) \cdot p(A_6|A_5) \\
&= p(A_1) \cdot p(A_2|A_1) \cdot p(A_3|A_1) \\
&= \frac{p(A_1, A_2) \cdot p(A_1, A_3)}{p(A_1)}. \qquad (20)
\end{aligned}$$

By the definition of conditional independency in Equation (14), A_2 and A_3 are conditionally independent given A_1 in Equation (20). In other words, the Bayesian network in Equation (16) encodes the *embedded* probabilistic conditional independency $I(A_3, A_1, A_2)$.

On the other hand, consider the marginal distribution $p(A_1, A_2, A_3)$ obtained from the Markov network in Equation (19):

$$\begin{aligned}
p(A_1, A_2, A_3) &= \sum_{A_4, A_5, A_6} p(A_1, A_2, A_3, A_4, A_5, A_6) \\
&= \sum_{A_4, A_5, A_6} \frac{p(A_1, A_2, A_3) \cdot p(A_2, A_3, A_4) \cdot p(A_2, A_3, A_5) \cdot p(A_5, A_6)}{p(A_2, A_3) \cdot p(A_2, A_3) \cdot p(A_5)} \\
&= \frac{p(A_1, A_2, A_3)}{p(A_2, A_3) \cdot p(A_2, A_3)} \cdot \sum_{A_4, A_5, A_6} \frac{p(A_2, A_3, A_4) \cdot p(A_2, A_3, A_5) \cdot p(A_5, A_6)}{p(A_5)} \\
&= \frac{p(A_1, A_2, A_3)}{p(A_2, A_3) \cdot p(A_2, A_3)} \cdot p(A_2, A_3) \cdot p(A_2, A_3) \\
&= p(A_1, A_2, A_3). \qquad (21)
\end{aligned}$$

Equation (21) indicates that the Markov network in Equation (19) does *not* encode the embedded probabilistic conditional independency $I(A_3, A_1, A_2)$.

Example 12 indicates that Bayesian networks are more expressive than Markov networks, since Bayesian networks encode *both* embedded and nonembedded independencies whereas Markov networks *only* encode nonembedded independencies. As with d-separation in DAGs, the method of *separation* (see Section 2.2) is used to infer nonembedded independencies from an acyclic hypergraph. The next example demonstrates that there are certain sets of nonembedded independencies which cannot be *faithfully* encoded by an acyclic hypergraph.

Example 13 Consider the following set $\mathbf{C}$ of nonembedded probabilistic conditional independencies on $\{A, B, C\}$:

$$\mathbf{C} = \{\ I(A, B, C),\ I(A, C, B)\ \}. \qquad (22)$$

$\mathbf{r}(R) =$

A_1	A_2	$\ldots$	A_m	A_p
$\mathbf{t}_1(A_1)$	$\mathbf{t}_1(A_2)$	$\ldots$	$\mathbf{t}_1(A_m)$	$\mathbf{t}_1(A_p) = p(t_1)$
$\mathbf{t}_2(A_1)$	$\mathbf{t}_2(A_2)$	$\ldots$	$\mathbf{t}_2(A_m)$	$\mathbf{t}_2(A_p) = p(t_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathbf{t}_s(A_1)$	$\mathbf{t}_s(A_2)$	$\ldots$	$\mathbf{t}_s(A_m)$	$\mathbf{t}_s(A_p) = p(t_s)$

Figure 11: A joint distribution p expressed as a relation $\mathbf{r}$ over $R = \{A_1, A_2, \ldots, A_m\}$.

There is no *single* acyclic hypergraph that can simultaneously encode both nonembedded independencies in $\mathbf{C}$.

As illustrated in Figure 1, Example 13 demonstrates that the class of conflict-free nonembedded independencies is a *special* class of the more general class of nonembedded independencies. That is, $\mathbf{C}$ in Equation (22) belongs to the class of nonembedded independencies in class (**2a**), but not the BAJD class (**2b**).

We conclude this section, by reiterating that Bayesian networks are not constructed using an *arbitrary* input set of embedded independencies chosen from class (**1a**), just as Markov networks do not use *arbitrary* sets of nonembedded independencies from class (**2a**).

2.4 A Bayesian Database Model

Here we review our Bayesian database model [41, 44] which serves as a unified approach for both Bayesian networks and relational databases.

A joint probability distribution p can be represented as a relation $\mathbf{r}$. The relation $\mathbf{r}$ representing the jpd $p(R)$ has attributes $R \cup \{A_p\}$, where the column labeled by A_p stores the probability value. For example, the relation $\mathbf{r}$ representing the jpd $p(R)$ on the set of variables $R = \{A_1, A_2, \ldots, A_m\}$ is shown in Figure 11. Each tuple $\mathbf{t} \in \mathbf{r}$ is defined by $\mathbf{t}(R) = t \in D$ and $\mathbf{t}(A_p) = p(t)$. That is, $\mathbf{t} = < t, p(t) >$. In our Bayesian database model, however, the relation $\mathbf{r}$ only contains the tuples t with a *positive* probability value, namely, $p(t) > 0$. For convenience we will say relation $\mathbf{r}$ is on R with the attribute A_p understood by context. That is, relations denoted by boldface represent probability distributions.

Let $\mathbf{r}(R)$ be a relation over $R = \{A_1, A_2, \ldots, A_m\}$ and X be a subset of R. The *marginalization of* $\mathbf{r}$ *onto* X, written $\tau_X(\mathbf{r})$, is defined as

$$\tau_X(\mathbf{r}) = \{\ \mathbf{t}(XA_{p(X)}) \mid \mathbf{t}(X) \in \pi_X(\mathbf{r}) \text{ and } \mathbf{t}(A_{p(X)}) = \sum_{\mathbf{t}' \in \mathbf{r}} \mathbf{t}'(A_p)\ \}, \tag{23}$$

where $\mathbf{t}'(X) = \mathbf{t}(X)$. In [17, 28, 31], the relation $\tau_X(\mathbf{r})$ is called the *marginal distribution* $p(X)$ of $p(R)$ onto X. By definition, $\tau_X(\mathbf{r})$ does not contain any tuples with zero probability.

Example 14 Given the relation $\mathbf{r}(A_1A_2A_3)$ in Figure 12, the marginalization of $\mathbf{r}$ onto A_1A_2 is the relation $\tau_{A_1A_2}(\mathbf{r})$ depicted in Figure 13. □

$\mathbf{r}(A_1A_2A_3)$ =

A_1	A_2	A_3	A_p
0	0	0	0.1
0	0	1	0.6
1	0	0	0.3

Figure 12: A relation $\mathbf{r}$ on the scheme $R = A_1A_2A_3$.

$\tau_{A_1A_2}(\mathbf{r})$ =

A_1	A_2	$A_{p(A_1A_2)}$
0	0	0.7
1	0	0.3

Figure 13: The marginalization $\tau_{A_1A_2}(\mathbf{r})$ of relation $\mathbf{r}(A_1A_2A_3)$ onto A_1A_2. This relation $\mathbf{r}(A_1A_2A_3)$ is defined in Figure 12.

The *product join* of two relations $\mathbf{r}_1(X)$ and $\mathbf{r}_2(Y)$, written $\mathbf{r}_1(X) \times \mathbf{r}_2(Y)$, is defined as

$$\begin{aligned} & \mathbf{r}_1(X) \times \mathbf{r}_2(Y) \qquad (24) \\ = \; & \{\mathbf{t}(XYA_{p(X)\cdot p(Y)}) \mid \mathbf{t}(XY) \in \pi_X(\mathbf{r}_1) \bowtie \pi_Y(\mathbf{r}_2) \text{ and } \mathbf{t}(A_{p(X)\cdot p(Y)}) = \mathbf{t}(A_{p(X)}) \cdot \mathbf{t}(A_{p(Y)})\}. \end{aligned}$$

Thus, $\mathbf{r}_1(X) \times \mathbf{r}_2(Y)$ represents the product distribution $p(X) \cdot p(Y)$ of the two individual distributions $p(X)$ and $p(Y)$.

Example 15 The product join $\mathbf{r}(A_1A_2) \times \mathbf{r}(A_2A_3)$ of relations $\mathbf{r}(A_1A_2)$ and $\mathbf{r}(A_2A_3)$ is shown in Figure 14. □

The important notion of probabilistic conditional independency is represented as *Bayesian* MVD (BMVD) in our Bayesian database model. BMVD is a generalization of MVD in the standard relational database model, and belongs to the class of *nonembedded* probabilistic conditional independencies in a Bayesian network.

Let R be a relation scheme, X and Y be subsets of R, and $Z = R - XY$. A relation $\mathbf{r}(R)$ satisfies the *Bayesian multivalued dependency* (BMVD) $X \Rightarrow\Rightarrow Y$ if, for any two tuples

A_1	A_2	$A_{p(A_1A_2)}$
1	1	0.2
2	1	0.4
1	2	0.4

×

A_2	A_3	$A_{p(A_2A_3)}$
1	1	0.2
1	2	0.5
3	1	0.3

=

A_1	A_2	A_3	$A_{p(A_1A_2)\cdot p(A_2A_3)}$
1	1	1	0.04
1	1	2	0.10
2	1	1	0.08
2	1	2	0.12

Figure 14: The product join $\mathbf{r}(A_1A_2) \times \mathbf{r}(A_2A_3)$ of relations $\mathbf{r}(A_1A_2)$ and $\mathbf{r}(A_2A_3)$.

$\mathbf{t}_1$ and $\mathbf{t}_2$ in $\mathbf{r}$ with $\mathbf{t}_1(X) = \mathbf{t}_2(X)$, there exists a tuple $\mathbf{t}_3$ in $\mathbf{r}$ satisfying the following two conditions:

(i) $\mathbf{t}_3(XY) = \mathbf{t}_1(XY)$ and $\mathbf{t}_3(XZ) = \mathbf{t}_2(XZ)$,

(ii) The probability value $\mathbf{t}_3(A_{p(XYZ)})$ can be written as:

$$\mathbf{t}_3(A_{p(XYZ)}) \quad = \quad \frac{\mathbf{t}'(A_{p(XY)}) \ \cdot \ \mathbf{t}''(A_{p(XZ)})}{\mathbf{t}'''(A_{p(X)})}, \tag{25}$$

where $\mathbf{t}' \in \tau_{XY}(\mathbf{r})$, $\mathbf{t}'' \in \tau_{XZ}(\mathbf{r})$, $\mathbf{t}''' \in \tau_X(\mathbf{r})$.

Condition (i) is the usual definition of the MVD $X \rightarrow\rightarrow Y$ in the relational database model. This is the *necessary* (qualitative) condition for the BMVD $X \Rightarrow\Rightarrow Y$ to hold. Condition (ii) stipulates the *sufficient* (quantitative) condition for this BMVD to hold.

It can be easily verified that the BMVD $X \Rightarrow\Rightarrow Y$ is a necessary and sufficient condition for a relation $\mathbf{r}(XYZ)$ to be losslessly decomposed as

$$\mathbf{r}(XYZ) \quad = \quad \tau_{XY}(\mathbf{r}) \times \tau_{XZ}(\mathbf{r}) \times \tau_X(\mathbf{r})^{-1}, \tag{26}$$

where the relation $\tau_X(\mathbf{r})^{-1}$ is defined using $\tau_X(\mathbf{r})$ as follows:

$$\tau_X(\mathbf{r})^{-1} = \{\mathbf{t}(XA_{1/p(X)}) \ \mid \ \left\{ \begin{array}{l} \mathbf{t}' \in \tau_X(\mathbf{r}) \\ \mathbf{t}(X) = \mathbf{t}'(X) \\ \mathbf{t}(A_{1/p(X)}) = 1/\mathbf{t}'(A_{p(X)}) \end{array} \right. \}.$$

Note that this inverse relation $\tau_X(\mathbf{r})^{-1}$ is well defined because $\tau_X(\mathbf{r})$ contains no tuple $\mathbf{t}'$ such that $\mathbf{t}'(A_{p(X)}) = 0$. By introducing a new operator $\otimes$, Equation (26) can be written as:

$$\tau_{XY}(\mathbf{r}) \otimes \tau_{XZ}(\mathbf{r}) \equiv \tau_{XY}(\mathbf{r}) \times \tau_{XZ}(\mathbf{r}) \times \tau_X(\mathbf{r})^{-1}.$$

We call this binary operator $\otimes$ the *Markov join*. Thus, in terms of this notation, we say that a relation $\mathbf{r}(R)$ satisfies the BMVD $X \Rightarrow\Rightarrow Y$, if and only if

$$\mathbf{r}(XYZ) = \tau_{XY}(\mathbf{r}) \otimes \tau_{XZ}(\mathbf{r}). \tag{27}$$

It should be noted that:

$$X \Rightarrow\Rightarrow Y \quad \Longleftrightarrow \quad X \Rightarrow\Rightarrow Y - X.$$

Example 16 Equation (27) indicates that the relation $\mathbf{r}(A_1A_2A_3)$ in Figure 15 satisfies the BMVD $A_2 \Rightarrow\Rightarrow A_1$, since we can easily verify that

$$\mathbf{r}(A_1A_2A_3) = \tau_{A_1A_2}(\mathbf{r}) \otimes \tau_{A_2A_3}(\mathbf{r}).$$

In contrast, the relations $\mathbf{r}(A_1A_2A_3)$ in Figures 16 and 17 do not satisfy the BMVD $A_2 \Rightarrow\Rightarrow A_1$. □

$\mathbf{r}$ =

A_1	A_2	A_3	A_p
0	0	0	0.3
0	0	1	0.3
0	1	1	0.2
1	0	0	0.1
1	0	1	0.1

=

A_1	A_2	$A_{p(A_1,A_2)}$
0	0	0.6
0	1	0.2
1	0	0.2

$\otimes$

A_2	A_3	$A_{p(A_2,A_3)}$
0	0	0.4
0	1	0.4
1	1	0.2

Figure 15: Relation $\mathbf{r}(A_1A_2A_3)$ satisfies the BMVD $A_2 \Rightarrow\Rightarrow A_1$.

$\mathbf{r}$ =

A_1	A_2	A_3	A_p
0	0	0	0.1
0	0	1	0.2
0	1	1	0.3
1	0	0	0.3
1	0	1	0.1

$\neq$

A_1	A_2	$A_{p(A_1A_2)}$
0	0	0.3
0	1	0.3
1	0	0.4

$\otimes$

A_2	A_3	$A_{p(A_2A_3)}$
0	0	0.4
0	1	0.3
1	1	0.3

$\neq$

A_1	A_2	A_3	$A_{\frac{p(A_1A_2)\cdot p(A_2A_3)}{p(A_2)}}$
0	0	0	0.1714285
0	0	1	0.1285714
0	1	1	0.3000000
1	0	0	0.2285714
1	0	1	0.1714285

= $\mathbf{r}'$

Figure 16: Relation $\mathbf{r}(A_1A_2A_3)$ does not satisfy the BMVD $A_2 \Rightarrow\Rightarrow A_1$.

