On the Implication Problem for Probabilistic Conditional Independency

S. K. M. Wong, C. J. Butz, and D. Wu

***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 emphasized 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.

***Index Terms*—Bayesian networks, embedded multivalued dependency, implication problem, probabilistic conditional independence, relational databases.**

I. INTRODUCTION

PROBABILITY theory provides a rigorous foundation for the management of uncertain knowledge [16], [28], [31]. We may assume that knowledge is 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 may be 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 probabilistic conditional independencies satisfied by a particular joint distribution. To facilitate the computation of marginal distributions, it is useful in practice to transform a Bayesian network into a (decomposable) Markov network by sacrificing certain independency information. A *Markov network* [16] consists of an *acyclic hypergraph* [4], [5] and a corresponding set of marginal distributions. By definition, both Bayesian and Markov networks encode the conditional independencies in a graphical structure. A graphical structure is called a *perfect-map* [4], [31] of a given set Σ of conditional independencies, if every conditional independency logically implied by Σ can be inferred from the graphical structure, and every conditional independency that can be inferred from the graphical structure is logically implied by Σ. (We say Σ *logically implies* σ and write $\Sigma \models \sigma$, if whenever any distribution that satisfies all the conditional independencies in Σ, then the distribution also satisfies σ.) However, it is important to realize that some sets of conditional independencies do *not* have a perfect-map. That is, Bayesian and Markov networks are not constructed from arbitrary sets of conditional independencies. Instead these networks only use special subclasses of probabilistic conditional independency.

Before Bayesian networks were proposed, the *relational database model* [9], [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 [13], [21], [25], [40] have noticed similarities between relational databases and Bayesian networks. Here we advocate that a Bayesian network is indeed a generalized relational database. Our *unified* approach [42], [45] is to express the concepts used in Bayesian networks by generalizing the corresponding concepts in relational databases. The proposed *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

Manuscript received October 29, 1999; revised June 23, 2000. This paper was recommended by Associate Editor W. Pedrycz.

S. K. M. Wong and D. Wu are with the Department of Computer Science, University of Regina, Regina, SK, Canada S4S 0A2 (e-mail: wong@cs.uregina.ca).

C. J. Butz is with the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada K1N 6N5.

Publisher Item Identifier S 1083-4427(00)08798-1.

[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.

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. Moreover, 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 [13]–[15], [30], [33], [36]. [37], [41], [46]. The implication problem is to test whether a given input set Σ of independencies logically implies another independency σ. Traditionally, *axiomatization* was studied in an attempt to solve the implication problem for data and probabilistic conditional independencies. In this approach, a finite set of inference axioms are used to generate symbolic proofs for a particular independency in a manner analogous to the proof procedures in mathematical logics.

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

$(\mathbf{1a})$ BEMVD;
$(\mathbf{1b})$ Conflict-free BEMVD;
$(\mathbf{2a})$ BMVD;
$(\mathbf{2b})$ Conflict-free BMVD.

Class $(\mathbf{1a})$ is the *general* class of probabilistic conditional independencies called Bayesian embedded multivalued dependency (BEMVD) in our unified model. It is important to realize that $(\mathbf{1b})$, $(\mathbf{2a})$ and $(\mathbf{2b})$ are *special* subclasses of $(\mathbf{1a})$. Subclass $(\mathbf{2a})$ contains those probabilistic conditional independencies involving *all* variables, called Bayesian (nonembedded) multivalued dependency (BMVD) in our approach. BMVD is also known as *full* probabilistic conditional independency [26], or *fixed context* probabilistic conditional independency [13]. Thus, $(\mathbf{2a})$ is a subclass of probabilistic conditional independency since $(\mathbf{1a})$ may include a set containing the mixture of embedded and nonembedded (full) probabilistic conditional independencies, whereas $(\mathbf{2a})$ can only include sets of nonembedded (full) probabilistic conditional independencies. Nonembedded probabilistic conditional independencies are graphically represented by acyclic hypergraphs, while the mixture of embedded and nonembedded probabilistic conditional independencies are graphically represented by DAGs. However, as already mentioned, there are some sets of probabilistic conditional independencies which do *not* have a perfect-map. Thus, we use the term *conflict-free* for those sets of conditional independencies which do have a perfect-map. Consequently, class $(\mathbf{2b})$ contains those sets of nonembedded (full) probabilistic conditional independencies which can be *faithfully* represented by a *single* acyclic hypergraph. Similarly, class $(\mathbf{1b})$ contains those sets of embedded and nonembedded probabilistic conditional independencies which can be *faithfully* represented by a *single* DAG. It is important to realize that $(\mathbf{1b})$ is a special subclass of $(\mathbf{1a})$, and that $(\mathbf{2b})$ is a special subclass of $(\mathbf{2a})$ (and of course $(\mathbf{1a})$). The subclass $(\mathbf{1b})$ of conflict-free BEMVDs is important since it is used in the construction of Bayesian networks. That is, subclass $(\mathbf{1b})$ allows a human expert to indirectly specify a joint distribution as a product of conditional probability distributions. The subclass $(\mathbf{2b})$ of conflict-free BMVDs is also important since it is used in the construction of Markov networks.

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 any dependency 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:

$(1a)$ EMVD;
$(1b)$ Conflict-free EMVD;
$(2a)$ MVD;
$(2b)$ Conflict-free MVD.

as depicted in Fig. 1. Since we advocate that the Bayesian database 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

$$\mathbf{C} \models \mathbf{c} \Longleftrightarrow C \models c \tag{2}$$

holds for the individual pairs ($\mathbf{1a}$, 1a), ($\mathbf{1b}$, 1b), ($\mathbf{2a}$, 2a), and ($\mathbf{2b}$, 2b). For example, we wish to know whether proposition (2) holds for the pair (BEMVD, EMVD), where $\mathbf{C}$ is a set of BEMVDs, $\mathbf{c}$ is any BEMVD, and C and c are the *corresponding* EMVDs.

We will show that

$$\{\text{BMVDs}\} \models \mathbf{c} \Longleftrightarrow \{\text{MVDs}\} \models c$$

holds for the pair (BMVD, BMVD). Since (conflict-free BMVD, conflict-free MVD) are special classes of (BMVD, BMVD), respectively, proposition (2) is obviously true for the pair ($\mathbf{2b}$, 2b), namely:

$$\{\text{CF BMVDs}\} \models \mathbf{c} \Longleftrightarrow \{\text{CF MVDs}\} \models c$$

where CF stands for *conflict-free*. It can also be shown that

$$\{\text{CF BEMVDs}\} \models \mathbf{c} \Longleftrightarrow \{\text{CF EMVDs}\} \models c$$

holds for the pair (conflict-free BEMVD, conflict-free EMVD). 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 independency and embedded multivalued dependency. In [37], it was pointed out that there exist cases where

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

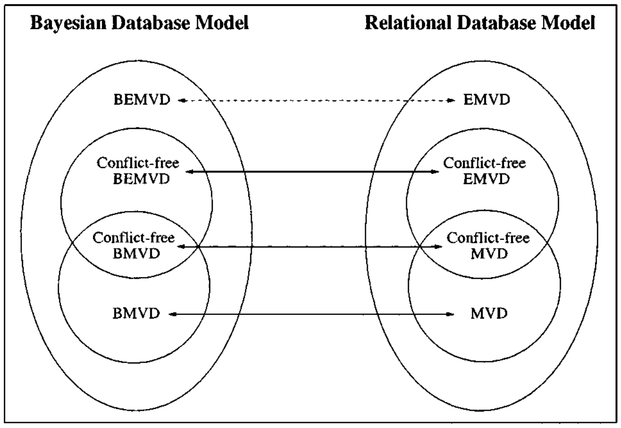

Fig. 1. Four classes of *probabilistic* dependencies (BEMVD, conflict-free BEMVD, BMVD, conflict-free BMVD) traditionally found in the Bayesian database model are depicted on the left. The corresponding class of *data* dependencies (EMVD, conflict-free EMVD, MVD, conflict-free MVD) in the standard relational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

and

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

(A double solid arrow in Fig. 1 represents the fact that proposition (2) holds, while a double dashed arrow indicates that proposition (2) does not hold.) Since the implication problems do not coincide in the pair (BEMVD, EMVD), it was suggested in [37] that Bayesian networks are intrinsically *different* from relational databases. This remark, 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* [17]. (By solvability, we mean there exists a method which in a finite number of steps can decide whether $\Sigma \models \sigma$ holds for an arbitrary instance (Σ, σ) of the implication problem.) Therefore, the observation in (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 this result is the observation in (4). Therefore, the fact that the implication problems in Bayesian networks and relational databases do not coincide is based on *unsolvable* classes of dependencies, as illustrated in Fig. 2. 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 II contains background knowledge including the traditional relational database model, our Bayesian database model, and formal definitions of the four classes of probabilistic conditional independencies studied here. In Section III, we introduce the basic notions pertaining to the implication problem. In Section IV, we present an in-depth analysis of the implication problem for the BMVD class. In particular, we present the *chase* algorithm as a *nonaxiomatic* method for testing the implication of this special class of *nonembedded* probabilistic conditional independencies. In Section V, we examine the implication problem for *embedded* dependencies. The conclusion is presented in Section VI, in which we emphasize that Bayesian networks are indeed a general form of relational databases.

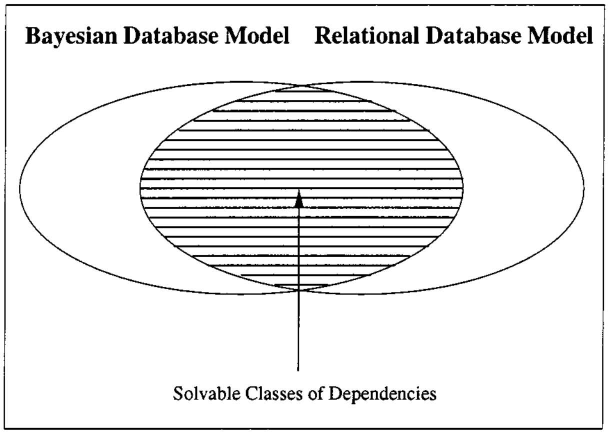

Fig. 2. Implication problems coincide on the *solvable* classes of dependencies.