A_1	A_2	A_3	A_p
0	0	0	0.1
0	0	1	0.2
1	0	0	0.3
1	1	1	0.4

$\neq$

A_1	A_2	$A_{p(A_1A_2)}$
0	0	0.3
1	0	0.3
1	1	0.4

$\otimes$

A_2	A_3	$A_{p(A_2A_3)}$
0	0	0.4
0	1	0.2
1	1	0.4

=

A_1	A_2	A_3	$A_{p'}$
0	0	0	0.2
0	0	1	0.1
1	0	0	0.2
1	0	1	0.1
1	1	1	0.4

Figure 17: Relation $\mathbf{r}(A_1A_2A_3)$ does not satisfy the BMVD $A_2 \Rightarrow\Rightarrow A_1$.

$$\mathbf{r}(ABCD) \;=\; \begin{array}{|cccc|c|}\hline A & B & C & D & A_{p(ABCD)} \\ \hline 0 & 0 & 0 & 0 & 0.1 \\ 0 & 0 & 1 & 1 & 0.1 \\ 1 & 0 & 0 & 0 & 0.2 \\ 1 & 0 & 1 & 0 & 0.2 \\ 1 & 1 & 1 & 1 & 0.4 \\ \hline \end{array} \;\neq\; \tau_{AB}(\mathbf{r}) \otimes \tau_{BCD}(\mathbf{r})$$

$$\mathbf{r}' \;=\; \tau_{ABC}(\mathbf{r}) \;=\; \begin{array}{|ccc|c|}\hline A & B & C & A_{p(ABD)} \\ \hline 0 & 0 & 0 & 0.1 \\ 0 & 0 & 1 & 0.1 \\ 1 & 0 & 0 & 0.2 \\ 1 & 0 & 1 & 0.2 \\ 1 & 1 & 1 & 0.4 \\ \hline \end{array} \;=\; \tau_{AB}(\mathbf{r}') \otimes \tau_{BC}(\mathbf{r}')$$

Figure 18: At the top of the figure, relation $\mathbf{r}(ABCD)$ does *not* satisfy the BMVD $B \Rightarrow\Rightarrow A$. However, its marginal $\tau_{ABC}(\mathbf{r})$ does satisfy $B \Rightarrow\Rightarrow A$ as shown at the bottom of the figure.

It is important to realize that BMVDs are also sensitive to context as the following example demonstrates.

Example 17 Consider the relation $\mathbf{r}(ABCD)$ at the top of Figure 18. It can be easily verified that $\mathbf{r}(ABCD)$ does not satisfy the BMVD $B \Rightarrow\Rightarrow A$, namely:

$$\mathbf{r}(ABCD) \;\neq\; \tau_{AB}(\mathbf{r}) \otimes \tau_{BCD}(\mathbf{r}).$$

However, the marginal $\tau_{ABC}(\mathbf{r})$ does satisfy $B \Rightarrow\Rightarrow A$:

$$\mathbf{r}(ABC) \;=\; \tau_{AB}(\mathbf{r}) \otimes \tau_{BC}(\mathbf{r}). \;\square$$

It should be clear that stating the generalized relation $\mathbf{r}(XYZW)$, for a given joint probability distribution $p(XYZW)$, satisfies the BEMVD $X \Rightarrow\Rightarrow Y|Z$ is equivalent to stating that Y and Z are conditionally independent given X under p in Equation (14). Thus, we can use the terms BEMVD and probabilistic conditional independency interchangeably.

A *conflict-free* set of BMVDs can be faithfully represented by a single acyclic hypergraph. As in relational databases, it can be shown that a conflict-free set of BMVDs is equivalent to a *Bayesian acyclic join dependency*. Let $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$ be an acyclic hypergraph on the set of attributes $R = R_1 \cup R_2 \cup \ldots \cup R_n$. We say a *Bayesian acyclic join dependency* (BAJD), written $\otimes\mathcal{R}$, is satisfied by a relation $\mathbf{r}$, if

$$\mathbf{r}(R) = (\ldots((\tau_{R_1}(\mathbf{r}) \otimes \tau_{R_2}(\mathbf{r})) \otimes \tau_{R_3}(\mathbf{r}))\ldots) \otimes \tau_{R_n}(\mathbf{r}), \tag{28}$$

where the sequence $R_1, R_2, \ldots, R_n$ is a hypertree construction ordering for $\mathcal{R}$.

$$\mathbf{r}(A_1A_2A_3) \;=\; \begin{array}{|ccc|c|}\hline A_1 & A_2 & A_3 & A_p \\ \hline 0 & 0 & 0 & 0.1 \\ 0 & 0 & 1 & 0.6 \\ 1 & 0 & 0 & 0.3 \\ \hline \end{array} \quad , \quad r(A_1A_2A_3) \;=\; \begin{array}{|ccc|}\hline A_1 & A_2 & A_3 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ \hline \end{array}$$

Figure 19: $\mathbf{r}(A_1A_2A_3)$ is an example of a (probabilistic) relation representing a joint probability distribution $p(A_1A_2A_3)$. The corresponding relation $r(A_1A_2A_3)$ of $\mathbf{r}(A_1A_2A_3)$ defined by Equation 29 is shown on the right.

$$\tau_{A_1A_2}(\mathbf{r}) \;=\; \begin{array}{|cc|c|}\hline A_1 & A_2 & A_{p(A_1A_2)} \\ \hline 0 & 0 & 0.7 \\ 1 & 0 & 0.3 \\ \hline \end{array} \quad , \quad \pi_{A_1A_2}(\mathbf{r}) \;=\; \begin{array}{|cc|}\hline A_1 & A_2 \\ \hline 0 & 0 \\ 1 & 0 \\ \hline \end{array}$$

Figure 20: The relation $\tau_{A_1A_2}(\mathbf{r})$ is the marginalization of $\mathbf{r}(A_1A_2A_3)$ in Figure 19, and $\pi_{A_1A_2}(\mathbf{r})$ is the projection of $\mathbf{r}(A_1A_2A_3)$.

2.5 Comparison of the Bayesian and Relational Database Models

In this section we provide a brief exposition on the relationship between the Bayesian and the standard relational database models. Our goal is to demonstrate that the main difference between these two models is the choice of operators. The traditional relational operators are special cases of the corresponding probabilistic operators. This realization will pave the way to adopt relational database techniques for solving similar problems in probabilistic database systems.

Let $R = \{A_1, A_2, \ldots, A_m\}$ denote a set of attributes (variables. Every (probabilistic) relation $\mathbf{r}(R)$ in the Bayesian database model consists of two components: a joint distribution $p(R)$ and a (standard) relation $r(R)$. The relation $r(R)$ is defined as:

$$r(R) \;=\; \{t(A_1A_2\ldots A_m) \mid \mathbf{t}(A_1A_2\ldots A_mA_p) \in \mathbf{r}(R)\}. \tag{29}$$

The relationship between $r(R)$ and $p(R)$ is illustrated by an example in Figure 19. Conversely, a joint probability distribution $p(R)$ can be represented as a (probabilistic) relation $\mathbf{r}(R)$ in the Bayesian database model.

The marginalization τ and the product join $\times$ in the Bayesian database model are obviously generalizations of the projection π and the natural join $\bowtie$ operators in the standard relational database model as illustrated in Figures 20 and 21.

In the relational database model, a relation $r(XYZ)$ has a lossless decomposition:

$$r(XYZ) = \pi_{XY}(r) \bowtie \pi_{XZ}(r)$$

if and only if the MVD $X \rightarrow\rightarrow Y$ holds in r. In parallel, a probabilistic relation $\mathbf{r}(XYZ)$ has a lossless decomposition:

$$\mathbf{r}(XYZ) = \tau_{XY}(\mathbf{r}) \otimes \tau_{XZ}(\mathbf{r})$$

A_1	A_2
1	1
2	1
1	2

$\bowtie$

A_2	A_3
1	1
1	2
3	1

$=$

A_1	A_2	A_3
1	1	1
1	1	2
2	1	1
2	1	2

A_1	A_2	$A_{p(A_1A_2)}$
1	1	0.2
2	1	0.4
1	2	0.4

$\times$

A_2	A_3	$A_{p(A_2A_3)}$
1	1	0.2
1	2	0.5
3	1	0.3

$=$

A_1	A_2	A_3	$A_{p(A_1A_2)\cdot p(A_2A_3)}$
1	1	1	0.04
1	1	2	0.10
2	1	1	0.08
2	1	2	0.12

Figure 21: The top of the Figure depicts the natural join $r(A_1A_2) \bowtie r(A_2A_3)$ of relations $r(A_1A_2)$ and $r(A_2A_3)$. The bottom depicts the product join $\mathbf{r}(A_1A_2) \times \mathbf{r}(A_2A_3)$ of relations $\mathbf{r}(A_1A_2)$ and $\mathbf{r}(A_2A_3)$.

if and only if the BMVD $X \Rightarrow\Rightarrow Y$ holds in $\mathbf{r}$, i.e., Y and Z are conditionally independent given X in the joint probability distribution $p(XYZ)$ used to define $\mathbf{r}(XYZ)$. Since the probabilistic relation $\mathbf{r}(XYZ)$ does not contain any tuples $\mathbf{t}(A_{p(XYZ)}) = 0$, the MVD $X \rightarrow\rightarrow Y$ is a *necessary* condition for $\mathbf{r}$ to have a lossless decomposition.

Pairwise lossless decomposition can be generalized by the notion of acyclic join dependency. A probabilistic relation $\mathbf{r}(XYZ)$ is said to satisfy the Bayesian acyclic join dependency (BAJD), $\otimes\mathcal{R}$, if $\mathbf{r}(R)$ can be expressed as:

$$\mathbf{r}(R) \quad = \quad \tau_{R_1}(\mathbf{r}) \otimes \tau_{R_2}(\mathbf{r}) \otimes \ldots \tau_{R_n}(\mathbf{r}),$$

where $R = R_1 \cup R_2 \cup \ldots \cup R_n$, and $\mathcal{R}$ is an acyclic hypergraph, i.e., the sequence $R_1, R_2, \ldots, R_n$ is a hypertree construction ordering.

A BAJD in the Bayesian database model represents a Markov distribution and is defined by two components:

(i) $r(R) = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \pi_{R_n}(r)$ is an acyclic join dependency $\bowtie \mathcal{R}$ in the relational database model, and

(ii) the joint probability distribution $p(R)$ which defines $\mathbf{r}(R)$ can be expressed as

$$p(R) \quad = \quad \frac{p(R_1) \cdot p(R_2) \cdot \ldots \cdot p(R_n)}{p(R_1 \cap R_2) \cdot p(R_2 \cap R_3) \cdot p(R_{n-1} \cap R_n)},$$

where $\mathcal{J} = \{R_1 \cap R_2, R_2 \cap R_3, \ldots, R_{n-1} \cap R_n\}$ is the set of J-keys of the acyclic hypergraph $\mathcal{R}$.

The above discussion clearly indicates that a probabilistic reasoning system is a general form of the traditional relational database model. The relationships between these two models are summarized in Table 1.

Relational Database	Bayesian Network	Bayesian Database
relation $r(R)$	distribution $p(R)$	relation $\mathbf{r}(R)$
projection $\pi_X(r)$	marginal $p(X)$	marginal $\tau_X(\mathbf{r})$
natural join $\bowtie$	multiplication $\cdot$	product join $\times$
MVD $X \rightarrow\rightarrow Y$	conditional independency $p(Y\|X,Z) = p(Y\|X)$	BMVD $X \Rightarrow\Rightarrow Y$
AJD $R_1 \bowtie R_2 \bowtie \ldots \bowtie R_n$	Markov Network $\frac{p(R_1)\cdot p(R_2)\cdot \ldots \cdot p(R_n)}{p(R_1 \cap R_2)\cdot \ldots \cdot p(R_{n-1} \cap R_n)}$	BAJD $((\tau_{R_1}(\mathbf{r}) \otimes \tau_{R_2}(\mathbf{r})) \otimes \ldots) \otimes \tau_{R_n}(\mathbf{r})$

Table 1: The terminology used for corresponding notions in the standard relational database model, Bayesian networks, and our Bayesian database model.

An important question naturally arises as: do the *implications problems* in the relational databases and Bayesian networks coincide with each other? An attempt to answer this question is the focus of the remaining part of this paper.

3 The Implication Problem for Different Classes of Dependencies

Before we study the implication problem in detail, let us first introduce some basic notions. Here we will use the terms *relation* and *joint probability distribution* interchangeably; similarly, for the terms *dependency* and *independency*.

Let $\sum$ be a set of dependencies defined on a set of attributes R. By $SAT_R(\sum)$, we denote the set of all relations on R that satisfy all of the dependencies in $\sum$. We write $SAT_R(\sum)$ as $SAT(\sum)$ when R is understood, and $SAT(\sigma)$ for $SAT(\{\sigma\})$, where σ is a single dependency. We say $\sum$ *logically implies* σ, written $\sum \models \sigma$, if $SAT(\sum) \subseteq SAT(\sigma)$. In other words, σ is logically implied by $\sum$ if there is no counter-example relation such that all of the dependencies in $\sum$ are satisfied but σ is not.

The *implication problem* is to test whether a given set $\sum$ of dependencies logically implies another dependency σ, namely,

$$\sum \models \sigma. \tag{30}$$

Clearly, the first question to answer is whether such a problem is *solvable*, i.e., whether there

exists some method to provide a positive or negative answer for any given instance of the implication problem. We consider two methods for answering this question.

The first method for testing implication is by axiomatization. An *inference axiom* is a rule that states if a relation satisfies certain dependencies, then it must satisfy certain other dependencies. Given a set $\sum$ of dependencies and a set of inference axioms, the *closure* of $\sum$, written $\sum^+$, is the smallest set containing $\sum$ such that the inference axioms cannot be applied to the set to yield a dependency not in the set. More specifically, the set $\sum$ *derives* a dependency σ, written $\sum \vdash \sigma$, if σ is in $\sum^+$. A set of inference axioms is *sound* if whenever $\sum \vdash \sigma$, then $\sum \models \sigma$. A set of inference axioms is *complete* if the converse holds, that is, if $\sum \models \sigma$, then $\sum \vdash \sigma$. In other words, if $\sum$ logically implies the dependency σ, then $\sum$ derives σ. A sequence P of dependencies over R is a *derivation sequence* on $\sum$ if every dependency in P is either

(i) a member of $\sum$, or

(ii) follows from previous dependencies in P by an application of one of the given inference axioms.

To solve the implication problem by axiomatization, we can (in principle) compute $\sum^+$ under a complete axiomatization, then we test whether $\sigma \in \sum^+$. In other words, if no complete axiomatization exists for a given class of dependency, then the implication problem for that class cannot be solved using the axiomatization method.

The second method for testing implication is a nonaxiomatic method such as the *chase* algorithm. The chase algorithm in relational database model is a powerful tool to obtain many nontrivial results. We will show that the chase algorithm can also be applied to the implication problem for probabilistic conditional independencies.

The rest of this paper is organized as follows. Since nonembedded dependencies are best understood, we therefore choose to analyze the pair (BMVD, MVD), and the subclasses (conflict-free BMVD, conflict-free MVD) before the others. Next we consider the embedded dependencies. First we study the pair of (conflict-free BEMVD, conflict-free EMVD). The conflict-free BEMVD class has been studied extensively as these dependencies form the basis for the construction of Bayesian networks. Finally, we analyze the pair (BEMVD, EMVD). This pair subsumes all the other previously studied pairs. This pair is particularly important to our discussion here, as its implication problems are *unsolvable* in contrast to the other *solvable* pairs such as (BMVD, MVD) and (conflict-free BEMVD, conflict-free EMVD).

4 Nonembedded Dependency

In this section, we study the implication problem for the class of nonembedded probabilistic conditional independency, called BMVD in our terminology. One way to demonstrate that the implication problem for BMVDs is solvable is to directly prove that a sound set of BMVD axioms are also *complete*. This is exactly the approach taken by Geiger and Pearl [14]. Here we take a different approach. Instead of directly demonstrating that the BMVD implication problem is solvable, we do it by establishing a one-to-one relationship between the implication problems of the pair (BMVD,MVD).

4.1 Nonembedded Multivalued Dependency

The MVD class of dependencies in the pair (BMVD,MVD) has been extensively studied in the standard relational database model. As mentioned before, MVD is the necessary and sufficient conditions for a lossless (binary) decomposition of a relation. In this section, we review two methods for solving the implication problem of the MVD class of data dependencies, namely, the axiomatic and nonaxiomatic methods.

(i) MVD Axiomatization

It is well known [3] that MVDs have a finite complete axiomatization.

Theorem 1 The following inference axioms (MVD1)-(MVD7) are both sound and complete for multivalued dependencies (MVDs):

$(MVD1)$ If $Y \subseteq X$, then $X \rightarrow\rightarrow Y$.
$(MVD2)$ If $X \rightarrow\rightarrow Y$ and $Y \rightarrow\rightarrow Z$, then $X \rightarrow\rightarrow Z - Y$.
$(MVD3)$ If $X \rightarrow\rightarrow Y$, and $X \rightarrow\rightarrow Z$, then $X \rightarrow\rightarrow YZ$.
$(MVD4)$ If $X \rightarrow\rightarrow Y$ and $X \rightarrow\rightarrow Z$, then $X \rightarrow\rightarrow Y \cap Z$, $X \rightarrow\rightarrow Y - Z$.
$(MVD5)$ If $X \rightarrow\rightarrow Y$, then $XZ \rightarrow\rightarrow Y$.
$(MVD6)$ If $X \rightarrow\rightarrow Y$ and $YW \rightarrow\rightarrow Z$, then $XW \rightarrow\rightarrow Z - (YW)$.
$(MVD7)$ If $X \rightarrow\rightarrow Y$, then $X \rightarrow\rightarrow R - (XY)$.

It should perhaps be noted that axioms (MVD1) and (MVD2) form a *minimal* set [27], i.e., all other axioms can be derived from these two axioms. Axioms (MVD1)-(MVD7) are called *reflexivity*, *transitivity*, *union*, *decomposition*, *augmentation*, *pseudotransitivity*, and *complementation*, respectively.

The usefulness of a *sound* axiomatization lies in the ability to derive dependencies that are not explicitly stated.

Example 18 Consider the following set C of MVDs:

$$C = \{AB \rightarrow\rightarrow D,\ AE \rightarrow\rightarrow F,\ BD \rightarrow\rightarrow G\},$$

on the set of attributes $R = ABDEFG$. The following is a *derivation sequence* of the MVD $AB \rightarrow\rightarrow G$.

1. $AB \rightarrow\rightarrow D$ (given)
2. $AB \rightarrow\rightarrow B$ (MVD1)
3. $AB \rightarrow\rightarrow BD$ (MVD3) from 1 and 2
4. $BD \rightarrow\rightarrow G$ (given)
5. $AB \rightarrow\rightarrow G$ (MVD2) from 3 and 4.

The derivation sequence of the MVD $AB \rightarrow\rightarrow G$ from the set C of MVDs using sound axioms ensures that C *logically implies* $AB \rightarrow\rightarrow G$, namely,

$$\{AB \rightarrow\rightarrow D,\ AE \rightarrow\rightarrow F,\ BD \rightarrow\rightarrow G\} \ \models\ AB \rightarrow\rightarrow G.$$

The above example demonstrates that whenever a dependency is derived using sound axioms, then the dependency is logically implied by the given input set. However, if the axioms are *not* complete, then there is no guarantee that the axioms will derive *all* of the logically implied dependencies.

(ii) A Nonaxiomatic Approach

The discussion presented here follows closely the description given in [23].

We begin by examining what it means for a relation to decompose losslessly. Let r be a relation on R, and $R_1 \cup R_2 \cup \ldots \cup R_n = R$. We say relation r *decomposes losslessly* onto a database scheme $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$ if

$$r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \bowtie \pi_{R_n}(r). \tag{31}$$

It can be shown that the left side is a subset of right side in Equation (31), namely,

$$r \ \subseteq\ \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \bowtie \pi_{R_n}(r).$$

In other words, every tuple $t \in r$ will also appear in $\pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \bowtie \pi_{R_n}(r)$. Thereby, the lossless decomposition in Equation (31) can be shown by demonstrating

$$\pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \bowtie \pi_{R_n}(r) \ \subseteq\ r.$$

In other words, showing that *every* tuple in the natural join of the projections is also a tuple in r. For example, the relation $r(ABC)$ in Figure 7 does not decompose losslessly onto database scheme $\mathcal{R} = \{R_1 = AB, R_2 = BC\}$ since the tuple $t = <\ 1\ 0\ 1\ >$ is in $\pi_{AB}(r) \bowtie \pi_{BC}(r)$ but is not an element of $r(ABC)$.

Shorthand notation is introduced for the right hand side of Equation (31). The *project-join mapping* defined by $\mathcal{R}$, written $m_{\mathcal{R}}$, is a function on relations on R defined by

$$m_{\mathcal{R}}(r) = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \bowtie \pi_{R_n}(r).$$

The important point to notice is that saying a relation $r(R)$ satisfies the AJD $\bowtie \mathcal{R}$ is the same as saying that $m_{\mathcal{R}}(r) = r$. For example, let $R = ABC$ and $\mathcal{R} = \{AB, BC\}$. The result of applying $m_{\mathcal{R}}$ to the relation $r(ABC)$ in Figure 7 (left) is the relation $r' = m_{\mathcal{R}}(r)$ in Figure 7 (right). Applying $m_{\mathcal{R}}$ to r' gives back r'. Project-join mappings can be represented in tabular form called tableaux.

A *tableau* T is both a tabular means of representing a project-join mapping and a template for a relation r on R. Whereas a relation contains tuples of values, a tableau contains rows

T =

A_1	A_2	A_3	A_4
a_1	b_1	a_3	b_2
b_3	a_2	a_3	b_4
a_1	b_5	a_3	a_4

Figure 22: A tableau T on the scheme $A_1A_2A_3A_4$.

r =

A_1	A_2	A_3	A_4
1	4	5	8
2	3	5	7
1	4	5	7

Figure 23: The relation r obtained as the result of applying ρ in Equation (32) to the tableau T in Figure 22.

of subscripted variables (symbols). The a and b variables are called *distinguished* and *non-distinguished* variables, respectively. We restrict the variables in a tableau to appear in only one column. We make the further restriction that at most one distinguished variable may appear in any column. By convention, if the scheme of a tableau is $A_1A_2 \ldots A_m$, then the distinguished variable appearing in the A_i-column will be a_i. For example, a tableau T on scheme $R = A_1A_2A_3A_4$ is shown in Figure 22. We obtain a relation from the tableau by substituting domain values for variables. Let T be a tableau and let

$$V = \{\ a_1, a_2, \ldots, a_m, b_1, b_2, \ldots\}$$

denote the set of its variables. A *valuation* ρ for T is a mapping from V to the Cartesian product $D_1 \times D_2 \times \ldots \times D_m$ such that $\rho(v)$ is in D_i when v is a variable appearing in the A_i-column. We extend the valuation from variables to rows and thence to the entire tableau. If $w = < v_1\ v_2 \ldots v_m >$ is a row in a tableau, we let $\rho(w) = < \rho(v_1)\ \rho(v_2) \ldots \rho(v_m) >$. We then let

$$\rho(T) = \{\ \rho(w) \mid w \text{ is a row in } T\ \}.$$

Example 19 Consider the following valuation ρ:

$$\rho(a_1) = 1, \quad \rho(a_2) = 3, \quad \rho(a_3) = 5, \quad \rho(a_4) = 7,$$
$$\rho(b_1) = 4, \quad \rho(b_2) = 8, \quad \rho(b_3) = 2, \quad \rho(b_4) = 7, \quad \rho(b_5) = 4,$$

The result of applying ρ to the tableau T in Figure 22 is the relation r in Figure 23. □

Similar to a project-join mapping, a tableau T on scheme R can be interpreted as a function on relations $r(R)$. In this interpretation we require that T have a distinguished

$T =$

A_1	A_2	A_3	A_4
a_1	a_2	b_1	b_2
b_3	a_2	a_3	b_4
b_5	b_6	a_3	a_4

Figure 24: The tableau T on $R = A_1A_2A_3A_4$.

$r =$

A_1	A_2	A_3	A_4
1	3	5	7
1	4	5	7
2	3	6	8

Figure 25: A relation r on $R = A_1A_2A_3A_4$.

variable in every column. Let w_d be the row of all distinguished variables. That is, if $R = A_1A_2 \ldots A_m$, then $w_d = < a_1\ a_2\ \ldots\ a_m >$. Row w_d is not necessarily in T. If r is a relation on scheme R, we let

$$T(r) = \{\ \rho(w_d) \mid \rho(T) \subseteq r\ \},$$

That is, if we find any valuation ρ that maps every row in T to a tuple in r, then $\rho(w_d)$ is in $T(r)$.

It is always possible to find a tableau $T_{\mathcal{R}}$ for representing a project-join mapping $m_{\mathcal{R}}$ defined by

$$m_{\mathcal{R}}(r) = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \bowtie \pi_{R_n}(r),$$

where $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$, and $R = R_1 \cup R_2 \cup \ldots \cup R_n$. The tableau $T_{\mathcal{R}}$ for $m_{\mathcal{R}}$ is defined as follows. The scheme for $T_{\mathcal{R}}$ is R. $T_{\mathcal{R}}$ has n rows, $w_1, w_2, \ldots, w_n$. Row w_i has the distinguished variable a_j in the A_j-column exactly when $A_j \in R_i$. The remaining nondistinguished variables in w_i are unique and do not appear in any other row of $T_{\mathcal{R}}$. For example, let $\mathcal{R} = \{\ R_1 = A_1A_2, R_2 = A_2A_3, R_3 = A_3A_4\ \}$ and R_1, R_2, R_3 be a hypertree construction for $\mathcal{R}$. The tableau $T_{\mathcal{R}}$ for $m_{\mathcal{R}}$ is depicted in Figure 24. Consider the relation r on $R = A_1A_2A_3A_4$ as shown in Figure 25. The valuation ρ, defined as

$$\begin{array}{llllll} \rho(a_1) = 1, & \rho(a_2) = 3, & \rho(a_3) = 6, & \rho(a_4) = 8, & & \\ \rho(b_1) = 5, & \rho(b_2) = 7, & \rho(b_3) = 2, & \rho(b_4) = 8, & \rho(b_5) = 2, & \rho(b_6) = 3, \end{array}$$

indicates that $< 1\ 3\ 6\ 8 >$ is in $T_{\mathcal{R}}(r)$. All of $T_{\mathcal{R}}(r)$ is depicted in Figure 26. It is easily verified that applying the project-join mapping $m_{\mathcal{R}}$ to the relation r in Figure 25 also produces the relation in Figure 26. That is, $T_{\mathcal{R}}(r) = m_{\mathcal{R}}(r)$.

$T(r)$ =

A_1	A_2	A_3	A_4
1	3	5	7
1	3	6	8
1	4	5	7
2	3	5	7
2	3	6	8

Figure 26: The relation $T(r)$, where $r(R)$ is the relation in Figure 25 and T is the tableau in Figure 24.

Lemma 1 [23] Let $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$ be a set of relation schemes, where $R = R_1 R_2 \ldots R_n$. The project-join mapping $m_{\mathcal{R}}$ and the tableau $T_{\mathcal{R}}$ define the same function between relations $r(R)$. That is, $m_{\mathcal{R}}(r) = T_{\mathcal{R}}(r)$ for all $r(R)$.

Lemma 1 indicates that saying that a relation $r(R)$ satisfies the AJD $\bowtie \mathcal{R}$ is the same as saying that $T_{\mathcal{R}}(r) = r$.

The notion of what it means for two tableaux to be equivalent is now described. Let T_1 and T_2 be tableaux on scheme R. We write $T_1 \sqsubseteq T_2$ if $T_1(r) \subseteq T_2(r)$ for all relations $r(R)$. Tableaux T_1 and T_2 are *equivalent*, written $T_1 \equiv T_2$, if $T_1 \sqsubseteq T_2$ and $T_2 \sqsubseteq T_1$. That is, $T_1 \equiv T_2$ if $T_1(r) = T_2(r)$ for every relation $r(R)$. Let $SAT(C)$ denote the set of relations $r(R)$ that satisfy all the constraints in C. If T_1 and T_2 are tableaux on R, then we say T_1 is *contained* by T_2 on $SAT(C)$, written $T_1 \sqsubseteq_{SAT(C)} T_2$, if $T_1(r) \subseteq T_2(r)$ for every relation r in $SAT(C)$. We say T_1 and T_2 are *equivalent* on $SAT(C)$, written

$$T_1 \equiv_{SAT(C)} T_2, \tag{32}$$

if $T_1 \sqsubseteq_{SAT(C)} T_2$ and $T_2 \sqsubseteq_{SAT(C)} T_1$.