II. Background Knowledge

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

A. Acyclic Hypergraphs

Acyclic hypergraphs are useful for graphically representing dependencies (independencies). Let $R = \{A_1, A_2, \cdots, A_m\}$ be a finite set of attributes. A *hypergraph* $\mathcal{R} = \{R_1, R_2, \cdots, R_n\}$ is a family 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 $R_1, R_2, \cdots, R_n$ of $\mathcal{R}$ such that there exists a branching function $b(i) < i$ such that $R_i \cap (R_1 \cup R_2 \cup \cdots \cup R_{i-1}) \subseteq R_{b(i)}$, for $i = 2, 3, \cdots, n$. We call $\mathcal{R}$ an *acyclic hypergraph*, if and only if $\mathcal{R}$ has the running intersection property [4]. Given an ordering $R_1, R_2, \cdots, R_n$ 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

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

These J-keys are in fact independent of a particular hypertree construction ordering, that is, 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 $\mathcal{R} = \{R_1 = \{A_1, A_2, A_3\}, R_2 = \{A_2, A_3, A_4\}, R_3 = \{A_2, A_3, A_5\}, R_4 = \{A_5, A_6\}\}$ define the hypergraph in Fig. 3. It can be easily verified that R_1, R_2, R_3, R_4 is a hypertree construction ordering for $\mathcal{R}$

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

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

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

In the probabilistic reasoning literature, the graphical structure of a (decomposable) Markov network [16], [31] is specified with 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 [4]. (In fact, a given acyclic hypergraph may have a number of jointrees.)

B. Relational Databases

To clarify the notations, we give a brief review of the standard relational database model [23]. The relational concepts presented here are generalized in Section II-D to express the probabilistic network concepts in Section II-C.

A *relation scheme* $R = \{A_1, A_2, \cdots, A_m\}$ 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 $D = D_{A_1} \cup D_{A_2} \cdots \cup D_{A_m}$. A *relation* r on the relation scheme R, written $r(R)$, is a finite set of mappings $\{t_1, t_2, \cdots, t_s\}$ 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 $R = \{A_1, A_2, \cdots, A_m\}$ in general is shown in Fig. 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, where $X \subseteq R$ is an arbitrary set of attributes.

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 conventions will be adopted. Uppercase letters A, B, C from the beginning of the alphabet will be used to denote attributes. A relation scheme $R = \{A_1, A_2, \cdots, A_m\}$ is written as simply $A_1 A_2 \cdots A_m$. A relation r on scheme R is denoted by either $r(R)$ or $r(A_1 A_2 \cdots A_m)$. 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 $R = ABCD$ is shown at the top of Fig. 5, where the domain of each attribute in R is $\{0, 1\}$.

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

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

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

$$\begin{aligned} r_1(X) &\bowtie r_2(Y) \\ &= \{t(XY) | t(X) \in r_1(X) \text{ and } t(Y) \in r_2(Y)\}. \end{aligned} \tag{7}$$

Let X, Y, Z be subsets of R such that $(Y \cap Z) \subseteq X$. We say relation $r(R)$ satisfies the embedded multivalued dependency (EMVD) $X \rightarrow\rightarrow Y|Z$ in the context XYZ, if the projection $\pi_{XYZ}(r)$ of $r(R)$ satisfies the condition

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

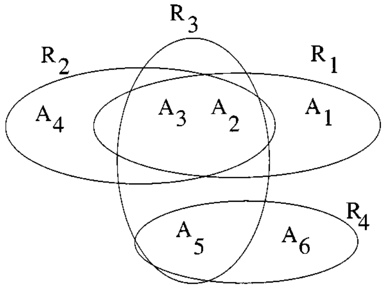

Fig. 3. Graphical representation of the acyclic hypergraph $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$.

$r(R) =$

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

Fig. 4. Relation r on the scheme $R = \{A_1, A_2, \cdots, A_m\}$.

$r(ABCD) =$

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

$\pi_{ABC}(r) =$

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

Fig. 5. Relation $r(ABCD)$ satisfies the EMVD $B \rightarrow\rightarrow A|C$, since $\pi_{ABC}(r) = \pi_{AB}(r) \bowtie \pi_{BC}(r)$.

Example 2: Relation $r(ABCD)$ at the top of Fig. 5 satisfies the EMVD $B \rightarrow\rightarrow A|C$, since $\pi_{ABC}(r) = \pi_{AB}(r) \bowtie \pi_{BC}(r)$.

In the special case when $XYZ = R$, we call $X \rightarrow\rightarrow Y|Z$ *(nonembedded) multivalued dependency* (MVD), or *full* MVD. It is therefore clear that MVD is a *special case* of the more general EMVD class, as shown in Fig. 1. We write the MVD $X \rightarrow\rightarrow Y|Z$ as $X \rightarrow\rightarrow Y$ since the context is understood. MVD can be equivalently defined as follows. 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

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

It is not necessary to assume that X and Y are disjoint since

$$X \rightarrow\rightarrow Y \Longleftrightarrow X \rightarrow\rightarrow Y - X.$$

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

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

As indicated in Fig. 1, there is subclass of (nonembedded) MVDs called *conflict-free* MVD. Unlike arbitrary sets of MVDs, conflict-free MVDs can be *faithfully* represented by 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 the acyclic hypergraph, and every MVD inferred from the acyclic hypergraph is logically implied by the conflict-free set. The next example illustrates the notion of a *perfect-map*.

Example 3: Consider the following set C of MVDs on $R = A_1A_2A_3A_4A_5A_6$:

$$\begin{aligned} C = \{&A_2A_3 \rightarrow\rightarrow A_1, A_2A_3 \rightarrow\rightarrow A_4, A_2A_3 \rightarrow\rightarrow A_5A_6,\\ &A_5 \rightarrow\rightarrow A_1A_2A_3A_4, A_5 \rightarrow\rightarrow A_6, A_2A_3A_5 \rightarrow\rightarrow A_1\}. \end{aligned} \tag{10}$$

This set of MVDs can be *faithfully* represented by the acyclic hypergraph $\mathcal{R}$ in Fig. 3. 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 Fig. 3 is a *perfect-map* of the set C of MVDs in (10).

Note that the set C of MVDs in (10) is *conflict-free*. It is important to realize that there are some sets of MVDs which cannot be faithfully represented by a single acyclic hypergraph.

Example 4: Consider the following set C of MVDs on $R = A_1A_2A_3$:

$$C = \{A_1 \rightarrow\rightarrow A_2, A_3 \rightarrow\rightarrow A_2\}. \tag{11}$$

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 $A_1 \rightarrow\rightarrow A_2$ in C can be inferred from $\mathcal{R}$ using the method of separation. However, the MVD $A_3 \rightarrow\rightarrow A_2$ 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 $A_3 \rightarrow\rightarrow A_2$ but not $A_1 \rightarrow\rightarrow A_2$.

Example 4 indicates that the class of *conflict-free* MVDs is a subclass of the MVD class. For example, C in (11) is a member of the MVD class, but is not a member of the conflict-free MVD class.

C. Bayesian Networks

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

Let $R = \{A_1, A_2, \cdots, A_m\}$ 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* [16], [28], [31] on D is a function p on D, p: $D \rightarrow [0, 1]$. That is, this function p assigns to each tuple $t \equiv \langle t(A_1), t(A_2), \cdots, t(A_m)\rangle \in D$ a real number $0 \leq p(t) \leq 1$ and p is normalized, namely, $\Sigma_{t \in D}\ p(t) = 1$. For convenience, we write a joint probability distribution p as $p(A_1, A_2, \cdots, A_m)$ over the set R of variables. In particular, we use $p(a_1, a_2, \cdots, a_m)$ to denote a particular value of $p(t) = p(\langle t(A_1), t(A_2), \cdots, t(A_m)\rangle)$. That is, $p(a_1, a_2, \cdots, a_m)$ denotes the probability value $p(\langle t(A_1), t(A_2), \cdots, t(A_m)\rangle)$ of the function p for a particular *instantiation* of the variables $A_1, A_2, \cdots, A_m$. In general, a *potential* [16] is a function q on D such that $q(t)$ is a nonnegative real number and $\Sigma_{t \in D}\ q(t)$ is positive, i.e., at least one $q(t) > 0$.

We now introduce the fundamental notion of *probabilistic conditional independency*. Let X, Y and Z be disjoint subsets of variables in R. Let x, y, and 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

$$p(y|xz) = p(y|x) \tag{12}$$

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

$$p(yxz) = \frac{p(yx) \cdot p(xz)}{p(x)}. \tag{13}$$

We write $I_p(Y, X, Z)$ as $I(Y, X, Z)$ if the joint probability distribution p is understood.

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

$$\begin{aligned} p(A_1, A_2, \cdots, A_m) =& p(A_1) \cdot p(A_2|A_1) \cdot p(A_3|A_1, A_2) \cdots\\ &\cdot p(A_m|A_1, A_2, \cdots, A_{m-1}). \end{aligned}$$

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

Example 5: Consider a joint probability distribution $p(A_1, A_2, A_3, A_4, A_5, A_6)$ which satisfies the set $\mathbf{C}$ of probabilistic conditional independencies

$$\begin{aligned} \mathbf{C} = \{&I(A_1, \emptyset, \emptyset), I(A_2, A_1, \emptyset), I(A_3, A_1, A_2)\\ &I(A_4, A_2A_3, A_1), I(A_5, A_2A_3, A_1A_4)\\ &I(A_6, A_5, A_1A_2A_3A_4)\}. \end{aligned} \tag{14}$$

Equivalently, we have

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

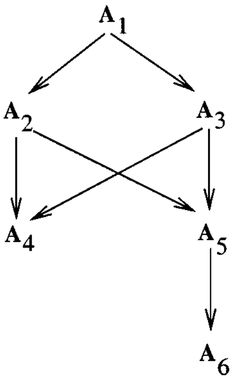

Fig. 6. DAG representing all of the probabilistic conditional independencies satisfied by the joint distribution defined by (15).

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

$$\begin{aligned} &p(A_1, A_2, A_3, A_4, A_5, A_6) \\ &\quad = p(A_1) \cdot p(A_2|A_1) \cdot p(A_3|A_1) \cdot p(A_4|A_2, A_3) \\ &\qquad \cdot p(A_5|A_2, A_3) \cdot p(A_6|A_5). \end{aligned} \tag{15}$$

We can represent all of the probabilistic conditional independencies satisfied by this joint distribution by the DAG shown in Fig. 6. 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].

Example 5 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.

To facilitate the computation of marginal distributions, it is useful to transform a Bayesian network into a (decomposable) Markov network. A *Markov network* [16] 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 [16], [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, the DAG in Fig. 6 can be transformed into the acyclic hypergraph depicted in Fig. 3. *Local computation* procedures [45] can be applied to transform the conditional probability distributions into marginal distributions defined over the acyclic hypergraph. The joint probability distribution in (15) can be rewritten, in terms of marginal distributions over the acyclic hypergraph in Fig. 3, as (16), shown at the bottom of the page. The Markov network representation of probabilistic knowledge in (16) is typically used for inference in many practical applications.