We now consider a method for modifying tableaux while preserving equivalence. A *J-rule* for a set C of AJDs is a means to modify an arbitrary tableau T to a tableau T' such that $T \equiv_{SAT(C)} T'$. Let $\mathcal{R} = \{R_1, R_2, \ldots, R_q\}$ be a set of relation schemes and let $\bowtie \mathcal{R}$ be a AJD on R. Let T be a tableau on R and let $w_1, w_2, \ldots, w_q$(not necessarily distinct) be rows of T that are joinable on $\mathcal{R}$ with result w. Applying the *J-rule* for $\bowtie \mathcal{R}$ to tableau T allows us to form the tableau

$$T' = T \cup \{w\}.$$

If we view the tableau T as a relation, the generated row w can be expressed as

$$w \;=\; w_1(R_1) \bowtie w_2(R_2) \bowtie \ldots \bowtie w_n(R_n). \tag{33}$$

Example 20 Let $C = \{ \bowtie \{A_1A_2,\ A_2A_3A_4\} \}$ and T be the tableau in Figure 27. Rows w_1 and w_2 are joinable on A_2. We can then apply the *J-rule* for $\bowtie \{A_1A_2, A_2A_3A_4\}$ in C to rows $w_1 = < a_1\ a_2\ b_1\ b_2 >$ and $w_2 = < b_3\ a_2\ a_3\ b_4 >$ of T to generate the new row

$$\begin{aligned} w &= w_1(A_1A_2) \bowtie w_2(A_2A_3A_4) \\ &= < a_1\ a_2\ > \bowtie < a_2\ a_3\ b_4 > \\ &= < a_1\ a_2\ a_3\ b_4 > . \end{aligned}$$

$T =$

A_1	A_2	A_3	A_4
a_1	a_2	b_1	b_2
b_3	a_2	a_3	b_4
b_5	b_6	a_3	a_4

Figure 27: The tableau T on $R = A_1A_2A_3A_4$.

$T' =$

A_1	A_2	A_3	A_4
a_1	a_2	b_1	b_2
b_3	a_2	a_3	b_4
b_5	b_6	a_3	a_4
a_1	a_2	a_3	b_4

Figure 28: The tableau $T' = T \cup \{< a_1\ a_2\ a_3\ b_4 >\}$, where T is the tableau in Figure 27.

Tableau $T' = T \cup \{w\}$ in Figure 28 is the result of this application. Even though rows $w = < a_1\ a_2\ a_3\ b_4 >$ and $w_3 = < b_5\ b_6\ a_3\ a_4 >$ are joinable on A_3, we cannot construct the new row $< a_1\ a_2\ a_3\ a_4 >$ since no *J-rule* exists in C which applies to attribute A_3. □

It is worth mentioning that J-rule is also applicable to MVDs since MVD is a special case of AJD.

Theorem 2 [23] Let $\mathcal{R} = \{R_1, R_2, \ldots, R_q\}$ and T' be the result of applying the *J-rule* for $\bowtie \mathcal{R}$ to tableau T. Tableaux T and T' are equivalent on $SAT(\bowtie \mathcal{R})$.

The *chase* algorithm can now be described. Given T and C, apply the *J-rules* associated with the AJDs in C, *until no further change is possible*. The resulting tableau, written $chase_C(T)$, is equivalent to T on all relations in $SAT(C)$, i.e., $T \equiv_{SAT(C)} chase_C(T)$, and $chase_C(T)$ considered as a relation is in $SAT(C)$.

Theorem 3 [23] $C \models \bowtie \mathcal{R}$ if and only if $chase_C(T_{\mathcal{R}})$ contains the row of all distinguished variables.

Theorem 3 states that the chase algorithm is equivalent to logical implication. We illustrate Theorem 3 with the following example.

Example 21 Suppose we wish to test the implication problem $C \models c$ on scheme $R = A_1A_2A_3A_4$, where $C = \{\ A_2 \rightarrow\rightarrow A_1,\ \ A_3 \rightarrow\rightarrow A_4\ \}$ is a set of MVDs and $c = \bowtie \{A_1A_2,\ A_2A_3,\ A_3A_4\ \}$ is an AJD. We construct the initial tableau $T_{\mathcal{R}}$ in Figure 24 according to the database scheme $\mathcal{R}$ defined by c. Rows w_1 and w_2 are joinable on A_2. We can then apply the *J-rule* for $A_2 \rightarrow\rightarrow A_1$ in C to rows $w_1 = < a_1\ a_2\ b_1\ b_2 >$ and $w_2 = < b_3\ a_2\ a_3\ b_4 >$ of $T_{\mathcal{R}}$ to generate the new row

$$\begin{aligned} w_4 &= w_1(A_1A_2) \bowtie w_2(A_2A_3A_4) \\ &= < a_1\ a_2\ a_3\ b_4 > . \end{aligned}$$

$T_{\mathcal{R}} =$

A_1	A_2	A_3	A_4
a_1	a_2	b_1	b_2
b_3	a_2	a_3	b_4
b_5	b_6	a_3	a_4
a_1	a_2	a_3	b_4

Figure 29: Since $T_{\mathcal{R}}$ satisfies the MVD $A_2 \rightarrow\rightarrow A_1$ in C, by definition, rows w_1 and w_2 being joinable on A_2 imply that row $w_4 = < a_1\ a_2\ a_3\ b_4 >$ is also in $T_{\mathcal{R}}$.

$T_{\mathcal{R}} =$

A_1	A_2	A_3	A_4
a_1	a_2	b_1	b_2
b_3	a_2	a_3	b_4
b_5	b_6	a_3	a_4
a_1	a_2	a_3	b_4
a_1	a_2	a_3	a_4

Figure 30: Since $T_{\mathcal{R}}$ satisfies the MVD $A_3 \rightarrow\rightarrow A_4$ in C, by definition, rows w_4 and w_3 being joinable on A_3 imply that row $w_d = < a_1\ a_2\ a_3\ a_4 >$ is also in $T_{\mathcal{R}}$.

Tableau $T_{\mathcal{R}} \cup \{w_4\}$ is depicted in Figure 28. Similarly, rows w_4 and w_3 are joinable on A_3. We can then the *J-rule* for $A_3 \rightarrow\rightarrow A_4$ in C to rows $w_4 = < a_1\ a_2\ a_3\ b_4 >$ and $w_3 = < b_5\ b_6\ a_3\ a_4 >$ to generate the new row

$$\begin{aligned} w_d &= w_4(A_1A_2A_3) \bowtie w_3(A_3A_4) \\ &= < a_1\ a_2\ a_3\ a_4 > \end{aligned}$$

as shown in Figure 30. Row w_d is the row of all distinguished variables. By Theorem 3, C logically implies c. That is, any relation that satisfies the MVDs in C must also satisfy the AJD c. □

It should be noted that the resulting tableau in the chase algorithm is *unique* regardless of the order in which the J-rules were applied.

Theorem 4 [23] The chase computation for a set of AJDs is a *finite Church-Rosser* replacement system. Therefore, $chase_C(T_{\mathcal{R}})$ is always a singleton set.

This completes the review of the implication problem for relational data dependencies.

4.2 Nonembedded Probabilistic Conditional Independency

We now turn our attention to the class of nonembedded probabilistic conditional independency (BMVD) in the pair (BMVD, MVD). As in the MVD case, we will consider both the

axiomatic and nonaxiomatic methods to solve the implication problem for the BMVD class of probabilistic dependencies. However, we first show an immediate relationship between the inference of BMVDs and that of MVDs.

Lemma 2 [25, 41] Let $\mathbf{C}$ be a set of BMVDs on R and $\mathbf{c}$ a single BMVD on R. Then

$$\mathbf{C} \models \mathbf{c} \implies C \models c,$$

where $C = \{X \rightarrow\rightarrow Y \mid X \Rightarrow\Rightarrow Y \in \mathbf{C}\}$ is the set of MVDs corresponding to the BMVDs in $\mathbf{C}$, and c is the MVD corresponding to the BMVD $\mathbf{c}$.

Proof: Suppose $\mathbf{C} \models \mathbf{c}$. We will prove the claim by contradiction. That is, suppose that $C \not\models c$. By definition, there exists a relation $r(R)$ such that $r(R)$ satisfies all of the MVDs in C, but $r(R)$ does not satisfy the MVD c. Let k denote the number of tuples in $r(R)$. We construct a probabilistic relation $\mathbf{r}(R)$ from $r(R)$ by appending the attribute A_p. For each of the k tuples in $\mathbf{r}(R)$, set $\mathbf{t}(A_p) = 1/k$. Thus, $\mathbf{r}(R)$ represents a *uniform* distribution. In the uniform case [25, 41], $\mathbf{r}(R)$ satisfies $\mathbf{C}$ if and only if $r(R)$ satisfies C. Again using the uniform case, $\mathbf{r}(R)$ does not satisfy $\mathbf{c}$ since $r(R)$ does not satisfy c. By definition, $\mathbf{C}$ does not logically imply $\mathbf{c}$, namely, $\mathbf{C} \not\models \mathbf{c}$. A contradiction to the initial assumption that $\mathbf{C} \models \mathbf{c}$. Therefore, $C \models c$. □

With respect to the pair (BMVD,MVD) of *nonembedded* dependencies, Lemma 2 indicates that the statement

$$\mathbf{C} \models \mathbf{c} \implies C \models c$$

is a *tautology*. We now consider ways to solve the implication problem $\mathbf{C} \models \mathbf{c}$.

(i) BMVD Axiomatization

It can be easily shown that the following inference axioms for BMVDs are *sound*:

$(BMVD1)$ If $Y \subseteq X$, then $X \Rightarrow\Rightarrow Y$.
$(BMVD2)$ If $X \Rightarrow\Rightarrow Y$ and $Y \Rightarrow\Rightarrow Z$, then $X \Rightarrow\Rightarrow Z - Y$.
$(BMVD3)$ If $X \Rightarrow\Rightarrow Y$, and $X \Rightarrow\Rightarrow Z$, then $X \Rightarrow\Rightarrow YZ$.
$(BMVD4)$ If $X \Rightarrow\Rightarrow Y$ and $X \Rightarrow\Rightarrow Z$, then $X \Rightarrow\Rightarrow Y \cap Z$, $X \Rightarrow\Rightarrow Y - Z$.
$(BMVD5)$ If $X \Rightarrow\Rightarrow Y$, then $XZ \Rightarrow\Rightarrow Y$.
$(BMVD6)$ If $X \Rightarrow\Rightarrow Y$ and $YW \Rightarrow\Rightarrow Z$, then $XW \Rightarrow\Rightarrow Z - (YW)$,
$(BMVD7)$ If $X \Rightarrow\Rightarrow Y$, then $X \Rightarrow\Rightarrow R - (XY)$.

Axiom (BMVD1) holds trivially for any relation $\mathbf{r}(R)$ with $XY \subseteq R$. We now show that axiom (BMVD2) is sound. Recall that

$$X \Rightarrow\Rightarrow Y \iff X \Rightarrow\Rightarrow Y - X.$$

Thus, without loss of generality, let $R = XYZW$, where X, Y, Z and W are pairwise disjoint. By definition, the BMVDs $X \Rightarrow\Rightarrow Y$ and $Y \Rightarrow\Rightarrow Z$ mean

$$p(XYZW) = \frac{p(XY) \cdot p(XZW)}{p(X)}, \tag{34}$$

and

$$p(XYZW) = \frac{p(YZ) \cdot p(XYW)}{p(Y)}, \tag{35}$$

respectively. Computing the marginal distribution $p(XYZ)$ from both Equations (34) and (35), we respectively obtain:

$$p(XYZ) = \frac{p(XY) \cdot p(XZ)}{p(X)}, \tag{36}$$

and

$$p(XYZ) = \frac{p(YZ) \cdot p(XY)}{p(Y)}. \tag{37}$$

By Equations (36) and (37), we have:

$$\frac{p(XZ)}{p(X)} = \frac{p(YZ)}{p(Y)}. \tag{38}$$

By Equations (38) and (35), we obtain:

$$p(XYZW) = \frac{p(XZ) \cdot p(XYW)}{p(X)}. \tag{39}$$

Equation (39) is the definition of the BMVD $X \Rightarrow\Rightarrow Z$. The other axioms can be shown sound in a similar fashion.

Note that there is a one-to-one correspondence between the above inference rules for BMVDs and those MVD inference axioms (MVD1)-(MVD7) in Theorem 1. Since the BMVD axioms (BMVD1)-(BMVD7) are *sound*, it can immediately be shown that the implication problems coincide in the pair (BMVD,MVD).

Theorem 5 Given the *complete* axiomatization (MVD1)-(MVD7) for the MVD class. Then

$$\mathbf{C} \models \mathbf{c} \iff C \models c,$$

where $\mathbf{C}$ is a set of BMVDs, $C = \{ X \rightarrow\rightarrow Y \mid X \Rightarrow\Rightarrow Y \in \mathbf{C} \}$ is the corresponding set of MVDs, and c is the MVD corresponding to a BMVD $\mathbf{c}$.

Proof: ($\Rightarrow$) Holds by Lemma 2.

($\Leftarrow$) Let $C \models c$. By Theorem 1, $C \models c$ implies that $C \vdash c$. That is, there exists a derivation sequence s of the MVD c by applying the MVD axioms to the MVDs in C. On the other hand, since each MVD axiom has a corresponding BMVD axiom. This means there exists a derivation sequence $\mathbf{s}$ of the BMVD $\mathbf{c}$ using the BMVDs axioms on the BMVDs in $\mathbf{C}$, which parallels the derivation sequence s of the MVD c. That is, $\mathbf{C} \vdash \mathbf{c}$. Since the BMVD axioms are sound, $\mathbf{C} \vdash \mathbf{c}$ implies that $\mathbf{C} \models \mathbf{c}$. $\Box$

Theorem 5 indicates that the implication problems coincide in the pair (BMVD,MVD), as indicated in Figure 1. The following result is an immediate consequence and is stated without proof.

Corollary 1 The axioms (BMVD1)-(BMVD7) are both *sound* and *complete* for the class of nonembedded probabilistic conditional independency.

By Corollary 1, it is not surprising then that Geiger and Pearl [14] showed that their alternative complete axioms for BMVDs were also complete for MVDs. The main point of this section is to foster the notion that the Bayesian database model is intrinsically related to the standard relational database model. For example, by examining the implication problem for BMVD in terms of MVD, it is clear and immediate that the implication problems coincide in the pair (BMVD,MVD).