D. A Bayesian Database Model

Here we review our Bayesian database model [42], [45] which serves as a unified approach for both Bayesian networks and relational databases.

A potential $q(R)$ can be represented as a *probabilistic* relation $\mathbf{r}(R, A_q)$, where the column labeled by A_q stores the probability value. The relation $\mathbf{r}(A_1, A_2, \cdots, A_m, A_q)$ representing a potential $q(A_1, A_2, \cdots, A_m)$ contains tuples of the form $\mathbf{t} = \langle t, q(t) \rangle$, as shown in Fig. 7. Let $r(R)$ be the standard database relation representing the tuples with *positive* probability, namely

$$r(R) = \{t(R)|q(t) > 0\}.$$

The probabilistic relation $\mathbf{r}(R, A_q)$ representing the potential $q(R)$ is defined as

$$\mathbf{r}(R, A_q) = \{\mathbf{t}(R, A_q)|\mathbf{t}(R) = t(R) \in r(R)$$

and

$$\mathbf{t}(A_q) = q(t)\}.$$

For convenience we will write $\mathbf{r}(R, A_q)$ as $\mathbf{r}(R)$ and say relation $\mathbf{r}$ is on R with the attribute A_q understood by context. That is, relations denoted by boldface represent probability distributions. For example, a potential $q(A_1A_2A_3)$ is shown at the top of Fig. 8. The traditional relation $r(A_1A_2A_3)$ and the probabilistic relation $\mathbf{r}(A_1A_2A_3)$ corresponding to $q(A_1A_2A_3)$ are shown at the bottom of Fig. 8.

Let $\mathbf{r}(R)$ be a relation and X be a subset of R. In our notation, the *marginalization of* $\mathbf{r}$ *onto* X, written $\tau_X(\mathbf{r})$, is defined as

$$\tau_X(\mathbf{r}) = \left\{ \mathbf{t}(XA_{q(X)})|\mathbf{t}(X) \in \pi_X(\mathbf{r}) \right.$$

and

$$\left. \mathbf{t}(A_{q(X)}) = \sum_{\substack{\mathbf{t}' \in \mathbf{r}, \\ \mathbf{t}'(X) = \mathbf{t}(X)}} \mathbf{t}'(A_q) \right\}. \tag{17}$$

The relation $\tau_X(\mathbf{r})$ represents the usual *marginal distribution* $q(X)$ of $q(R)$ onto X. By definition of $\mathbf{r}(R)$, $\tau_X(\mathbf{r})$ does not contain any tuples with zero probability.

Example 6: Given the relation $\mathbf{r}(A_1A_2A_3)$ at the top of Fig. 9, the marginalization of $\mathbf{r}$ onto A_1A_2 is the relation $\tau_{A_1A_2}(\mathbf{r})$ shown at the bottom.

$$p(A_1, A_2, A_3, A_4, A_5, A_6) = \frac{p(A_1, A_2, A_3) \cdot p(A_2, A_3, A_4) \cdot p(A_2, A_3, A_5) \cdot p(A_5, A_6)}{p(A_2, A_3) \cdot p(A_2, A_3) \cdot p(A_5)}. \tag{16}$$

A_1	A_2	$\dots$	A_m	A_q
$\mathbf{t}_1(A_1)$	$\mathbf{t}_1(A_2)$	$\dots$	$\mathbf{t}_1(A_m)$	$\mathbf{t}_1(A_q) = q(t_1)$
$\mathbf{t}_2(A_1)$	$\mathbf{t}_2(A_2)$	$\dots$	$\mathbf{t}_2(A_m)$	$\mathbf{t}_2(A_q) = q(t_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathbf{t}_s(A_1)$	$\mathbf{t}_s(A_2)$	$\dots$	$\mathbf{t}_s(A_m)$	$\mathbf{t}_s(A_q) = q(t_s)$

Fig. 7. Potential $q(R)$ expressed as a *probabilistic* relation $\mathbf{r}(R)$.

$q(A_1A_2A_3) =$

A_1	A_2	A_3	
0	0	0	0.1
0	0	1	0.2
0	1	0	0.0
0	1	1	0.1
1	0	0	0.0
1	0	1	0.0
1	1	0	0.1
1	1	1	0.3

$r(A_1A_2A_3) =$

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

$\mathbf{r}(A_1A_2A_3) =$

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

Fig. 8. Potential $q(A_1A_2A_3)$ is shown at the top of the figure. The database relation $r(A_1A_2A_3)$ and the probabilistic relation $\mathbf{r}(A_1A_2A_3)$ corresponding to $q(A_1A_2A_3)$ are shown at the bottom of the figure.

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

$$\begin{aligned}&\mathbf{r}_1(X) \times \mathbf{r}_2(Y)\\ &\quad = \{\mathbf{t}(XYA_{q_1(X)\cdot q_2(Y)}) | \mathbf{t}(XY) \in \pi_X(\mathbf{r}_1) \bowtie \pi_Y(\mathbf{r}_2) \text{ and}\\ &\mathbf{t}(A_{q_1(X)\cdot q_2(Y)})\\ &\quad = \mathbf{t}(A_{q_1(X)}) \cdot \mathbf{t}(A_{q_2(Y)})\}.\end{aligned}$$

That is, $\mathbf{r}_1(X) \times \mathbf{r}_2(Y)$ represents the potential $q_1(X) \cdot q_2(Y)$ obtained by multiplying the two individual potentials $q_1(X)$ and $q_2(Y)$.

Example 7: Let $\mathbf{r}_1(A_1A_2)$ and $\mathbf{r}_2(A_2A_3)$ represent potentials $q_1(A_1A_2)$ and $q_2(A_2A_3)$. The product join $\mathbf{r}_1(A_1A_2) \times \mathbf{r}_2(A_2A_3)$ of relations $\mathbf{r}_1(A_1A_2)$ and $\mathbf{r}_2(A_2A_3)$ is shown in Fig. 10.

Probabilistic conditional independency is defined as *Bayesian* EMVD (BEMVD) in our Bayesian database model. A probabilistic relation $\mathbf{r}(XYZW)$ satisfies the *Bayesian embedded multivalued dependency* (BEMVD), $X \Rightarrow\Rightarrow Y|Z$, if

$$\tau_{XYZ}(\mathbf{r}) = \tau_{XY}(\mathbf{r}) \times \tau_{XZ}(\mathbf{r}) \times \tau_X(\mathbf{r})^{-1} \qquad (18)$$

$\mathbf{r}(A_1A_2A_3) =$

A_1	A_2	A_3	A_q
0	0	0	0.1
0	0	1	0.2
1	0	0	0.5

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

A_1	A_2	$A_{q(A_1A_2)}$
0	0	0.3
1	0	0.5

Fig. 9. Relation $\mathbf{r}(A_1A_2A_3)$ representing a potential $q(A_1A_2A_3)$ is shown at the top. At the bottom is the marginalization $\tau_{A_1A_2}(\mathbf{r})$ of relation $\mathbf{r}(A_1A_2A_3)$ onto A_1A_2.

$\mathbf{r}_1(A_1A_2) \times \mathbf{r}_2(A_2A_3)$

$=$

A_1	A_2	$A_{q_1(A_1A_2)}$
1	1	0.1
2	1	0.2
1	2	0.3

$\times$

A_2	A_3	$A_{q_2(A_2A_3)}$
1	1	0.2
1	2	0.3
3	1	0.4

$=$

A_1	A_2	A_3	$A_{q_1(A_1A_2)\cdot q_2(A_2A_3)}$
1	1	1	0.02
1	1	2	0.03
2	1	1	0.04
2	1	2	0.06

Fig. 10. Product join $\mathbf{r}_1(A_1A_2) \times \mathbf{r}_2(A_2A_3)$ of relations $\mathbf{r}_1(A_1A_2)$ and $\mathbf{r}_2(A_2A_3)$.

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

$$\begin{aligned}\tau_X(\mathbf{r})^{-1} = &\{\mathbf{t}(XA_{1/p(X)}) | \mathbf{t}(X) = \mathbf{t}'(X) \in \tau_X(\mathbf{r}) \text{ and}\\ &\mathbf{t}(A_{1/p(X)}) = 1/\mathbf{t}'(A_{p(X)})\}.\end{aligned}$$

Note that this inverse relation $\tau_X(\mathbf{r})^{-1}$ is well defined because by definition $\tau_X(\mathbf{r})$ does not contain any tuples with zero probability. By introducing a binary operator $\otimes$ called *Markov join*, the right-hand side of (18) can be written as

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

Thus, in terms of this notation, we say that a relation $\mathbf{r}(XYZ)$ satisfies the BEMVD $X \Rightarrow\Rightarrow Y|Z$, if and only if

$$\tau_{XYZ}(\mathbf{r}) = \tau_{XY}(\mathbf{r}) \otimes \tau_{XZ}(\mathbf{r}). \qquad (19)$$

It is not necessary to assume that X, Y, and Z are disjoint since

$$X \Rightarrow\Rightarrow Y|Z \Longleftrightarrow X \Rightarrow\Rightarrow (Y - X)|(Z - X).$$

Example 8: Relation $\mathbf{r}(ABCD)$ at the top of Fig. 11 satisfies the BEMVD $B \Rightarrow\Rightarrow A|C$, since the marginal $\tau_{ABC}(\mathbf{r})$ can be written as $\tau_{ABC}(\mathbf{r}) = \tau_{AB}(\mathbf{r}) \otimes \tau_{BC}(\mathbf{r})$.

In the special case when $XYZ = R$, we call the BEMVD $X \Rightarrow\Rightarrow Y|Z$ *nonembedded* BEMVD, *full* BEMVD, or simply *Bayesian multivalued dependency* (BMVD). For notational convenience we write the BMVD $X \Rightarrow\Rightarrow Y|Z$ as $X \Rightarrow\Rightarrow Y$ if $Z = R - XY$ is understood by context.

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

$\mathbf{r}(ABCD)$ =

A	B	C	D	$A_{p(ABCD)}$
0	0	0	0	0.1
0	0	0	1	0.1
0	0	1	1	0.2
1	0	0	0	0.1
1	0	1	0	0.1
1	1	1	1	0.4

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

A	B	C	$A_{p(ABC)}$
0	0	0	0.2
0	0	1	0.2
1	0	0	0.1
1	0	1	0.1
1	1	1	0.4

=

A	B	$A_{p(AB)}$
0	0	0.4
1	0	0.2
1	1	0.4

⊗

B	C	$A_{p(BC)}$
0	0	0.3
0	1	0.3
1	1	0.4

=

A	B	C	$A_{\frac{p(AB)p(BC)}{p(B)}}$
0	0	0	(0.4)(0.3)/(0.6) = 0.2
0	0	1	(0.4)(0.3)/(0.6) = 0.2
1	0	0	(0.2)(0.3)/(0.6) = 0.1
1	0	1	(0.2)(0.3)/(0.6) = 0.1
1	1	1	(0.4)(0.4)/(0.4) = 0.4

Fig. 11. Relation $\mathbf{r}(ABCD)$ satisfies the BEMVD $B \Rightarrow\Rightarrow A|C$, since $\tau_{ABC}(\mathbf{r}) = \tau_{AB}(\mathbf{r}) \otimes \tau_{BC}(\mathbf{r})$.

to stating that Y and Z are conditionally independent given X under p in (13), namely

$$X \Rightarrow\Rightarrow Y|Z \Longleftrightarrow I(Y, X, Z). \tag{20}$$

Thus, we can use the terms BEMVD and probabilistic conditional independency interchangeably.

E. Terminology in the Bayesian and Relational Database Models

Our goal here is to demonstrate that there is a direct correspondence between the notions used in relational databases and probabilistic networks.

As already mentioned, any *potential* $q(R)$ can be viewed as a *probabilistic* relation $\mathbf{r}(R)$ in our Bayesian database model. Obviously, the only difference between a probabilistic relation $\mathbf{r}(R)$ and a standard relation $r(R)$ is the additional column labeled by A_q for storing the probability value. As shown in Fig. 12, in the Bayesian database model it is crucial to *count* the duplicate tuples, whereas duplicate tuples are *ignored* in the relational 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 Figs. 13 and 14.

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:

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

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.

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 I.

$\mathbf{r}(A_1A_2A_3)$ =

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

$r(A_1A_2A_3)$ =

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

Fig. 12. In the Bayesian database model it is crucial to *count* the duplicate tuples, as reflected by the probabilistic relation $\mathbf{r}(A_1A_2A_3)$. On the other hand, duplicate tuples are *ignored* in the relational database model, as reflected by the standard relation $r(A_1A_2A_3)$.

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

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

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

A_1	A_2
0	0
1	0

Fig. 13. Relation $\tau_{A_1A_2}(\mathbf{r})$ is the marginalization of $\mathbf{r}(A_1A_2A_3)$ in Fig. 12, and $\pi_{A_1A_2}(r)$ is the projection of $r(A_1A_2A_3)$.

A_1	A_2
1	1
2	1
1	2

⋈

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

×

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

Fig. 14. Natural join $r(A_1A_2) \bowtie r(A_2A_3)$ of relations $r(A_1A_2)$ and $r(A_2A_3)$ (top). Product join $\mathbf{r}(A_1A_2) \times \mathbf{r}(A_2A_3)$ of relations $\mathbf{r}(A_1A_2)$ and $\mathbf{r}(A_2A_3)$ (bottom).

TABLE I
CORRESPONDING TERMINOLOGY IN THE THREE MODELS

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 $\times$
EMVD $X \rightarrow\rightarrow Y \vert Z$	conditional independency $I(Y, X, Z)$	BEMVD $X \Rightarrow\Rightarrow Y \vert Z$

III. SUBCLASSES OF PROBABILISTIC CONDITIONAL INDEPENDENCIES

In this section, we emphasize the fact that probabilistic networks are constructed using special *conflict-free* subclasses within the general class of probabilistic conditional independencies. That is, Bayesian networks are not constructed using *arbitrary* sets of probabilistic conditional independencies, just as Markov networks are not constructed using *arbitrary* sets of nonembedded (full) probabilistic conditional independencies.

Probabilistic conditional independency is called *Bayesian embedded multivalued dependency* (BEMVD) in our approach. We define the general BEMVD class as follows:

$$(\mathbf{1a})\ \text{BEMVD} = \{\mathbf{C} | \mathbf{C} \text{ is a set of probabilistic conditional independencies}\}. \tag{21}$$

Bayesian networks are defined by a DAG and a corresponding set of conditional probability distributions. Such a DAG encodes 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 Fig. 6 using the d-separation method. However, it is important to realize that there are some sets of probabilistic conditional independencies that cannot be *faithfully* encoded by a single DAG.

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

$$\mathbf{C} = \{I(A, B, C), I(A, C, B), I(AB, C, D)\}. \tag{22}$$

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

Example 9 clearly indicates that Bayesian networks are defined only using a subclass of probabilistic conditional independencies. In order to label this subclass of independencies, we first recall the notion of perfect-map. A graphical structure is called a *perfect-map* [4], [31] of a given set $\mathbf{C}$ of probabilistic conditional independencies, if every conditional independency logically implied by $\mathbf{C}$ can be inferred from the graphical structure, and every conditional independency that can be inferred from the graphical structure is logically implied by $\mathbf{C}$. (We say $\mathbf{C}$ *logically implies* $\mathbf{c}$ and write $\mathbf{C} \models \mathbf{c}$, if whenever any distribution that satisfies all the conditional independencies in $\mathbf{C}$, then the distribution also satisfies $\mathbf{c}$.) A set $\mathbf{C}$ of probabilistic conditional independencies is called *conflict-free* if there exists a DAG which is a perfect-map of $\mathbf{C}$.

We now can define the *conflict-free BEMVD* subclass used by Bayesian networks as follows:

$$(\mathbf{1b})\ \text{Conflict-free BEMVD} = \{\mathbf{C} | \text{ there exists a DAG which is a } \cdot perfect - map \text{ of } \mathbf{C}\}. \tag{23}$$

It should be clear that a causal input list is a *cover* [23] of a conflict-free set of conditional independencies. (A *causal input list* [32] or a *stratified protocol* [39] over a set $R = A_1A_2 \cdots A_m$ of variables would contain precisely m conditional independency statements $I(X, Y, Z)$. For example, the set $\mathbf{C}$ of conditional independencies in (14) is an example of a causal input list since $\mathbf{C}$ precisely defines the DAG in Fig. 6. Since the conditional independency $I(A_5, A_2A_3A_4, A_1)$ can be inferred from the DAG in Fig. 6, $\mathbf{C} \cup \{I(A_5, A_2A_3A_4, A_1)\}$ is still a conflict-free set but not a causal input list.)

As illustrated in Fig. 1, the main point is that the conflict-free BEMVD class is a subclass within the BEMVD class. For example, the set $\mathbf{C}$ of conditional independencies in (22) belongs to the general BEMVD class in (21) but does not belong to conflict-free BEMVD subclass in (23).

Another subclass within the general BEMVD class are the *nonembedded* probabilistic conditional independencies. Nonembedded probabilistic conditional independency is also called *full* [26] or *fixed context* [13]. *Nonembedded* conditional independencies are those which involve *all* variables, i.e., $I(Y, X, Z)$ where $XYZ = R$.

Example 10: Let $R = \{A, B, C, D\}$. Consider the following set $\mathbf{C}$ of probabilistic conditional independencies:

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

The first independency $I(A, BC, D)$ is *nonembedded* (full) since $\{A, B, C, D\} = R$, but the second independency $I(A, B, C)$ is not full because $\{A, B, C\} \subset R$.

The class of nonembedded probabilistic conditional independencies is called *Bayesian multivalued dependency* (BMVD) in our approach. We define the BMVD class as follows:

$$(\mathbf{2a})\ \text{BMVD} = \{\mathbf{C} | \mathbf{C} \text{ is a set of } nonembedded \text{ probabilistic conditional independencies}\}. \tag{24}$$

Nonembedded (full) independencies are important since Markov networks do not reflect *embedded* conditional independencies. For instance, the Bayesian distribution in (15) satisfies the (embedded) probabilistic conditional independency $I(A_3, A_1, A_2)$, while the Markov distribution in (16) does not. That is, Markov distributions only reflect *nonembedded* probabilistic conditional independencies.

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

Example 11: 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: $S_1 = ABC, S_2 = E, S_3 = FGH$. By definition, the BMVDs $D \Rightarrow\Rightarrow ABC$, $D \Rightarrow\Rightarrow E$, $D \Rightarrow\Rightarrow FGH$, and $D \Rightarrow\Rightarrow ABCE$ can be inferred from $\mathcal{R}$. On the other hand, the BMVD $D \Rightarrow\Rightarrow BC$ is *not* inferred from $\mathcal{R}$ since BC is not equal to the union of some of the sets in $\{S_1, S_2, S_3\}$.

Just as Bayesian networks are not constructed using arbitrary sets of BEMVDs, Markov networks are not constructed using *arbitrary* sets of BMVDs. That is, there are sets of nonembedded independencies which cannot be *faithfully* encoded by a single acyclic hypergraph.

Example 12: Consider the following set $\mathbf{C}$ of nonembedded probabilistic conditional independencies on $\{A, B, C\}$:

$$\mathbf{C} = \{I(A, B, C), I(A, C, B)\}. \tag{25}$$

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

Example 12 clearly indicates that Markov networks are defined only using a subclass of nonembedded probabilistic conditional independencies. The notion of conflict-free is again used to label this subclass. A set $\mathbf{C}$ of nonembedded probabilistic conditional independencies is called *conflict-free* if there exists an acyclic hypergraph which is a perfect-map of $\mathbf{C}$.

We now can define the *conflict-free BMVD* subclass used by Markov networks as follows:

$$\begin{aligned}(\mathbf{2b})\ &\text{Conflict-free BMVD}\\ &\quad = \{\mathbf{C} \mid \text{there exists an acyclic}\\ &\qquad \text{hypergraph which is a } perfect - map \text{ of } \mathbf{C}\}.\end{aligned} \tag{26}$$

As illustrated in Fig. 1 (left), the main point is that the conflict-free BMVD class is a subclass within the BMVD class. For example, the set $\mathbf{C}$ of nonembedded probabilistic conditional independencies in (25) belongs to the BMVD class in (24) but not to the conflict-free BMVD class in (26).