(ii) A Nonaxiomatic Method

We now present a *nonaxiomatic* method for testing the implication problem for nonembedded probabilistic conditional independencies. The standard chase algorithm can be modified for such a purpose by appropriately defining the manipulation of tableaux. However, we will then demonstrate that such a generalization is not necessary.

We briefly outline how a probabilistic chase can be formulated. A more complete description is given in [40]. The standard tableau T on a set of attributes $R = A_1A_2 \ldots A_m$ is augmented with attribute A_p. Each traditional row $w = < a_1a_2 \ldots a_m >$ is appended with probability symbol $p(a_1, a_2, \ldots, a_m)$. That is, a probabilistic tableau $\mathbf{T}$ contains rows $\mathbf{w} = < w, p(w) >$. In testing whether $\mathbf{C} \models \mathbf{c}$, we construct the initial tableau $\mathbf{T}_{\mathcal{R}}$ in the same fashion as in testing $C \models c$, where C and c are the corresponding MVDs, and $\mathcal{R}$ is the acyclic hypergraph corresponding to $\mathbf{c}$ (and c).

We now consider a method to modify probabilistic tableaux. We generalize the notion of J-rule for a MVD $X \rightarrow\rightarrow Y$ as follows. Let $\mathbf{T}$ be a probabilistic tableau on XYZ, $X \Rightarrow\Rightarrow Y$ a BMVD in a given set $\mathbf{C}$ of BMVDs, and $\mathbf{w}_1, \mathbf{w}_2$ be two *joinable* rows on X. A *Markov-join* rule (MJ-rule) for the BMVD $X \Rightarrow\Rightarrow Y$ is a means to add the new row $\mathbf{w} = < w, p(w) >$ to $\mathbf{T}$, where w is defined in the usual sense according the J-rule for the corresponding MVD $X \rightarrow\rightarrow Y$, and the probability symbol $p(w)$ is defined as:

$$p(w) \quad = \quad \frac{p(w_1(XY)) \cdot p(w_2(XZ))}{p(w_1(X))}. \tag{40}$$

Example 22 Let $\mathbf{C} = \{A_2 \Rightarrow\Rightarrow A_1,\ A_3 \Rightarrow\Rightarrow A_4\}$ and consider the tableau $\mathbf{T}_{\mathcal{R}}$ in Figure 31. It can be seen that rows

$$\mathbf{w}_1 = < a_1\ a_2\ b_1\ b_2\ p(a_1a_2b_1b_2) >$$

and

$$\mathbf{w}_2 = < b_3\ a_2\ a_3\ b_4\ p(b_3a_2a_3b_4) >$$

are joinable on A_2. We can then apply the MJ-rule for the BMVD $A_2 \Rightarrow\Rightarrow A_1$ in $\mathbf{C}$ to generate a new row $\mathbf{w}_4 = < a_1, a_2, a_3, b_4, p(a_1, a_2, a_3, b_4) >$, where by Equation (40),

$$p(a_1, a_2, a_3, b_4) \quad = \quad \frac{p(a_1a_2) \cdot p(a_2a_3b_4)}{p(a_2)}.$$

The new row $\mathbf{w}_4$ is added to $\mathbf{T}_{\mathcal{R}}$ as shown in Figure 32 on the left. Similarly, rows $\mathbf{w}_3 = < b_3\ b_5\ a_3\ a_4\ p(b_3b_5a_3a_4) >$ and $\mathbf{w}_4 = < a_1\ a_2\ a_3\ b_4\ \frac{p(a_1a_2)p(a_2a_3b_4)}{p(a_2)} >$ are joinable on A_3. By Equation (40), the MJ-rule for the BMVD $A_3 \Rightarrow\Rightarrow A_4$ in $\mathbf{C}$ can be applied to rows $\mathbf{w}_3$ and $\mathbf{w}_4$ to generate the new row

$$\mathbf{w}_5 \quad = \quad < a_1\ a_2\ a_3\ a_4\ \frac{p(a_1a_2)p(a_2a_3)p(a_3a_4)}{p(a_2)p(a_3)} >.$$

The tableau $\mathbf{T}_{\mathcal{R}} \cup \{\mathbf{w}_4\} \cup \{\mathbf{w}_5\}$ is shown in Figure 31 on the left. □

$\mathbf{T}_{\mathcal{R}}$ =

A_1	A_2	A_3	A_4	A_p
a_1	a_2	b_1	b_2	$p(a_1a_2b_1b_2)$
b_3	a_2	a_3	b_4	$p(b_3a_2a_3b_4)$
b_5	b_6	a_3	a_4	$p(b_5b_6a_3a_4)$

$T_{\mathcal{R}}$ =

A_1	A_2	A_3	A_4
a_1	a_2	b_1	b_2
b_3	a_2	a_3	b_4
b_5	b_6	a_3	a_4

Figure 31: The initial tableau $\mathbf{T}_{\mathcal{R}}$ constructed according to the BAJD $\mathbf{c} = \otimes\{A_1A_2,\ A_2A_3,\ A_3A_4\}$ is shown on the left of the figure. The initial tableau $T_{\mathcal{R}}$ constructed according to the AJD $c = \bowtie \{A_1A_2,\ A_2A_3,\ A_3A_4\}$ is shown on the right.

The *probabilistic* chase algorithm is now introduced. Given $\mathbf{T}$ and $\mathbf{C}$, apply the *MJ-rules* associated with the BMVDs in $\mathbf{C}$, *until no further change is possible*. The resulting tableau, written $chase_{\mathbf{C}}(\mathbf{T})$, is equivalent to $\mathbf{T}$ on all relations in $SAT(\mathbf{C})$. That is, $\mathbf{T}(\mathbf{r}) = chase_{\mathbf{C}}(\mathbf{T})(\mathbf{r})$, for every probabilistic relation $\mathbf{r}$ satisfying every BMVD in $\mathbf{C}$. Furthermore, $chase_{\mathbf{C}}(\mathbf{T})$ considered as a relation is in $SAT(\mathbf{C})$. The next result indicates that the probabilistic chase algorithm is a *nonaxiomatic* method for testing the implication problem for the BMVD class.

Theorem 6 Let $\mathbf{C}$ be a set of BMVDs on $R = A_1A_2 \ldots A_m$, and $\mathbf{c}$ be the BMVD $X \Rightarrow\Rightarrow Y$ on R. Then

$$\mathbf{C} \models \mathbf{c} \iff < a_1\ a_2 \ldots a_m\ p(a_1, a_2, \ldots, a_m) = \frac{p(xy) \cdot p(xz)}{p(x)} > \text{ is a row in } chase_{\mathbf{C}}(\mathbf{T}_{\mathcal{R}}),$$

where $\mathcal{R} = \{XY, XZ\}$ is the acyclic hypergraph corresponding to $\mathbf{c}$.

	A_1	A_2	A_3	A_4	A_p
$\mathbf{w}_1$	a_1	a_2	b_1	b_2	$p(a_1a_2b_1b_2)$
$\mathbf{w}_2$	b_3	a_2	a_3	b_4	$p(b_3a_2a_3b_4)$
$\mathbf{w}_3$	b_5	b_6	a_3	a_4	$p(b_5b_6a_3a_4)$
$\mathbf{w}_4$	a_1	a_2	a_3	b_4	$\frac{p(a_1a_2)p(a_2a_3b_4)}{p(a_2)}$
$\mathbf{w}_5$	a_1	a_2	a_3	a_4	$\frac{p(a_1a_2)p(a_2a_3)p(a_3a_4)}{p(a_2)p(a_3)}$

	A_1	A_2	A_3	A_4
w_1	a_1	a_2	b_1	b_2
w_2	b_3	a_2	a_3	b_4
w_3	b_5	b_6	a_3	a_4
w_4	a_1	a_2	a_3	b_4
w_5	a_1	a_2	a_3	a_4

Figure 32: The tableaux obtained by adding the new rows $\mathbf{w}_4$ and $\mathbf{w}_5$ is shown on the left of the figure. The standard use of the corresponding J-rules is shown on the right.

Proof: ($\Rightarrow$) We first show that the row of all distinguished variables $< a_1\, a_2 \ldots a_m\, p(a_1, a_2, \ldots, a_m) >$ must appear in $chase_{\mathbf{C}}(\mathbf{T}_{\mathcal{R}})$. Given $\mathbf{C} \models \mathbf{c}$. By contradiction, suppose that the row $< a_1\ a_2 \ldots a_m\ p(a_1, a_2, \ldots, a_m) >$ does not appear in $chase_{\mathbf{C}}(\mathbf{T}_{\mathcal{R}})$. This means that the MJ-rules corresponding to the BMVDs in $\mathbf{C}$ cannot be applied to the joinable rows to generate the row $< a_1\ a_2 \ldots a_m\ p(a_1, a_2, \ldots, a_m) >$. This implies that the J-rules corresponding to the MVDs in $C = \{V \twoheadrightarrow W \mid V \Rightarrow\Rightarrow W \in \mathbf{C}\}$ cannot be applied to the joinable rows in $T_{\mathcal{R}}$ to generate the row $< a_1\ a_2 \ldots a_m >$ of all distinguished variables, where c is the MVD corresponding to the BMVD $\mathbf{c}$. By Theorem 3, the row $< a_1\ a_2 \ldots a_m >$ not appearing in $chase_C(T_{\mathcal{R}})$ means that $C \not\models c$, where $chase_C(T_{\mathcal{R}})$ is the result of chasing c under C. By Theorem 5, $C \not\models c$ implies that $\mathbf{C} \not\models \mathbf{c}$. A contradiction. Therefore, the row $< a_1\ a_2 \ldots a_m\ p(a_1, a_2, \ldots, a_m) >$ must appear in $chase_{\mathbf{C}}(\mathbf{T}_{\mathcal{R}})$.

We now show that $p(a_1, a_2, \ldots, a_m)$ can be factorized as desired. By contradiction, suppose that:

$$p(a_1, a_2, \ldots, a_m) \quad \neq \quad \frac{p(xy) \cdot p(xz)}{p(x)}.$$

This means that $chase_{\mathbf{C}}(\mathbf{T}_{\mathcal{R}})$, considered as a probabilistic relation, satisfies the BMVDs in $\mathbf{C}$ but does not satisfy the BMVD $\mathbf{c}$. By definition, $\mathbf{C} \not\models \mathbf{c}$. A contradiction. Therefore,

$$p(a_1, a_2, \ldots, a_m) \quad = \quad \frac{p(xy) \cdot p(xz)}{p(x)}.$$

($\Leftarrow$) Given the row $< a_1\ a_2 \ldots a_m\ p(a_1, a_2, \ldots, a_m) >$ appears in $chase_{\mathbf{C}}(\mathbf{T}_{\mathcal{R}})$. This means that the MJ-rules corresponding to the BMVDs in $\mathbf{C}$ can be applied to $\mathbf{T}_{\mathcal{R}}$ to generate the row $< a_1\ a_2 \ldots a_m\ p(a_1, a_2, \ldots, a_m) >$. This implies that the J-rules corresponding to the MVDs in $C = \{V \twoheadrightarrow W \mid V \Rightarrow\Rightarrow W \in \mathbf{C}\}$ can be applied to the joinable rows in $T_{\mathcal{R}}$ to generate the row $< a_1\ a_2 \ldots a_m >$ of all distinguished variables, where c is the MVD corresponding to the BMVD $\mathbf{c}$. By Theorem 3, the row $< a_1\ a_2 \ldots a_m >$ appearing in $chase_C(T_{\mathcal{R}})$ means that $C \models c$, where $chase_C(T_{\mathcal{R}})$ is the result of chasing c under C. By Theorem 5, $C \models c$ implies that $\mathbf{C} \models \mathbf{c}$. □

Theorem 6 indicates that $\mathbf{C} \models \mathbf{c}$ if and only if the row of all distinguished variables appears in $chase_{\mathbf{C}}(\mathbf{T}_{\mathcal{R}})$, i.e., $p(a_1, a_2, \ldots, a_m)$ can always be factorized according to the BMVD being tested.

Example 23 Suppose we wish to test whether $\mathbf{C} \models \mathbf{c}$, where $\mathbf{C} = \{ A_1 \Rightarrow\Rightarrow A_2,\ A_2 \Rightarrow\Rightarrow A_1,\ A_3 \Rightarrow\Rightarrow A_1A_2,\ A_3 \Rightarrow\Rightarrow A_5,\ A_4 \Rightarrow\Rightarrow A_1A_2 \}$ is a set of BMVDs and $\mathbf{c}$ is the BAJD $\otimes\{A_1A_2, A_1A_3, A_3A_4, A_3A_5\}$. We initially construct $\mathbf{T}_{\mathcal{R}}$ according to the BAJD $\mathbf{c}$ as shown in Figure 33. The row $< a_1\ a_2\ a_3\ a_4\ a_5\ p(a_1\ a_2\ a_3\ a_4\ a_5) >$ of all distinguished variables can be constructed as follows. Since rows $\mathbf{w}_1 = < a_1\ a_2\ b_1\ b_2\ b_3\ p(a_1a_2b_1b_2b_3) >$ and $\mathbf{w}_2 = < a_1\ b_4\ a_3\ b_5\ b_6\ p(a_1b_4a_3b_5b_6) >$ are joinable on A_1, we can apply the MJ-rule corresponding to the BMVD $A_1 \Rightarrow\Rightarrow A_2$ in $\mathbf{C}$ to rows $\mathbf{w}_1$ and $\mathbf{w}_2$ to obtain the new row $\mathbf{w}_5 = < a_1\ a_2\ a_3\ b_5\ b_6\ p(a_1a_2a_3b_5b_6) >$, where $p(a_1a_2a_3b_5b_6)$ is defined as

$$p(a_1a_2a_3b_5b_6) = \frac{p(a_1a_2) \cdot p(a_1a_3b_5b_6)}{p(a_1)}. \tag{41}$$

Similarly, the MJ-rule corresponding to the BMVD $A_3 \Rightarrow\Rightarrow A_1A_2$ in $\mathbf{C}$ can be applied to joinable on A_3 rows $\mathbf{w}_5 = < a_1\ a_2\ a_3\ b_5\ b_6\ p(a_1a_2a_3b_5b_6) >$ and $\mathbf{w}_3 = < b_7\ b_8\ a_3\ a_4\ b_9\ p(b_7b_8a_3a_4b_9) >$ to generate the new row $\mathbf{w}_6 = < a_1\ a_2\ a_3\ a_4\ b_9\ p(a_1a_2a_3a_4b_9) >$, where

$$p(a_1a_2a_3a_4b_9) = \frac{p(a_1a_2a_3) \cdot p(a_3a_4b_9)}{p(a_3)}. \tag{42}$$