We conclude this section by pointing out another similarity between relational databases and Bayesian networks. The notion of conflict-free MVDs was originally proposed by Lien [22] in the study of the relationship between various database models. It has been shown [4] that a conflict-free set C of MVDs is *equivalent* to another data dependency called *acyclic join dependency* (AJD) (defined below). That is, whenever any relation satisfies all of the MVDs in C, then the relation also satisfies a corresponding AJD, and vice versa. An AJD guarantees that a relation can be decomposed losslessly into two or more projections (smaller relations). Let $\mathcal{R} = \{R_1, R_2, \cdots, R_n\}$ be an acyclic hypergraph on the set of attributes $R = R_1 \cup R_2 \cup \cdots \cup R_n$. We say that a relation $r(R)$ satisfies the *acyclic join dependency* (AJD), $\bowtie \{R_1, R_2, \cdots, R_n\}$ if:

$$r(R) = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \cdots \bowtie \pi_{R_n}(r). \tag{27}$$

That is, r decomposes losslessly onto $\mathcal{R}$. We also write $\bowtie \{R_1, R_2, \cdots, R_n\}$ as $\bowtie \mathcal{R}$.

Example 13: Relation $r(R)$ at the top of of Fig. 15 satisfies the AJD, $\bowtie \mathcal{R}$, where $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$ is the acyclic hypergraph in Fig. 3. That is,

$$\begin{aligned}r(A_1A_2A_3A_4A_5A_6) = {}& \pi_{A_1A_2A_3}(r) \bowtie \pi_{A_2A_3A_4}(r)\\ & \cdot \bowtie \pi_{A_2A_3A_5}(r) \bowtie \pi_{A_5A_6}(r).\end{aligned}$$

The conflict-free class of MVDs, namely, AJDs, play a major role in database design since it exhibits many desirable properties in database applications [4]. In our unified model, a Markov network can be easily seen as a *generalized* form of AJD.

Let $\mathcal{R} = \{R_1, R_2, \cdots, R_n\}$ be an acyclic hypergraph on the set of attributes $R = R_1 \cup R_2 \cup \cdots \cup R_n$. We say a *Bayesian acyclic join dependency* (BAJD), written $\otimes\mathcal{R}$, is satisfied by a relation $\mathbf{r}(R)$, if

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

where the sequence $R_1, R_2, \cdots, R_n$ is a hypertree construction ordering for $\mathcal{R}$. Since the probabilistic relation $\mathbf{r}(R)$ does not contain any tuples $\mathbf{t}(A_p) = 0$, the AJD, $\bowtie \mathcal{R}$, is a *necessary* condition for $\mathbf{r}(R)$ to satisfy the BAJD, $\otimes\mathcal{R}$.

Example 14: Recall the distribution defined by the Markov network in (16), namely (29), shown at the bottom of the next page, where $\mathcal{R} = \{A_1A_2A_3, A_2A_3A_4, A_2A_3A_5, A_5A_6\}$ is the acyclic hypergraph in Fig. 3. Let $\mathbf{r}(A_1A_2A_3A_4A_5A_6)$ be the probabilistic relation representing $p(A_1, A_2, A_3, A_4, A_5, A_6)$ in (29). It can be seen that $\mathbf{r}(A_1A_2A_3A_4A_5A_6)$ satisfies the BAJD $\otimes\ \mathcal{R}$, namely

$$\begin{aligned}&\mathbf{r}(A_1A_2A_3A_4A_5A_6)\\ &\quad = (((\tau_{A_1A_2A_3}(\mathbf{r}) \otimes \tau_{A_2A_3A_4}(\mathbf{r})) \otimes \tau_{A_2A_3A_5}(\mathbf{r}))\\ &\qquad \otimes \tau_{A_5A_6}(\mathbf{r})\\ &\quad = \tau_{A_1A_2A_3}(\mathbf{r}) \times \tau_{A_2A_3A_4}(\mathbf{r}) \times \tau_{A_2A_3}(\mathbf{r})^{-1}\\ &\qquad \times \tau_{A_2A_3A_5}(\mathbf{r}) \times \tau_{A_2A_3}(\mathbf{r})^{-1} \times \tau_{A_5A_6}(\mathbf{r}) \times \tau_{A_5}(\mathbf{r})^{-1}.\end{aligned}$$

The relation $\mathbf{r}(R)$ at the bottom of Fig. 15 satisfies this BAJD $\otimes\mathcal{R}$.

Example 14 clearly demonstrates that the representation of knowledge in practice is the *same* for both relational and probabilistic applications. An acyclic join dependency (AJD)

$$r(R) = \bowtie \{R_1, R_2, \cdots, R_n\}$$

and a (decomposable) Markov network

$$p(R) = \frac{p(R_1) \cdot p(R_2) \cdot \cdots \cdot p(R_n)}{p(R_1 \cap R_2) \cdot \cdots \cdot p(R_{n-1} \cap R_n)}$$

$r(R) =$

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

$\mathbf{r}(R) =$

A_1	A_2	A_3	A_4	A_5	A_6	A_p
0	0	0	0	1	0	0.4
0	0	0	1	1	0	0.2
1	0	1	1	0	1	0.2
1	1	0	1	1	0	0.2

Fig. 15. Relation $r(R)$ at the top satisfies the AJD, $\bowtie \mathcal{R}$. Relation $\mathbf{r}(R)$ at the bottom satisfies the BAJD, $\otimes\mathcal{R}$. The acyclic hypergraph $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$ is depicted in Fig. 3.

or in our terminology, the BAJD

$$\mathbf{r}(R) = \otimes\{R_1, R_2, \cdots, R_n\}$$

are both defined over an acyclic hypergraph.

The discussion in Section II-E explicitly demonstrates that there is a *direct* correspondence between the concepts used in relational databases and Bayesian networks. The discussion at the end of this section clearly indicates that *both* intelligent systems represent their knowledge over acyclic hypergraphs in practice. However, the relationship between relational databases and Bayesian networks can be rigorously formalized by studying the *implication problems* for the four classes of probabilistic conditional independencies defined in this section.

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

The *implication problem* is to test whether a given set Σ 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.

A 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 Σ of dependencies and a set of inference axioms, the *closure* of Σ, written Σ^+, is the smallest set containing Σ such that the inference axioms cannot be applied to the set to yield a dependency not in the set. More specifically, the set Σ *derives* a dependency σ, written $\Sigma \vdash \sigma$, if σ is in Σ^+. A set of inference axioms is *sound* if whenever $\Sigma \vdash \sigma$, then $\Sigma \models \sigma$. A set of inference axioms is *complete* if the converse holds, that is, if $\Sigma \models \sigma$, then $\Sigma \vdash \sigma$. In other words, saying a set of axioms are complete means that if Σ logically implies the dependency σ, then Σ derives σ. A sequence s of dependencies over R is a *derivation sequence* on Σ if every dependency in s is either

1) a member of Σ, or
2) follows from previous dependencies in s by an application of one of the given inference axioms.

Note that R is the set of attributes which appear in Σ. If the axioms are complete, to solve the implication problem we can simply compute Σ^+ and then test whether $\sigma \in \Sigma^+$.

Another approach for testing implication is to use a nonaxiomatic technique such as the *chase* algorithm [23]. 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 a particular class of probabilistic conditional independencies. Computational properties of both the chase algorithm and inference axioms can be found in [12] and [23].

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 their 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).

V. Nonembedded Dependency

In this section, we study the implication problem for the class of nonembedded (full) probabilistic conditional independency,

$$p(A_1, A_2, A_3, A_4, A_5, A_6) = \frac{p(A_1, A_2, A_3) \cdot p(A_2, A_3, A_4) \cdot p(A_2, A_3, A_5) \cdot p(A_5, A_6)}{p(A_2, A_3) \cdot p(A_2, A_3) \cdot p(A_5)}, \tag{29}$$

called BMVD in our Bayesian database model. 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 [13]. 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).

A. 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 database relation. In this section, we review *two* methods for solving the implication problem of MVDs, namely, the *axiomatic* and *nonaxiomatic* methods.

1) Axiomatization: It is well known [3] that MVDs have a finite complete axiomatization.

Theorem 1: The following inference axioms (M1)–(M7) are both sound and complete for multivalued dependencies (MVDs):

$(M1)$ If $Y \subseteq X,$ then $X \rightarrow\rightarrow Y.$

$(M2)$ If $X \rightarrow\rightarrow Y$ and $Y \rightarrow\rightarrow Z,$ then $X \rightarrow\rightarrow Z - Y.$

$(M3)$ If $X \rightarrow\rightarrow Y,$ and $X \rightarrow\rightarrow Z,$ then $X \rightarrow\rightarrow YZ.$

$(M4)$ If $X \rightarrow\rightarrow Y$ and $X \rightarrow\rightarrow Z,$ then $X \rightarrow\rightarrow Y \cap Z,$ $X \rightarrow\rightarrow Y - Z.$

$(M5)$ If $X \rightarrow\rightarrow Y,$ then $XZ \rightarrow\rightarrow Y.$

$(M6)$ If $X \rightarrow\rightarrow Y$ and $YW \rightarrow\rightarrow Z,$ then $XW \rightarrow\rightarrow Z - (YW).$

$(M7)$ If $X \rightarrow\rightarrow Y,$ then $X \rightarrow\rightarrow R - (XY).$

Axioms (M1)–(M7) are called *reflexivity*, *transitivity*, *union*, *decomposition*, *augmentation*, *pseudotransitivity*, and *complementation*, respectively.

The usefulness of a *sound* axiomatization lies in the ability to derive new dependencies from a given set.

Example 15: 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 $AB \rightarrow\rightarrow G$:

$s_1. AB \rightarrow\rightarrow D$ (given)
$s_2. AB \rightarrow\rightarrow B$ (M1)
$s_3. AB \rightarrow\rightarrow BD$ (M3) from s_1 and s_2
$s_4. BD \rightarrow\rightarrow G$ (given)
$s_5. AB \rightarrow\rightarrow G$ (M2) from s_3 and s_4.

Since the above derivation sequence $s = (s_1, s_2, s_3, s_4, s_5)$ is constructed based on sound axioms, this means that C *logically implies* $AB \rightarrow\rightarrow G$, written:

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

The above example demonstrates that whenever a dependency is derived using sound axioms, the inferred dependency is logically implied by the given input set. However, if the inference axioms are *not* complete, then there is no guarantee that the axioms will derive *all* of the logically implied dependencies. Thus, in this approach the main task in solving the implication problem for a class of dependencies is to construct a set of complete inference axioms.

2) A Nonaxiomatic method—the Chase: Here we want to discuss an alternative method to solve the implication problem for the MVD class of dependencies. 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 $R_1 \cup R_2 \cup \cdots \cup R_n = R$. We say relation r *decomposes losslessly* onto a database scheme $\mathcal{R} = \{R_1, R_2, \cdots, R_n\}$ if

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

It can be easily verified that

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

holds for any decomposition. In other words, every tuple $t \in r$ will also appear in the expression $\pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \cdots \bowtie \pi_{R_n}(r)$. Thereby, for lossless decomposition it is sufficient to show

$$r \supseteq \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \cdots \bowtie \pi_{R_n}(r).$$

That is, to show that *every* tuple in the natural join of the projections is also a tuple in r.