As shown in Figure 34, the row $\mathbf{w}_7 = < a_1\ a_2\ a_3\ a_4\ a_5\ p(a_1a_2a_3a_4a_5) >$ of all distinguished variables can be generated by applying the MJ-rule corresponding to the BMVD $A_3 \Rightarrow\Rightarrow A_5$ to joinable rows $\mathbf{w}_6 = < a_1\ a_2\ a_3\ a_4\ b_9\ p(a_1a_2a_3a_4b_9) >$ and $\mathbf{w}_4 = < b_{10}\ b_{11}\ a_3\ b_{12}\ a_5\ p(b_{10}b_{11}a_3b_{12}a_5) >$, where $p(a_1a_2a_3a_4a_5)$ is factorized as

$$p(a_1a_2a_3a_4a_5) = \frac{p(a_1a_2a_3a_4) \cdot p(a_3a_5)}{p(a_3)}. \tag{43}$$

Clearly, the factorization of $p(a_1a_2a_3a_4a_5)$ in Equation (43) is not in the form required by the BAJD $\mathbf{c}$ being tested:

$$p(a_1a_2a_3a_4a_5) = \frac{p(a_1a_2) \cdot p(a_1a_3) \cdot p(a_3a_4) \cdot p(a_3a_5)}{p(a_1) \cdot p(a_3) \cdot p(a_3)}. \tag{44}$$

The important point to realize is that we can always factorize $p(a_1a_2a_3a_4a_5)$ according to Equation (44). From Equation (42), we obtain:

$$p(a_1a_2a_3a_4) = \frac{p(a_1a_2a_3) \cdot p(a_3a_4)}{p(a_3)}. \tag{45}$$

By substituting Equation (45) into Equation (43), we obtain:

$$p(a_1a_2a_3a_4a_5) = \frac{p(a_1a_2a_3) \cdot p(a_3a_4) \cdot p(a_3a_5)}{p(a_3) \cdot p(a_3)}. \tag{46}$$

Similarly, Equation (41) indicates that:

$$p(a_1a_2a_3) = \frac{p(a_1a_2) \cdot p(a_1a_3)}{p(a_1)}. \tag{47}$$

A_1	A_2	A_3	A_4	A_5	A_p
a_1	a_2	b_1	b_2	b_3	$p(a_1a_2b_1b_2b_3)$
a_1	b_4	a_3	b_5	b_6	$p(a_1b_4a_3b_5b_6)$
b_7	b_8	a_3	a_4	b_9	$p(b_7b_8a_3a_4b_9)$
b_{10}	b_{11}	a_3	b_{12}	a_5	$p(b_{10}b_{11}a_3b_{12}a_5)$

Figure 33: The initial tableau $\mathbf{T}_{\mathcal{R}}$ constructed according to the BAJD $\mathbf{c} = \otimes\{A_1A_2,\ A_1A_3,\ A_3A_4,\ A_3A_5\}$.

	A_1	A_2	A_3	A_4	A_5	A_p
$\mathbf{w}_1$	a_1	a_2	b_1	b_2	b_3	$p(a_1a_2b_1b_2b_3)$
$\mathbf{w}_2$	a_1	b_4	a_3	b_5	b_6	$p(a_1b_4a_3b_5b_6)$
$\mathbf{w}_3$	b_7	b_8	a_3	a_4	b_9	$p(b_7b_8a_3a_4b_9)$
$\mathbf{w}_4$	b_{10}	b_{11}	a_3	b_{12}	a_5	$p(b_{10}b_{11}a_3b_{12}a_5)$
$\mathbf{w}_5$	a_1	a_2	a_3	b_5	b_6	$\frac{p(a_1a_2)\cdot p(a_1a_3b_5b_6)}{p(a_1)}$
$\mathbf{w}_6$	a_1	a_2	a_3	a_4	b_9	$\frac{p(a_1a_2a_3)\cdot p(a_3a_4b_9)}{p(a_3)}$
$\mathbf{w}_7$	a_1	a_2	a_3	a_4	a_5	$\frac{p(a_1a_2a_3a_4)\cdot p(a_3a_5)}{p(a_3)} = \frac{p(a_1a_2)\cdot p(a_1a_3)\cdot p(a_3a_4)\cdot p(a_3a_5)}{p(a_1)\cdot p(a_3)\cdot p(a_3)}$

Figure 34: If the row $< a_1\ a_2\ a_3\ a_4\ a_5\ p(a_1a_2a_3a_4a_5) >$ of all distinguished variables is generated, then $p(a_1a_2a_3a_4a_5)$ can always be factorized according to the BAJD $\mathbf{c}$ being tested.

Substituting Equation (47) into Equation (46), we obtain our desired factorization:

$$p(a_1a_2a_3a_4a_5) = \frac{p(a_1a_2) \cdot p(a_1a_3) \cdot p(a_3a_4) \cdot p(a_3a_5)}{p(a_1) \cdot p(a_3) \cdot p(a_3)}. \square \quad (48)$$

Example 23 demonstrates that we can always factorize the probability value of the row of all distinguished variables according to the BAJD $\mathbf{c}$ being tested. This means that if the row of all distinguished variables appears in $chase_{\mathbf{C}}(\mathbf{T_c})$, then $\mathbf{C} \models \mathbf{c}$.

As promised, we now show that developing a probabilistic chase algorithm for the Bayesian network model is not necessary because of the intrinsic relationship between the Bayesian and relational database models.

Theorem 7 Let $\mathbf{C}$ be a set of BMVDs on $R = A_1A_2 \ldots A_m$, and $\mathbf{c}$ be a single BMVD on R. Then

$$\mathbf{C} \models \mathbf{c} \iff < a_1\ a_2 \ldots a_m > \text{ is a row in } chase_C(T_{\mathcal{R}}),$$

where $C = \{X \rightarrow\rightarrow Y \mid X \Rightarrow\Rightarrow Y \in \mathbf{C}\}$ is the set of MVDs corresponding to $\mathbf{C}$, c is the MVD corresponding to $\mathbf{c}$, and $chase_C(T_{\mathcal{R}})$ is the result of chasing c under C.

Proof: By Theorem 5,

$$\mathbf{C} \models \mathbf{c} \iff C \models c.$$

By Theorem 3,

$$C \models c \iff < a_1\ a_2 \ldots a_m > \text{ is a row in } chase_C(T_{\mathcal{R}}).$$

The claim follows immediately. $\square$

Theorem 7 indicates that the standard chase algorithm, developed for testing the implication of *data* dependencies, can in fact be used to test the implication of nonembedded probabilistic conditional independency.

4.3 Conflict-free Nonembedded Dependency

In this section, we examine the pair (conflict-free BMVD, conflict-free MVD). Recall that conflict-free BMVD is a subclass within the BMVD class. Similarly, conflict-free MVD is a subclass of MVD. Since we have already shown that the implication problems coincide in the pair (BMVD, MVD), obviously the implication problems coincide in the pair (conflict-free BMVD, conflict-free MVD) as mentioned in [26]. However, it is worthwhile to study these special classes since they exhibit many desirable properties in practical applications. We begin the class of conflict-free MVDs in relational databases.

As mentioned in Section 2.2, a set of MVDs is called *conflict-free* if it can be faithfully represented in a single acyclic hypergraph. In fact, a conflict-free set C of MVDs has an

unique acyclic hypergraph $\mathcal{R}$ which is a perfect-map. That is, every MVD logically implied by C can be inferred from the acyclic hypergraph $\mathcal{R}$ using the separation method, and every MVD inferred from $\mathcal{R}$ is logically implied by C. As mentioned in Section 2.2, a conflict-free set C of MVDs is equivalent to the *acyclic join dependency* (AJD) $\bowtie \mathcal{R}$. Whenever any relation satisfies all the MVDs in C, then it also satisfies the AJD $\bowtie \mathcal{R}$, and whenever any relation satisfies the AJD $\bowtie \mathcal{R}$, then it also satisfies all the MVDs in C. The important point is that the special class of conflict-free MVDs exhibits many desirable properties in database applications [4].

Since the class of conflict-free MVD plays a crucial role in the design and implementation of relational databases, we would like to take this opportunity to introduce the class of *conflict-free* BMVD in our Bayesian database model. We call a set $\mathbf{C}$ of BMVDs *conflict-free* if $\mathbf{C}$ has an acyclic hypergraph $\mathcal{R}$ which is a perfect-map. That is, every BMVD logically implied by $\mathbf{C}$ can be inferred from the acyclic hypergraph $\mathcal{R}$ using the separation method, and every BMVD inferred from $\mathcal{R}$ is logically implied by $\mathbf{C}$.

The first desirable property of the class of conflict-free BMVD is that it has a favourable graphical property by definition. Obviously, there are some sets of nonembedded probabilistic conditional independencies that do not enjoy this luxury. For example, the set $\mathbf{C}$ of BMVDs in Example 13 cannot be faithfully represented by a single acyclic hypergraph.

The second desirable property of the class of conflict-free BMVDs is that every set of conflict-free BMVDs is equivalently characterized by a new dependency, called *Bayesian acyclic-join dependency* (BAJD), defined in Section 2.5. (The notion of BAJD corresponds to that of AJD in relational databases.) Whenever any probabilistic relation satisfies all the BMVDs in $\mathbf{C}$, then it also satisfies the BAJD $\otimes\mathcal{R}$, and whenever any probabilistic relation satisfies the BAJD $\otimes\mathcal{R}$, then it also satisfies all the BMVDs in $\mathbf{C}$.

The third property of the conflict-free BMVD class is that a Markov network can be equivalently states as a joint probability distribution satisfying a BAJD. That is, a joint distribution is written in terms of marginal distributions defined over an acyclic hypergraph. For example, if $\mathbf{r}(A_1A_2A_3A_4A_5A_6)$ is the probabilistic relation representing the joint probability distribution $p(A_1, A_2, A_3, A_4, A_5, A_6)$ in Equation (19), then relation $\mathbf{r}(A_1A_2A_3A_4A_5A_6)$ satisfies the BAJD $\otimes\mathcal{R}$, where $\mathcal{R}$ is the acyclic hypergraph in Figure 2.

Theorem 8 Let $\mathbf{C}$ denote a conflict-free set of BMVDs. Let $C = \{X \rightarrow\rightarrow Y \mid X \Rightarrow\Rightarrow Y \in \mathbf{C}\}$ be the conflict-free set of MVDs corresponding to $\mathbf{C}$. Then $\mathbf{C}$ and C have the same perfect-map $\mathcal{R}$.

Proof: The same separation method is used to infer both BMVDs and MVDs from acyclic hypergraphs. Therefore, for any given acyclic hypergraph $\mathcal{R}$, the BMVD $X \Rightarrow\Rightarrow Y$ can be inferred from $\mathcal{R}$ if and only if the corresponding MVD $X \rightarrow\rightarrow Y$ can be inferred from $\mathcal{R}$. Let $\mathcal{R}_1$ be the acyclic hypergraph which is a perfect-map of the conflict-free set $\mathbf{C}$ of BMVDs. Let $\mathcal{R}_2$ the perfect-map of C. We need to show that $\mathcal{R}_1$ and $\mathcal{R}_2$ denote the same acyclic hypergraph. Since a conflict-free set of MVDs has a unique perfect-map, it suffices to show that $\mathcal{R}_1$ is a perfect-map of the set C of MVDs.

Suppose $C \models X \rightarrow\rightarrow Y$. By Theorem 5, $\mathbf{C} \models \mathbf{c}$ if and only if $C \models c$. Thus, $\mathbf{C} \models X \Rightarrow\Rightarrow Y$. Since $\mathcal{R}_1$ is a perfect-map of $\mathbf{C}$, $X \Rightarrow\Rightarrow Y$ can be inferred from $\mathcal{R}_1$ using the separation

method. By the above observation, this means that the MVD $X \rightarrow\rightarrow Y$ can be inferred from $\mathcal{R}_1$.

Suppose the MVD can be inferred from $\mathcal{R}_1$ using the separation method. By the above observation, this means that the BMVD $X \Rightarrow\Rightarrow Y$ can be inferred from $\mathcal{R}_1$. Since $\mathcal{R}_1$ is a perfect-map of $\mathbf{C}$, $\mathbf{C} \models X \Rightarrow\Rightarrow Y$. By Theorem 5, this implies that $C \models X \rightarrow\rightarrow Y$. □

The special classes in the pair (conflict-free BMVD, conflict-free MVD) both have favourable graphical structures. Given corresponding conflict-free sets of nonembedded probabilistic conditional independencies and MVDs, Theorem 8 indicates that the acyclic hypergraph used to define the Markov network in Bayesian database model is the same one used to define the AJD in the relational database model.

5 Embedded Dependencies

In this section, we examine the implication problem for the two pairs of *embedded* dependencies, namely, (conflict-free BEMVD, conflict-free EMVD) and (BEMVD, EMVD). As shown in Figure 1, the class of conflict-free BEMVD is a subclass of BEMVD, and conflict-free EMVD is a subclass of EMVD. We choose to first discuss the implication problem for the pair (conflict-free BEMVD, conflict-free EMVD) since the implication problems for these two classes are *solvable*. We then conclude our discussion by looking at the implication problem for the pair (BEMVD, EMVD), namely, the general classes of probabilistic conditional independency and EMVD.

5.1 Conflict-free Embedded Dependencies

Here we study the implication problem for the pair (conflict-free BEMVD, conflict-free EMVD). We begin the conflict-free BEMVD class.

The class of conflict-free BEMVDs plays a key role in the design of Bayesian networks. A set of BEMVDs is called *conflict-free* if and only if they can be faithfully represented by a single directed acyclic graph (DAG). The *d-separation* method [31] is used to infer BEMVDs from a DAG. Thus, one desirable property of the conflict-free BEMVD class is that every conflict-free set of BEMVDs has a DAG which is a *perfect-map*.

A set of conflict-free BEMVDs can be characterized as another dependency, called *Bayesian embedded acyclic join dependency* (BEAJD). We say a relation $\mathbf{r}(R)$ satisfies the BEAJD, written $\times D$, if $\mathbf{r}(R)$ can be expressed as:

$$\mathbf{r}(R) \quad = \quad \mathbf{r}(A_1) \times \mathbf{r}(A_2|pa(A_2)) \times \mathbf{r}(A_3|pa(A_3)) \times \ldots \mathbf{r}(A_m|pa(A_m)),$$

where D is a DAG, $pa(A_m)$ is the parent set of A_2 in D, and $\mathbf{r}(A_i|pa(A_i))$ is the probabilistic relation representing the *conditional probability distribution* $p(A_i|pa(A_i))$. It should be clear that BEAJD in our terminology represents a *Bayesian network* [31].

Example 24 Let $\mathbf{r}(A_1A_2A_3A_4A_5A_6)$ be the relation representing the joint probability distribution $p(A_1, A_2, A_3, A_4, A_5, A_6)$ in Example 10. Thus, $\mathbf{r}$ satisfies the BEAJD $\times D$, where

D is the DAG in Figure 10. That is, $\mathbf{r}(A_1A_2A_3A_4A_5A_6)$ can be losslessly decomposed as:

$$\mathbf{r}(A_1A_2A_3A_4A_5A_6) \;=\; \mathbf{r}(A_1) \times \mathbf{r}(A_2|A_1) \times \mathbf{r}(A_3|A_1) \times \mathbf{r}(A_4|A_2A_3) \times \mathbf{r}(A_5|A_2A_3) \times \mathbf{r}(A_6|A_5).$$

The class of conflict-free BEMVD is a *special case* of the general BEMVD class, as shown in Figure 1. This special class of probabilistic dependencies has several desirable properties [31] including a complete axiomatization.

Theorem 9 [31] The class of *conflict-free BEMVD* has a *complete* axiomatization. Let X, Y, Z, W be pairwise disjoint subsets of R such that $XYZW = R$.

(CF-BEMVD1) If $X \Rightarrow\Rightarrow Y$ then $X \Rightarrow\Rightarrow ZW$
(CF-BEMVD2) If $X \Rightarrow\Rightarrow YW \mid Z$, then $X \Rightarrow\Rightarrow Y \mid Z$
(CF-BEMVD3) If $X \Rightarrow\Rightarrow YZ$, then $XZ \Rightarrow\Rightarrow Y$,
(CF-BEMVD4) If $X \Rightarrow\Rightarrow Y \mid Z$ and $XZ \Rightarrow\Rightarrow Y$, then $X \Rightarrow\Rightarrow Y$.

The axioms (CF-BEMVD1)-(CF-BEMVD4) are respectively called *symmetry*, *decomposition*, *weak union*, and *contraction*. Theorem 9 indicates that the implication problem for the conflict-free BEMVD class is solvable.

We now turn our attention to the other class of dependency in the pair (conflict-free BEMVD, conflict-free EMVD), namely, conflict-free EMVD. In order to solve the implication problem for the class of *conflict-free* EMVD, we again use the method of drawing a one-to-one correspondence between the class of conflict-free BEMVD and the class of conflict-free EMVD.

We can easily demonstrate that the following EMVD inference axioms are sound, where X, Y, Z, W be pairwise disjoint subsets of R such that $XYZW = R$.

(CF-EMVD1) If $X \rightarrow\rightarrow Y$ then $X \rightarrow\rightarrow ZW$
(CF-EMVD2) If $X \rightarrow\rightarrow YW \mid Z$, then $X \rightarrow\rightarrow Y \mid Z$
(CF-EMVD3) If $X \rightarrow\rightarrow YZ$, then $XZ \rightarrow\rightarrow Y$,
(CF-EMVD4) If $X \rightarrow\rightarrow Y \mid Z$ and $XZ \rightarrow\rightarrow Y$, then $X \rightarrow\rightarrow Y$.

Axioms (CF-EMVD1)-(CF-EMVD3) are well-known properties of EMVDs [3]. As an example, we show (CF-EMVD4) is sound. Let $r(XYZW)$ be a relation. Suppose there exists two tuples t_1 and t_2 in $r(XYZW)$ such that $t_1(X) = t_2(X)$. By the EMVD $X \rightarrow\rightarrow Y \mid Z$, there exists a tuple t_3 in $r(XYZW)$ such that

$$t_3(XY) \;=\; t_1(XY) \quad \text{and} \quad t_3(XZ) \;=\; t_2(XZ),$$

and there is no restriction on $t_3(W)$. By the MVD $XZ \rightarrow\rightarrow Y$, $t_3(XZ) = t_2(XZ)$ implies that there exists a t_4 in $r(XYZW)$ such that

$$t_4(XYZ) \;=\; t_3(XYZ) \quad \text{and} \quad t_4(XZW) \;=\; t_2(XZW).$$

To show that the MVD $X \rightarrow\rightarrow Y$ holds in $r(XYZW)$, we seek a tuple t such that

$$t(XY) \;=\; t_1(XY) \quad \text{and} \quad t(XZW) \;=\; t_2(XZW).$$

Tuple t_4 is the desired tuple t since $t_4(XY) = t_3(XY) = t_1(XY)$ and $t_4(XZW) = t_2(XZW)$. □

Theorem 10 Given the *complete* axiomatization (CF-BEMVD1)-(CF-BEMVD4) for the CF-BEMVD class. Then

$$\mathbf{C} \models \mathbf{c} \;\;\Longrightarrow\;\; C \models c,$$

where $\mathbf{C}$ is a conflict-free set of BEMVDs, $C = \{X \rightarrow\rightarrow Y|Z \;\mid\; X \Rightarrow\Rightarrow Y|Z \in \mathbf{C}\}$ is the corresponding conflict-free set of EMVDs, and c is the EMVD corresponding to a BEMVD $\mathbf{c}$.

Proof: Suppose that $\mathbf{C} \models \mathbf{c}$. By Theorem 9, $\mathbf{C} \models \mathbf{c}$ implies that $\mathbf{C} \vdash \mathbf{c}$. That is, there exists a derivation sequence $\mathbf{s}$ of the BEMVD $\mathbf{c}$ from the conflict-free set $\mathbf{C}$ of BEMVDs. The above discussion demonstrates that there are *sound* axioms (CF-EMVD1)-(CF-EMVD4) corresponding to the axioms (CF-BEMVD1)-(CF-BEMVD4). This implies that there is a derivation sequence s of the EMVD c from the conflict-free set C of EMVDs, such that s parallels $\mathbf{s}$. That is, $C \vdash c$. Since axioms (CF-EMVD1)-(CF-EMVD4) are sound, $C \vdash c$ implies that $C \models c$. □

Theorem 10 indicates that

$$\mathbf{C} \models \mathbf{c} \;\;\Longrightarrow\;\; C \models c,$$

holds in the pair (conflict-free BEMVD, conflict-free EMVD). We now consider whether

$$\mathbf{C} \models \mathbf{c} \;\;\Longleftarrow\;\; C \models c,$$

is also true in this pair of dependencies. The following result is useful to answer this question.

Theorem 11 [31] The axioms (CF-EMVD1)-(CF-EMVD4) are *complete* for the class of *conflict-free* EMVD.

Theorem 12 Given the complete axiomatization (CF-EMVD1)-(CF-EMVD4) for the CF-EMVD class. Then

$$\mathbf{C} \models \mathbf{c} \;\;\Longleftarrow\;\; C \models c,$$

where C is a conflict-free set of BEMVDs, $\mathbf{C} = \{X \Rightarrow\Rightarrow Y|Z \;\mid\; X \rightarrow\rightarrow Y|Z \in C\}$ is the corresponding conflict-free set of EMVDs, and $\mathbf{c}$ is the BEMVD corresponding to a EMVD c.

Proof: Can be shown using a similar argument to the one given in the proof of Theorem 10. □

The important point to remember is that Theorems 10 and 12 indicate that

$$\mathbf{C} \models \mathbf{c} \iff C \models c \tag{49}$$

holds in the pair (conflict-free BEMVD, conflict-free EMVD). As already mentioned, the class of conflict-free BEMVDs is used to construct Bayesian networks. On the other hand, however, the entire class of EMVD has traditionally been ignored in the design and implementation of relational databases. The above result is useful since it implies that it may be advantageous to utilize the special class of *conflict-free* EMVD.

5.2 Embedded Dependencies in General

The last pair of dependencies we study is (BEMVD, EMVD). All of the previously studied classes of probabilistic dependencies are a subclass of BEMVD (probabilistic conditional independency). A similar remark holds for EMVD. Before we study the implication problem $\mathbf{C} \models \mathbf{c}$ for probabilistic conditional independency, we first examine the implication problem $C \models c$ for the general class of EMVD. Unfortunately, it is not possible to solve the implication problem for the EMVD class using axiomatization.

Theorem 13 [29, 34] The EMVD class does not have a *finite* complete axiomatization.

The chase algorithm also does *not* solve the implication problem for the EMVD class. By definition, a J-rule for an EMVD $X \rightarrow\rightarrow Y|Z$ in a given set C of EMVDs would only generate a *partial* new row. To modify the chase algorithm for EMVDs, the partial row is padded out with *unique* nondistinguished variables in the remaining attributes. This is precisely the reason why the chase does not work for EMVDs. In using an EMVD, the chase adds a new row containing new symbols. This enables further applications of EMVDs in C, which will add more new rows with new symbols, and this process can continue forever. (With MVDs, on the other hand, a new row consists only of existing symbols meaning that eventually there are no new rows to generate.)

The chase algorithm, however, is a *proof procedure* for implication of EMVDs [13]. Given C and c, if $C \models c$, then the row of all distinguished variables will eventually be generated. The generation of the row of all a's can be used as a stopping criterion.

Example 25 Suppose we wish to verify that $C \models c$, where C and c are defined as:

$$\{A_1 \rightarrow\rightarrow A_3 \mid A_4,\ A_2 \rightarrow\rightarrow A_3 \mid A_4,\ A_3A_4 \rightarrow\rightarrow A_1 \mid A_2\} \models A_1A_2 \rightarrow\rightarrow A_3. \tag{50}$$

The initial tableau $T_{\mathcal{R}}$ is constructed according to c as shown in Figure 35 (left). We can apply the J-rule corresponding to the EMVD $A_1 \rightarrow\rightarrow A_3 \mid A_4$ in C to joinable rows $w_1 = < a_1\ a_2\ a_3\ b_1 >$ and $w_2 = < a_1\ a_2\ b_2\ a_4 >$ to generate the new row $w_3 = < a_1\ b_3\ a_3\ a_4 >$ as shown in Figure 35 (right). Similarly, we can apply the J-rule corresponding to the EMVD $A_2 \rightarrow\rightarrow A_3 \mid A_4$ in C to joinable rows $w_1 = < a_1\ a_2\ a_3\ b_1 >$ and $w_2 = < a_1\ a_2\ b_2\ a_4 >$ to generate the new row $w_4 = < b_4\ a_2\ a_3\ a_4 >$ as shown in Figure 35 (right). Finally, we

$T_{\mathcal{R}} =$

A_1	A_2	A_3	A_4
a_1	a_2	a_3	b_1
a_1	a_2	b_2	a_4

	A_1	A_2	A_3	A_4
w_1	a_1	a_2	a_3	b_1
w_2	a_1	a_2	b_2	a_4
w_3	a_1	b_3	a_3	a_4
w_4	b_4	a_2	a_3	a_4
w_5	a_1	a_2	a_3	a_4

Figure 35: On the left, the initial tableau $T_{\mathcal{R}}$ constructed according to the EMVD c defined as $A_1A_2 \rightarrow\rightarrow A_3$. The row $< a_1\ a_2\ a_3\ a_4 >$ of all distinguished variables appears in $chase_{\mathbf{C}}(T_{\mathcal{R}})$ indicating $C \models c$.

can obtain the row $< a_1\ a_2\ a_3\ a_4 >$ of all distinguished variables by applying the J-rule corresponding to the MVD $A_3A_4 \rightarrow\rightarrow A_1 \mid A_2$ in C to joinable rows w_3 and w_4. Therefore, $C \models c$. □

For over a decade, a tremendous amount of effort was put forth in the database community to show that the implication problem for EMVDs is in fact *unsolvable*. Herrmann [18] recently succeeded in showing this elusive result.

Theorem 14 [18] The implication problem for the EMVD class is *unsolvable*.

Theorem 14 is important since it indicates that *no* method exists for deciding the implication problem for the EMVD class. This concludes our discussion on the EMVD class.

We now study the corresponding class of probabilistic dependencies in the pair (BEMVD, EMVD), namely, the general class of probabilistic conditional independency. Pearl [31] conjectured that the semi-graphoid axioms (CF-BEMVD1)-(CF-BEMVD4) could solve the implication problem for probabilistic conditional independency (BEMVD) in general. This conjecture has been refuted.

Theorem 15 [37, 45] BEMVDs do not have a *finite* complete axiomatization.

Theorem 15 indicates that it is not possible to solve the implication problem for the BEMVD class using a finite axiomatization. This result does not rule out the possibility that some alternative method exists for solving this implication problem. The following result, however, says no such method exists.

Theorem 16 The implication problem for the BEMVD class is *unsolvable*.

The above Theorem can be proven in a similar fashion as to the one given by Herrmann [18] for the EMVD class. The proof is quite lengthy and will be shown in a more complete paper.

Like Theorem 14, Theorem 16 is important since it indicates that *no* method exists to solve the implication problem for probabilistic conditional independency in general.

As with the other classes or probabilistic dependencies, we now examine the relationship between $\mathbf{C} \models \mathbf{c}$ and $C \models c$ in the pair (BEMVD,EMVD). The following two examples [37] indicate that the implication problems for EMVD and BEMVD do not coincide.

$r(A_1A_2A_3A_4)$ =

A_1	A_2	A_3	A_4
0	0	0	0
0	0	0	1
0	1	0	0
1	0	0	0
1	1	0	0
1	1	1	0

Figure 36: Relation r satisfies all of the EMVDs in C but does not the EMVD c, where C and c are defined in Example 26. Therefore, $C \not\models c$.

Example 26 Consider the set $\mathbf{C} = \{A_3A_4 \Rightarrow\Rightarrow A_1|A_2,\ A_1 \Rightarrow\Rightarrow A_3|A_4,\ A_2 \Rightarrow\Rightarrow A_3|A_4,\ \emptyset \Rightarrow\Rightarrow A_1|A_2\}$ of BEMVDs, and $\mathbf{c}$ the single BEMVD $\emptyset \Rightarrow\Rightarrow A_3|A_4$. In [36], Studeny proved that $\mathbf{C} \models \mathbf{c}$. Now consider the set $C = \{X \rightarrow\rightarrow Y|Z \mid X \Rightarrow\Rightarrow Y|Z \in \mathbf{C}\}$ of EMVDs corresponding to the set $\mathbf{C}$ of BEMVDs, and the single EMVD $\emptyset \rightarrow\rightarrow A_3|A_4$ corresponding to the BEMVD $\mathbf{c}$. Consider the relation $r(A_1A_2A_3A_4)$ in Figure 36. It can be verified that $r(A_1A_2A_3A_4)$ satisfies all of the EMVDs in C but does not satisfy the EMVD c. Thus, $C \not\models c$. □

Example (26) indicates that

$$\mathbf{C} \models \mathbf{c} \quad \not\Longrightarrow \quad C \models c. \tag{51}$$

Example 27 Consider the set $C = \{A_1 \rightarrow\rightarrow A_3 \mid A_4,\ A_2 \rightarrow\rightarrow A_3 \mid A_4,\ A_3A_4 \rightarrow\rightarrow A_1 \mid A_2\}$ of EMVDs, and let c be the single EMVD $A_1A_2 \rightarrow\rightarrow A_3$. The chase algorithm was used in Example 25 to show that $C \models c$. Now consider the corresponding set of BEMVDs $\mathbf{C} = \{A_1 \Rightarrow\Rightarrow A_3 \mid A_4,\ A_2 \Rightarrow\Rightarrow A_3 \mid A_4,\ A_3A_4 \Rightarrow\Rightarrow A_1 \mid A_2\}$ and $\mathbf{c}$ is the BMVD $A_1A_2 \Rightarrow\Rightarrow A_3$. It is easily verified that relation $\mathbf{r}(A_1A_2A_3A_4)$ in Figure 37 satisfies all of the BEMVDs in $\mathbf{C}$ but does not satisfy the BEMVD $\mathbf{c}$. Therefore, $\mathbf{C} \not\models \mathbf{c}$. □

Example 27 indicates that

$$\mathbf{C} \models \mathbf{c} \quad \not\Longleftarrow \quad C \models c. \tag{52}$$

In the next section, we attempt to answer why the implication problems coincide for some classes but not for others.