The notion of lossless decomposition can be conveniently expressed by the *project-join mapping* $m_{\mathcal{R}}$ which is a function on relations on R defined by

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

The important point to notice is that saying a relation $r(R)$ decomposes losslessly onto scheme $\mathcal{R}$ is the same as saying that $m_{\mathcal{R}}(r) = r$. 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 of subscripted variables (symbols). The a and b variables are called *distinguished* and *nondistinguished* 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 $A_1A_2 \cdots A_m$, 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 Fig. 16. We obtain a relation from the tableau by substituting domain values for variables. Let T be a tableau and let $V = \{a_1, a_2, \cdots, a_m, b_1, b_2, \cdots\}$ denote the set of its variables. A *valuation* ρ for T is a mapping from V to the Cartesian product $D_1 \times D_2 \times \cdots \times D_m$ 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 $w = \langle v_1 v_2 \cdots v_m \rangle$ is a row in a tableau, we let $\rho(w) = \langle \rho(v_1)\rho(v_2)\cdots\rho(v_m)\rangle$. We then let

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

Example 16: Consider the following valuation ρ:

$$\begin{aligned} &\rho(a_1) = 1, \rho(a_2) = 3, \quad \rho(a_3) = 5, \quad \rho(a_4) = 7 \\ &\rho(b_1) = 4, \quad \rho(b_2) = 8, \quad \rho(b_3) = 2, \quad \rho(b_4) = 7 \\ &\rho(b_5) = 4. \end{aligned} \tag{32}$$

The result of applying ρ to the tableau T in Fig. 16 is the relation r in Fig. 17.

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 variable in every column. Let w_d be the row of all distinguished variables. That is, if $R = A_1 A_2 \cdots A_m$, then $w_d = \langle a_1 a_2 \cdots a_m \rangle$. Row w_d is not necessarily in T. If r is a relation on scheme R, we let

$$T(r) = \{\rho(w_d) | \rho(T) \subseteq r\}.$$

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

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

where $\mathcal{R} = \{R_1, R_2, \cdots, R_n\}$, and $R = R_1 \cup R_2 \cup \cdots \cup R_n$. 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, $w_1, w_2, \cdots, w_n$. 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 $\mathcal{R} = \{R_1 = A_1 A_2, R_2 = A_2 A_3, R_3 = A_3 A_4\}$ 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 Fig. 18.

Lemma 1: [23] Let $\mathcal{R} = \{R_1, R_2, \cdots, R_n\}$ be a set of relation schemes, where $R = R_1 R_2 \cdots R_n$. 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 $r(R)$ decomposes losslessly onto scheme $\mathcal{R}$ is the same as saying that $T_{\mathcal{R}}(r) = r$.

Example 17: Consider the relation $r(A_1 A_2 A_3 A_4)$, as shown on the left side of Fig. 19. The valuation ρ, defined as

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

indicates that $\langle 1368 \rangle$ is in $T_{\mathcal{R}}(r)$. All of $T_{\mathcal{R}}(r)$ is depicted on the right side of Fig. 19. It is easily verified that applying the project-join mapping $m_{\mathcal{R}}$ to the relation $r(R)$ in Fig. 19 also produces the relation on the right side of Fig. 19. That is, $T_{\mathcal{R}}(r) = m_{\mathcal{R}}(r)$.

$T =$

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

Fig. 16. Tableau T on the scheme $A_1 A_2 A_3 A_4$.

$r =$

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

Fig. 17. Relation r obtained as the result of applying ρ in (32) to the tableau T in Fig. 16.

$T =$

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

Fig. 18. Tableau T on $R = A_1 A_2 A_3 A_4$.

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 $T_1 \equiv_{SAT(C)} T_2$, 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 *M-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 $\mathcal{R} = \{R_1, R_2, \cdots, R_q\}$ 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 $w_1, w_2, \cdots, w_q$ (not necessarily distinct) be rows of T that are joinable on $\mathcal{R}$ with result w. Applying the M-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

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

Example 18: Let $C = \{\bowtie \{A_1 A_2, A_2 A_3 A_4\}\}$ and T be the tableau in Fig. 20. Rows w_1 and w_2 are joinable on A_2. We can then apply the M-rule for $\bowtie \{A_1 A_2, A_2 A_3 A_4\}$ in C to rows $w_1 = \langle a_1 a_2 b_1 b_2 \rangle$ and $w_2 = \langle b_3 a_2 a_3 b_4 \rangle$ of T to generate the new row

$$\begin{aligned} w &= w_1(A_1 A_2) \bowtie w_2(A_2 A_3 A_4) \\ &= \langle a_1 a_2 \rangle \bowtie \langle a_2 a_3 b_4 \rangle \\ &= \langle a_1 a_2 a_3 b_4 \rangle. \end{aligned}$$

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

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

Fig. 19. Relation $r(A_1A_2A_3A_4)$ on the left. On the right, the relation $T(r)$, where T is the tableau in Fig. 18.

Tableau $T' = T \cup \{w\}$ in Fig. 21 is the result of this application. Even though rows $w = \langle a_1a_2a_3b_4\rangle$ and $w_3 = \langle b_5b_6a_3a_4\rangle$ are joinable on A_3, we cannot construct the new row $\langle a_1a_2a_3a_4\rangle$ since no M-rule exists in C which applies to attribute A_3.

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

Theorem 2: [23] Let $\mathcal{R} = \{R_1, R_2, \cdots, R_q\}$ and T' be the result of applying the M-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 M-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 19: 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 Fig. 18 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 M-rule for $A_2 \rightarrow\rightarrow A_1$ in C to rows $w_1 = \langle a_1a_2b_1b_2\rangle$ and $w_2 = \langle b_3a_2a_3b_4\rangle$ of $T_\mathcal{R}$ to generate the new row

$$\begin{aligned} w_4 &= w_1(A_1A_2) \bowtie w_2(A_2A_3A_4) \\ &= \langle a_1a_2a_3b_4\rangle. \end{aligned}$$

Tableau $T_\mathcal{R} \cup \{w_4\}$ is depicted in Fig. 21. Similarly, rows w_4 and w_3 are joinable on A_3. We can then apply the M-rule for $A_3 \rightarrow\rightarrow A_4$ in C to rows $w_4 = \langle a_1a_2a_3b_4\rangle$ and $w_3 = \langle b_5b_6a_3a_4\rangle$ to generate the new row

$$\begin{aligned} w_d &= w_4(A_1A_2A_3) \bowtie w_3(A_3A_4) \\ &= \langle a_1a_2a_3a_4\rangle \end{aligned}$$

as shown in Fig. 22. 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 M-rules were applied.

$T =$

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

Fig. 20. 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

Fig. 21. Tableau $T' = T \cup \{\langle a_1a_2a_3b_4\rangle\}$, where T is the tableau in Fig. 20.

$T_\mathcal{R} =$

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

Fig. 22. 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 $w_d = \langle a_1a_2a_3a_4\rangle$ is also in $T_\mathcal{R}$.

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.

B. 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: Let $\mathbf{C}$ be a set of BMVDs on R and $\mathbf{c}$ a single BMVD on R. Then

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

where $C = \{X \rightarrow\rightarrow Y | X \Rightarrow\Rightarrow Y \in \mathbf{C}\}$ 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], [42], $\mathbf{r}(R)$ 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

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

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

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

$(BM1)$ If $Y \subseteq X,$ then $X \Rightarrow\Rightarrow Y.$

$(BM2)$ If $X \Rightarrow\Rightarrow Y$ and $Y \Rightarrow\Rightarrow Z,$ then $X \Rightarrow\Rightarrow Z - Y.$

$(BM3)$ If $X \Rightarrow\Rightarrow Y,$ and $X \Rightarrow\Rightarrow Z,$ then $X \Rightarrow\Rightarrow YZ.$

$(BM4)$ If $X \Rightarrow\Rightarrow Y$ and $X \Rightarrow\Rightarrow Z,$ then $X \Rightarrow\Rightarrow Y \cap Z,$ $X \Rightarrow\Rightarrow Y - Z.$

$(BM5)$ If $X \Rightarrow\Rightarrow Y,$ then $XZ \Rightarrow\Rightarrow Y.$

$(BM6)$ If $X \Rightarrow\Rightarrow Y$ and $YW \Rightarrow\Rightarrow Z,$ then $XW \Rightarrow\Rightarrow Z - (YW).$

$(BM7)$ If $X \Rightarrow\Rightarrow Y,$ then $X \Rightarrow\Rightarrow R - (XY).$

Axiom (BM1) holds trivially for any relation $\mathbf{r}(R)$ with $XY \subseteq R$. We now show that axiom (BM2) is sound. Recall that

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

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 (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 (36) and (37) we have

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

By (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 (M1)–(M7) in Theorem 1. Since the BMVD axioms (BM1)–(BM7) are *sound*, it can immediately be shown that the implication problems coincide in the pair (BMVD,MVD).

Theorem 5: Given the *complete* axiomatization (M1)–(M7) for the MVD class. Then

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

where $\mathbf{C}$ is a set of BMVDs, $C = \{X \rightarrow\rightarrow Y | X \Rightarrow\Rightarrow Y \in \mathbf{C}\}$ 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, 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 $\mathbf{C} \models \mathbf{c}$. ■

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

Corollary 1: The axioms (BM1)–(BM7) 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 [13] 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).

2) 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 [41]. The standard tableau T on a set of attributes $R = A_1 A_2 \cdots A_m$ is augmented with attribute A_p. Each traditional row $w = \langle a_1 a_2 \cdots a_m \rangle$ is appended with probability symbol $p(a_1, a_2, \cdots, a_m)$. That is, a probabilistic tableau $\mathbf{T}$ contains rows $\mathbf{w} = \langle w, p(w) \rangle$. 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 M-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 *B-rule* rule for the BMVD $X \Rightarrow\!\!\Rightarrow Y$ is a means to add the new row $\mathbf{w} = \langle w, p(w)\rangle$ to $\mathbf{T}$, where w is defined in the usual sense according the M-rule for the corresponding MVD $X \rightarrow\!\!\rightarrow Y$, and the probability symbol $p(w)$ is defined as

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

Example 20: Let $\mathbf{C} = \{A_2 \Rightarrow\!\!\Rightarrow A_1,\ A_3 \Rightarrow\!\!\Rightarrow A_4\}$ and consider the tableau $\mathbf{T}_{\mathcal{R}}$ at the top of Fig. 23. It can be seen that rows

$$\mathbf{w}_1 = \langle a_1 a_2 b_1 b_2 p(a_1 a_2 b_1 b_2)\rangle$$

and

$$\mathbf{w}_2 = \langle b_3 a_2 a_3 b_4 p(b_3 a_2 a_3 b_4)\rangle$$

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

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

The new row $\mathbf{w}_4$ is added to $\mathbf{T}_{\mathcal{R}}$, as shown at the top of Fig. 24. Similarly, rows

$$\mathbf{w}_3 = \langle b_3 b_5 a_3 a_4 p(b_3 b_5 a_3 a_4)\rangle \quad \text{and}$$

$$\mathbf{w}_4 = \left\langle a_1 a_2 a_3 b_4 \frac{p(a_1 a_2) p(a_2 a_3 b_4)}{p(a_2)} \right\rangle$$

are joinable on A_3. By (40), the B-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

$$\mathbf{w}_5 = \left\langle a_1 a_2 a_3 a_4 \frac{p(a_1 a_2) p(a_2 a_3) p(a_3 a_4)}{p(a_2) p(a_3)} \right\rangle.$$

The tableau $\mathbf{T}_{\mathcal{R}} \cup \{\mathbf{w}_4\} \cup \{\mathbf{w}_5\}$ is shown at the top of Fig. 24.

The *probabilistic* chase algorithm is now introduced. Given $\mathbf{T}$ and $\mathbf{C}$, apply the B-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 $R = A_1 A_2 \cdots A_m$, and $\mathbf{c}$ be the BMVD $X \Rightarrow\!\!\Rightarrow Y$ on R. Then

$$\mathbf{C} \models \mathbf{c} \Longleftrightarrow \mathbf{w}_d \text{ is a row in } chase_{\mathbf{C}}(\mathbf{T}_{\mathcal{R}})$$

where $\mathcal{R} = \{XY, XZ\}$ is the acyclic hypergraph corresponding to $\mathbf{c}$, and $\mathbf{w}_d$ is defined as

$$\mathbf{w}_d = \left\langle a_1 a_2 \cdots a_m p(a_1, a_2, \cdots, a_m) = \frac{p(xy) \cdot p(xz)}{p(x)} \right\rangle.$$

Proof: ($\Rightarrow$) We first show that the row of all distinguished variables $\langle a_1 a_2 \cdots a_m p(a_1, a_2, \cdots, a_m)\rangle$ must appear in $chase_{\mathbf{C}}(\mathbf{T}_{\mathcal{R}})$. Given $\mathbf{C} \models \mathbf{c}$. By contradiction, suppose that the row $\langle a_1 a_2 \cdots a_m p(a_1, a_2, \cdots, a_m)\rangle$ does not appear in $chase_{\mathbf{C}}(\mathbf{T}_{\mathcal{R}})$. This means that the B-rules corresponding to the BMVDs in $\mathbf{C}$ cannot be applied to the joinable rows to generate the row $\langle a_1 a_2 \cdots a_m p(a_1, a_2, \cdots, a_m)\rangle$. This implies that the M-rules corresponding to the MVDs in $C = \{V \rightarrow\!\!\rightarrow W | V \Rightarrow\!\!\Rightarrow W \in \mathbf{C}\}$ cannot be applied to the joinable rows in $T_{\mathcal{R}}$ to generate the row $\langle a_1 a_2 \cdots a_m\rangle$ of all distinguished variables, where c is the MVD corresponding to the BMVD $\mathbf{c}$. By Theorem 3, the row $\langle a_1 a_2 \cdots a_m\rangle$ 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 $\langle a_1 a_2 \cdots a_m p(a_1, a_2, \cdots, a_m)\rangle$ must appear in $chase_{\mathbf{C}}(\mathbf{T}_{\mathcal{R}})$.

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

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

$T_{\mathcal{R}} =$

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

Fig. 23. 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 at the top 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 bottom.)

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

	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

Fig. 24. Tableaux obtained by adding the new rows $\mathbf{w}_4$ and $\mathbf{w}_5$ is shown on the top of the figure. (The standard use of the corresponding M-rules is shown on the bottom.)

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

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

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,

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

($\Leftarrow$) Given the row $\langle a_1 a_2 \cdots a_m p(a_1, a_2, \cdots, a_m)\rangle$ appears in $chase_{\mathbf{C}}(\mathbf{T}_{\mathcal{R}})$. This means that the B-rules corresponding to the BMVDs in $\mathbf{C}$ can be applied to $\mathbf{T}_{\mathcal{R}}$ to generate the row $\langle a_1 a_2 \cdots a_m p(a_1, a_2, \cdots, a_m)\rangle$. This implies that the M-rules corresponding to the MVDs in $C = \{V \rightarrow\!\!\rightarrow W | V \Rightarrow\!\!\Rightarrow W \in \mathbf{C}\}$ can be applied to the joinable rows in $T_{\mathcal{R}}$ to generate the row $\langle a_1 a_2 \cdots a_m\rangle$ of all distinguished variables, where c is the MVD corresponding to the BMVD $\mathbf{c}$. By Theorem 3, the row

$\langle a_1 a_2 \cdots a_m \rangle$ 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., $p(a_1, a_2, \cdots, a_m)$ can always be factorized according to the BMVD being tested.

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 $R = A_1 A_2 \cdots A_m$, and $\mathbf{c}$ be a single BMVD on R. Then

$$\mathbf{C} \models \mathbf{c} \Longleftrightarrow \langle a_1 a_2 \cdots a_m \rangle \text{ is a row in } chase_C(T_{\mathcal{R}}),$$

where $C = \{X \rightarrow\rightarrow Y | X \Rightarrow\Rightarrow Y \in \mathbf{C}\}$ 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,

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

By Theorem 3,

$$C \models c \Longleftrightarrow \langle a_1 a_2 \cdots a_m \rangle \text{ is a row in } chase_C(T_{\mathcal{R}}).$$

The claim follows immediately. ■

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.

C. 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, here we would like to take this opportunity to show that every conflict-free set $\mathbf{C}$ of BMVDs is *equivalent* to a Bayesian acyclic-join dependency (BAJD), $\otimes\mathcal{R}$. That is, whenever any probabilistic relation satisfies all the BMVDs in $\mathbf{C}$, then it also satisfies the BAJD $\otimes\mathcal{R}$, and vice versa.

Theorem 8: Let $\mathbf{C}$ denote a conflict-free set of BMVDs. Let $C = \{X \rightarrow\rightarrow Y | X \Rightarrow\Rightarrow Y \in \mathbf{C}\}$ 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 [4], 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 $X \rightarrow\rightarrow Y$ 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 $C \models X \rightarrow\rightarrow Y$. ■.

Theorem 8 indicates that every conflict-free set of nonembedded probabilistic dependencies is equivalent to a Bayesian acyclic join dependency.

VI. Embedded Dependencies

We now examine the implication problem for *embedded* dependencies. As shown in Fig. 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 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) which represent the general classes of probabilistic conditional independency and embedded multivalued dependency.

A. Conflict-Free Embedded Dependencies

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

The class of conflict-free BEMVDs plays a key role in the design of Bayesian networks. Recall that a set of BEMVDs is *conflict-free* if they can be faithfully represented by a *single* DAG. We can use the *d-separation* method [31] to infer BEMVDs from a DAG. One desirable property of the conflict-free BEMVD class is that every conflict-free set of BEMVDs has a DAG as its *perfect-map*.

The class of conflict-free BEMVD is a *special case* of the general BEMVD class, as shown in Fig. 1. This special class of probabilistic dependencies has 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 $XYZW = R$.

(BE1) If $X \Rightarrow\Rightarrow Y$ then $X \Rightarrow\Rightarrow ZW$,

(BE2) If $X \Rightarrow\Rightarrow YW|Z$, then $X \Rightarrow\Rightarrow Y|Z$,

(BE3) If $X \Rightarrow\Rightarrow YZ$, then $XZ \Rightarrow\Rightarrow Y$,

(BE4) If $X \Rightarrow\Rightarrow Y|Z$ and $XZ \Rightarrow\Rightarrow Y$, then $X \Rightarrow\Rightarrow Y$.

The axioms (BE1)–(BE4) are respectively called *symmetry*, *decomposition*, *weak union*, and *contraction*. Clearly, 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 classes of conflict-free BEMVD and conflict-free EMVD.

It is known that the following EMVD inference axioms are sound [3], [38], where X, Y, Z, W be pairwise disjoint subsets of R such that $XYZW = R$.

(E1) If $X \rightarrow\rightarrow Y$, then $X \rightarrow\rightarrow ZW$,

(E2) If $X \rightarrow\rightarrow YW|Z$, then $X \rightarrow\rightarrow Y|Z$,

(E3) If $X \rightarrow\rightarrow YZ$, then $XZ \rightarrow\rightarrow Y$,

(E4) If $X \rightarrow\rightarrow Y|Z$ and $XZ \rightarrow\rightarrow Y$, then $X \rightarrow\rightarrow Y$.

Theorem 10: Given the *complete* axiomatization (BE1)–(BE4) for the CF-BEMVD class. Then

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

where $\mathbf{C}$ is a conflict-free set of EMVDs, $C = \{X \rightarrow\rightarrow Y|Z \,|\, X \Rightarrow\Rightarrow Y|Z \in \mathbf{C}\}$ is the corresponding conflict-free set of BEMVDs, and c is the EMVD corresponding to the 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 $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{c})$ of the BEMVD $\mathbf{c}$ from the conflict-free set $\mathbf{C}$ of BEMVDs using the inference axioms (BE1)–(BE4). The above discussion demonstrates that the corresponding inference axioms (E1)–(E4) are *sound* for deriving new EMVDs. This means that there is a derivation sequence $s = (s_1, s_2, \cdots, c)$ of the EMVD c from the conflict-free set C of EMVDs using the inference axioms (E1)-(E4), such that s parallels $\mathbf{s}$. That is, $C \vdash c$. We obtain our desired result since $C \vdash c$ 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). Conversely, we want to know whether

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

is also true for this pair of dependencies. It was shown that there exists a complete axiomatization for conflict-free EMVDs [31].

Theorem 11: [31] The axioms (E1)–(E4) are *complete* for the class of *conflict-free* EMVD.

Based on this theorem, the following result is immediate.

Theorem 12: Given the complete axiomatization (E1)–(E4) for the CF-EMVD class. Then

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

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

Proof: The proof follows from a similar argument given in the Proof of Theorem 10. ■

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

$$\mathbf{C} \models \mathbf{c} \Longleftrightarrow C \models c \tag{41}$$

holds for the pair (conflict-free BEMVD, conflict-free EMVD). As already mentioned, the class of conflict-free BEMVDs is the basis for constructing a Bayesian network. However, conflict-free EMVDs have traditionally been ignored in relational databases. The above observation indicates that the special class of *conflict-free* EMVDs is equally useful in the design and implementation of traditional database applications.