5.3 The Role of Solvability

We have shown that

$\mathbf{C} \models \mathbf{c} \iff C \models c$ for the pair (BMVD, MVD) in Theorem 5,
$\mathbf{C} \models \mathbf{c} \iff C \models c$ for the pair (Conflict-free BMVD, Conflict-free MVD) in Theorem 5,
$\mathbf{C} \models \mathbf{c} \iff C \models c$ for the pair (Conflict-free BEMVD, Conflict-free EMVD) in Equation 49.

$\mathbf{r}(A_1A_2A_3A_4)$ =

A_1	A_2	A_3	A_4	A_p
0	0	0	0	0.2
0	0	0	1	0.2
0	0	1	0	0.2
0	0	1	1	0.1
0	1	1	1	0.1
1	0	1	1	0.1
1	1	1	1	0.1

Figure 37: Relation $\mathbf{r}$ satisfies all of the BEMVDs in $\mathbf{C}$ but does not the BEMVD $\mathbf{c}$, where $\mathbf{C}$ and $\mathbf{c}$ are defined in Example 27. Therefore, $\mathbf{C} \not\models \mathbf{c}$.

That is, the implication problems coincide in these three pairs of classes. However, Examples (26) and (27) demonstrate that

$$\mathbf{C} \models \mathbf{c} \not\Longleftrightarrow C \models c \qquad \text{for the pair (BEMVD, EMVD).}$$

The main difference between the first three pairs of classes and the last pair is that the implication problems for the former are *solvable*, whereas for the latter they are *unsolvable*. These observations lead us to make the following conjecture.

Conjecture 1 Consider any pair (BD-class, RD-class), where BD-class is a class of probabilistic dependencies in the Bayesian database model and RD-class is the corresponding class of data dependencies in the relational database model. Let $\mathbf{C}$ be a set of probabilistic dependencies chosen from BD-class, and $\mathbf{c}$ a single dependency in BD-class. Let C and c denote the corresponding set of data dependencies of $\mathbf{C}$ and $\mathbf{c}$, respectively, in RD-class.

(i) If the implication problem is *solvable* for the class BD-class, then

$$\mathbf{C} \models \mathbf{c} \quad \Longrightarrow \quad C \models c.$$

(ii) If the implication problem is *solvable* for the class RD-class, then

$$\mathbf{C} \models \mathbf{c} \quad \Longleftarrow \quad C \models c.$$

In [37], Studeny studied the relationship between the implication problem for probabilistic conditional independency (BEMVD) and embedded multivalued dependency (EMVD). Based on Conjecture (i), his observation that:

$$\mathbf{C} \models \mathbf{c} \quad \not\Longrightarrow \quad C \models c,$$

would indicate that the implication problem for the general class of probabilistic conditional independency is *unsolvable*. Similarly, based on Conjecture (ii), his observation that:

$$\mathbf{C} \models \mathbf{c} \quad \not\Longleftarrow \quad C \models c,$$

would indicate that the implication problem for the class of EMVD is *unsolvable*.

A successful proof of this conjecture would provide an alternative proof that EMVD and BEMVD (probabilistic conditional independency) are both unsolvable.

6 Conclusion

The results of this paper and our previous work [41, 43, 44] clearly indicate that there is a *direct* correspondence between the notions used in the Bayesian database model and the relational database model. The notions of distribution, multiplication, and marginalization in Bayesian networks are *generalizations* of relation, natural join, and projection in relational databases. Both models use *nonembedded* dependencies in practice, i.e., the Markov network and acyclic join dependency representations are both defined over the classes of nonembedded dependencies. The same conclusions have been reached regarding *query processing* in acyclic hypergraphs [4, 20, 35], and as to whether a set of *pairwise consistent* distributions (relations) are indeed marginal distributions from the same joint probability distribution [4, 11]. Even the recent attempts to generalize the standard Bayesian database model, including *horizontal independencies* [7, 43], *complex-values* [21, 43], and *distributed* Bayesian networks [8, 42, 46], parallel the development of *horizontal dependencies* [12], *complex-values* [1, 19], and *distributed* databases [9] in the relational database model. More importantly, the implication problem for both models coincide with respect to two important classes of independencies, the BMVD class [14] (used in the construction of Markov networks) and the conflict-free sets [31] (used in the construction of Bayesian networks).

Initially, we were quite surprised by the suggestion [37] that the Bayesian database model and the relational database model are *different*. However, our study reveals that this observation [37] was based on the analysis of the BEMVD class of probabilistic conditional independencies. The implication problem for this general BEMVD class of embedded independencies is *unsolvable*, as is the EMVD class of embedded multivalued dependencies in relational databases [5]. Obviously, only *solvable* classes of independencies are useful for the representation of and reasoning with probabilistic knowledge. We therefore maintain that there is no *real* difference between the Bayesian database model and the relational database model in a *practical* sense. In fact, there exists an *inherent* relationship between these two knowledge systems. We conclude the present discussion by making the following conjecture:

Conjecture 2 The Bayesian database model generalizes the relational database model on *all* solvable classes of dependencies.

This conjecture is illustrated in Figure 38. The truth of this conjecture would formally establish the claim that the Bayesian database model and the relational database model are the *same* in practical terms; they differ only in unsolvable classes of dependencies.

References

[1] S. Abiteboul, P. Fischer, and H. Schek. *Nested Relations and Complex Objects in Databases*, volume 361. Springer-Verlag, 1989.

[2] W.W. Armstrong. Dependency structures of database relationships. In *Proceedings of IFIP 74*, pages 580–583, Amsterdam, 1974. North Holland.

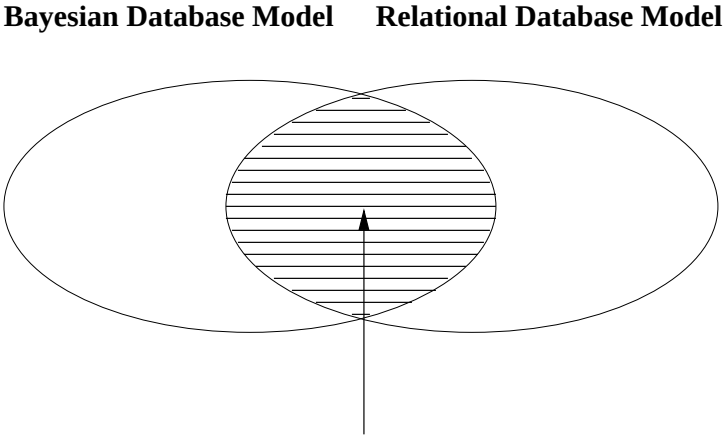

Figure 38: The Bayesian database model is a generalization of the relational database model with respect to all *solvable* classes of dependencies.

[3] C. Beeri, R. Fagin, and J.H. Howard. A complete axiomatization for functional and multivalued dependencies in database relations. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 47–61, 1977.

[4] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis. On the desirability of acyclic database schemes. *Journal of the ACM*, 30(3):479–513, July 1983.

[5] C. Beeri and M. Vardi. Formal systems for tuple and equality generating dependencies. *SIAM Journal on Computing*, 13(10):76–98, 1984.

[6] C. Berge. *Graphs and Hypergraphs*. North-Holland, Amsterdam, 1976.

[7] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in bayesian networks. In *Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 115–123. Morgan Kaufmann Publishers, 1996.

[8] C.J. Butz and S.K.M. Wong. Recovery protocols in multi-agent probabilistic reasoning systems. In *International Database Engineering and Applications Symposium*, pages 302–310. IEEE Press, 1999.

[9] Stefano Ceri and Guiseppe Pelagatti. *Distributed Databases: Principles & Systems*. McGraw-Hill, 1984.

[10] E.F. Codd. A relational model of data for large shared data banks. *Communication of ACM*, 13(6):377–387, June 1970.

[11] A.P. Dawid and S.L. Lauritzen. Hyper markov laws in the statistical analysis of decomposable graphical models. *Ann. Stat.*, 21:1272–1317, 1993.

[12] R. Fagin. Normal forms and relational database operators. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 153–160, 1979.

[13] R. Fagin and M.Y. Vardi. The theory of data dependencies: A survey. *Mathematics of Information Processing: Proceedings of Symposia in Applied Mathematics*, 34:19–71, 1986.

[14] D. Geiger and J. Pearl. Logical and algorithmic properties of conditional independence. Technical Report R-97-II-L, University of California, 1989.

[15] D. Geiger and J. Pearl. Logical and algorithmic properties of conditional independence and graphical models. *The Annals of Statistics*, 21(4):2001–2021, 1993.

[16] D. Geiger, T. Verma, and J. Pearl. Identifying independence in bayesian networks. Technical Report R-116, University of California, 1988.

[17] P. Hajek, T. Havranek, and R. Jirousek. *Uncertain Information Processing in Expert Systems*. CRC Press, 1992.

[18] C. Herrmann. On the undecidability of implications between embedded multivalued database dependencies. *Information and Computation*, 122(2):221–235, 1995.

[19] G. Jaeschke and H.J. Schek. Remarks on the algebra on non first normal form relations. In *Proceedings of First ACM SIGACT-SIGMOD Symposium on the Principles of Database Systems*, pages 124–138, 1982.

[20] F.V. Jensen, S.L. Lauritzen, and K.G. Olesen. Bayesian updating in causal probabilistic networks by local computation. *Computational Statistics Quarterly*, 4:269–282, 1990.

[21] D. Koller and A. Pfeffer. Object-oriented bayesian networks. In *Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 302–313. Morgan Kaufmann Publishers, 1997.

[22] T.T. Lee. An information-theoretic analysis of relational databases-part i: Data dependencies and information metric. *IEEE Transactions on Software Engineering*, SE-13(10):1049–1061, 1987.

[23] D. Maier. *The Theory of Relational Databases*. Principles of Computer Science. Computer Science Press, Rockville, Maryland, 1983.

[24] D. Maier, A.O. Mendelzon, and Y. Sagiv. Testing implications of data dependencies. *ACM Transactions on Database Systems*, 4(4):455–469, 1979.

[25] F. Malvestuto. A unique formal system for binary decompositions of database relations, probability distributions and graphs. *Information Sciences*, 59:21–52, 1992.

[26] F. Malvestuto. A complete axiomatization of full acyclic join dependencies. *Information Processing Letters*, 68(3):133–139, 1998.

[27] A. Mendelzon. On axiomatizing multivalued dependencies in relational databases. *Journal of the ACM*, 26(1):37–44, 1979.

[28] R.E. Neapolitan. *Probabilistic Reasoning in Expert Systems.* Wiley, New York, 1990.

[29] D. Parker and K. Parsaye-Ghomi. Inference involving embedded multivalued dependencies and transitive dependencies. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 52–57, 1980.

[30] A. Paz. Membership algorithm for marginal independencies. Technical Report CSD-880095, University of California, 1988.

[31] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, San Francisco, California, 1988.

[32] J. Pearl, D. Geiger, and T. Verma. Conditional independence and its representations. *Kybernetica*, 25(2):33–44, 1989.

[33] J. Pearl and A. Paz. Graphoids: Graph-based logic for reasoning about relevance relations. Technical Report R-53-L, University of California, 1985.

[34] Y. Sagiv and F. Walecka. Subset dependencies and a complete result for a subclass of embedded multivalued dependencies. *Journal of the ACM*, 20(1):103–117, 1982.

[35] G. Shafer. An axiomatic study of computation in hypertrees. School of Business Working Papers 232, University of Kansas, 1991.

[36] M. Studeny. Multiinformation and the problem of characterization of conditional-independence relations. *Problems of Control and Information Theory*, 18(1):3–16, 1989.

[37] M. Studeny. Conditional independence relations have no finite complete characterization. In *Eleventh Prague Conference on Information Theory, Statistical Decision Foundation and Random Processes*, pages 377–396. Kluwer, 1990.

[38] T. Verma and J. Pearl. Causal networks: Semantics and expressiveness. In *Fourth Conference on Uncertainty in Artificial Intelligence*, pages 352–359, St. Paul, MN, 1988.

[39] W.X. Wen. From relational databases to belief networks. In *Seventh Conference on Uncertainty in Artificial Intelligence*, pages 406–413. Morgan Kaufmann Publishers, 1991.

[40] S.K.M. Wong. Testing implication of probabilistic dependencies. In *Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 545–553. Morgan Kaufmann Publishers, 1996.

[41] S.K.M. Wong. An extended relational data model for probabilistic reasoning. *Journal of Intelligent Information Systems*, 9:181–202, 1997.

[42] S.K.M. Wong and C.J. Butz. Probabilistic reasoning in a distributed multi-agent environment. In *Third International Conference on Multi-Agent Systems*, pages 341–348. IEEE Press, 1998.

[43] S.K.M. Wong and C.J. Butz. Contextual weak independence in bayesian networks. In *Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 670–679. Morgan Kaufmann Publishers, 1999.

[44] S.K.M. Wong, C.J. Butz, and Y. Xiang. A method for implementing a probabilistic model as a relational database. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 556–564. Morgan Kaufmann Publishers, 1995.

[45] S.K.M. Wong and Z.W. Wang. On axiomatization of probabilistic conditional independence. In *Tenth Conference on Uncertainty in Artificial Intelligence*, pages 591–597. Morgan Kaufmann Publishers, 1994.

[46] Y. Xiang. A probabilistic framework for cooperative multi-agent distributed interpretation and optimization of communication. *Artificial Intelligence*, 87:295–342, 1996.