B. 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). Similarly, EMVD is the general class of multivalued dependencies. Before we study BEMVDs, we first examine the implication problem for EMVDs.

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

The chase algorithm also does *not* solve the implication problem for the EMVD class. If $C \not\models c$, then the chase algorithm can continue forever. The reason is that, by definition, a M-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. Thus, 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 does not terminate and 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 [12]. This means that if $C \models c$, then the row of all distinguished variables will eventually be generated. The generation of the row of all as can be used as a stopping criterion.

Example 21: Suppose we wish to verify that $C \models c$, where $C = \{A_1 \rightarrow\rightarrow A_3|A_4,\ A_2 \rightarrow\rightarrow A_3|A_4,\ A_3A_4 \rightarrow\rightarrow A_1|A_2\}$ and c is the the EMVD $A_1A_2 \rightarrow\rightarrow A_3$. The initial tableau $T_{\mathcal{R}}$ is constructed according to c, as shown in Fig. 25 (left). We can apply the M-rule corresponding to the EMVD $A_1 \rightarrow\rightarrow A_3|A_4$ in C to joinable rows $w_1 = \langle a_1a_2a_3b_1\rangle$ and $w_2 = \langle a_1a_2b_2a_4\rangle$ to generate the new row $w_3 = \langle a_1b_3a_3a_4\rangle$, as shown in Fig. 25 (right). Similarly, we can apply the M-rule corresponding to the EMVD $A_2 \rightarrow\rightarrow A_3|A_4$ in C to joinable rows $w_1 = \langle a_1a_2a_3b_1\rangle$ and $w_2 = \langle a_1a_2b_2a_4\rangle$ to generate the new row $w_4 = \langle b_4a_2a_3a_4\rangle$, as shown in Fig. 25 (right). Finally, we can obtain the row $\langle a_1a_2a_3a_4\rangle$ of all distinguished variables by applying the M-rule corresponding to the MVD $A_3A_4 \rightarrow\rightarrow A_1|A_2$ in C to joinable rows w_3 and w_4. Therefore, $C \models c$.

For over a decade, considerable effort was put forth in the database research community to show that the implication problem for EMVDs is in fact *unsolvable*. Herrmann [17] recently succeeded in showing this elusive result.

Theorem 14: [17] The implication problem for the general 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 (BE1)–(BE4) could solve

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

Fig. 25. 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 $\langle a_1a_2a_3a_4\rangle$ of all distinguished variables appears in $chase_{\mathbf{C}}(T_{\mathcal{R}})$ indicating $C \models c$.

the implication problem for probabilistic conditional independency (BEMVD) in general. This conjecture was refuted [37], [46].

Theorem 15: [37], [46] 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.

As with the other classes of 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.

Example 22: 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 showed that $\mathbf{C} \models \mathbf{c}$. Now consider the set $C = \{X \rightarrow\!\!\rightarrow Y|Z|X \Rightarrow\!\!\Rightarrow Y|Z \in \mathbf{C}\}$ 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 Fig. 26. 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. That is, $C \not\models c$.

Example 22 indicates that

$$\mathbf{C} \models \mathbf{c} \not\Longrightarrow C \models c. \tag{42}$$

Example 23: Consider the set $C = \{A_1 \rightarrow\!\!\rightarrow A_3|A_4, A_2 \rightarrow\!\!\rightarrow A_3|A_4, A_3A_4 \rightarrow\!\!\rightarrow A_1|A_2\}$ of EMVDs, and let c be the single EMVD $A_1A_2 \rightarrow\!\!\rightarrow A_3$. The chase algorithm was used in Example 21 to show that $C \models c$. Now consider the corresponding set of BEMVDs $\mathbf{C} = \{A_1 \Rightarrow\!\!\Rightarrow A_3|A_4, A_2 \Rightarrow\!\!\Rightarrow A_3|A_4, A_3A_4 \Rightarrow\!\!\Rightarrow A_1|A_2\}$ 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 Fig. 27 satisfies all of the BEMVDs in $\mathbf{C}$ but does not satisfy the BEMVD $\mathbf{c}$. Therefore, $\mathbf{C} \not\models \mathbf{c}$.

Example 23 indicates that

$$\mathbf{C} \models \mathbf{c} \not\Longleftarrow C \models c. \tag{43}$$

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

C. The Role of Solvability

We have shown that

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

holds for the pairs (BMVD, MVD) in Theorem 5, (Conflict-free BMVD, Conflict-free MVD) in Theorem 5, and (Conflict-free BEMVD, Conflict-free EMVD) in (41). That is, the implication problems coincide in these three pairs of classes. However, Examples 22 and 23 demonstrate that

$r(A_1A_2A_3A_4) =$

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

Fig. 26. Relation r satisfies all of the EMVDs in C but does not satisfy the EMVD c, where C and c are defined in Example 22. Therefore, $C \not\models c$.

$\mathbf{r}(A_1A_2A_3A_4) =$

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

Fig. 27. 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 23. Therefore, $\mathbf{C} \not\models \mathbf{c}$.

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

The implication problems for each class in the first three pairs are *solvable*. However, the implication problem for the general EMVD class in the pair (BEMVD, EMVD) is *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

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

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

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

In [37], Studeny studied the relationship between the implication problems in the pair (BEMVD, EMVD), namely, probabilistic conditional independency (BEMVD) and embedded multivalued dependency. Based on Conjecture 1(i), his observation

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

would indicate that the implication problem for the general class of probabilistic conditional independency is *unsolvable*. Similarly, based on Conjecture 1(ii), his observation

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

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

A successful proof of this conjecture would provide a proof that the implication problems for EMVD and BEMVD (probabilistic conditional independency) are both unsolvable.

VII. Conclusion

The results of this paper and our previous work [42], [44], [45], 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], [19], [35], and as to whether a set of *pairwise consistent* distributions (relations) are indeed marginal distributions from the same joint probability distribution [4], [10]. Even the recent attempts to generalize the standard Bayesian database model, including *horizontal independencies* [6], [44], *complex-values* [20], [44], and *distributed* Bayesian networks [7], [43], [47], parallel the development of *horizontal dependencies* [11], *complex-values* [1], [18], and *distributed* databases [8] 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 [13] (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 pair (BEMVD, EMVD), namely, the general classes of probabilistic conditional independencies and *embedded* multivalued dependencies. The implication problem for the general EMVD class is *unsolvable* [17], as is the general class of probabilistic conditional independencies. 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.

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*. New York: Springer-Verlag, 1989, vol. 361.

[2] W. W. Armstrong, "Dependency structures of database relationships," in *Proc. IFIP 74*. Amsterdam, The Netherlands, 1974, pp. 580–583.

[3] C. Beeri, R. Fagin, and J. H. Howard, "A complete axiomatization for functional and multivalued dependencies in database relations," in *Proc. ACM-SIGMOD Int. Conf. Management of Data*, 1977, pp. 47–61.

[4] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis, "On the desirability of acyclic database schemes," *J. ACM*, vol. 30, no. 3, pp. 479–513, July 1983.

[5] C. Berge, *Graphs and Hypergraphs*. Amsterdam, The Netherlands: North-Holland, 1976.

[6] C. Boutiliere, N. Friedman, M. Goldszmidt, and D. Koller, "Context-specific independence in bayesian networks," in *12th Conf. Uncertainty in Artificial Intelligence*. San Mateo, CA, 1996, pp. 115–123.

[7] C. J. Butz and S. K. M. Wong, "Recovery protocols in multi-agent probabilistic reasoning systems," in *Int. Database Engineering and Applications Symp.*. Piscataway, NJ, 1999, pp. 302–310.

[8] S. Ceri and G. Pelagatti, *Distributed Databases: Principles & Systems*. New York: McGraw-Hill, 1984.

[9] E. F. Codd, "A relational model of data for large shared data banks," *Commun. ACM*, vol. 13, no. 6, pp. 377–387, June 1970.

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

[11] R. Fagin, "Normal forms and relational database operators," in *Proc. ACM-SIGMOD Int. Conf. Management of Data*, 1979, pp. 153–160.

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

[13] D. Geiger and J. Pearl, "Logical and algorithmic properties of conditional independence," Univ. California, Tech. Rep. R-97-II-L, 1989.

[14] ——, "Logical and algorithmic properties of conditional independence and graphical models," *Ann. Stat.*, vol. 21, no. 4, pp. 2001–2021, 1993.

[15] D. Geiger, T. Verma, and J. Pearl, "Identifying independence in bayesian networks," Univ. California, Tech. Rep. R-116, 1988.

[16] P. Hajek, T. Havranek, and R. Jirousek, *Uncertain Information Processing in Expert Systems*. Boca Raton, FL: CRC, 1992.

[17] C. Herrmann, "On the undecidability of implications between embedded multivalued database dependencies," *Inf. Comput.*, vol. 122, no. 2, pp. 221–235, 1995.

[18] G. Jaeschke and H. J. Schek, "Remarks on the algebra on non first normal form relations," in *Proc. 1st ACM SIGACT-SIGMOD Symp. Principles of Database Systems*, 1982, pp. 124–138.

[19] F. V. Jensen, S. L. Lauritzen, and K. G. Olesen, "Bayesian updating in causal probabilistic networks by local computation," *Comput. Stat. Quarterly*, vol. 4, pp. 269–282, 1990.

[20] D. Koller and A. Pfeffer, "Object-oriented bayesian networks," in *13th Conf. Uncertainty in Artificial Intelligence*. San Mateo, CA, 1997, pp. 302–313.

[21] T. T. Lee, "An information-theoretic analysis of relational databases—Part I: Data dependencies and information metric," *IEEE Trans. Software Eng.*, vol. SE-13, no. 10, pp. 1049–1061, 1987.

[22] Y. E. Lien, "On the equivalence of database models," *J. ACM*, vol. 29, no. 2, pp. 336–362, Oct. 1982.

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

[24] D. Maier, A. O. Mendelzon, and Y. Sagiv, "Testing implications of data dependencies," *ACM Trans. Database Syst.*, vol. 4, no. 4, pp. 455–469, 1979.

[25] F. Malvestuto, "A unique formal system for binary decompositions of database relations, probability distributions and graphs," *Inf. Sci.*, vol. 59, pp. 21–52, 1992.

[26] ——, "A complete axiomatization of full acyclic join dependencies," *Inf. Process. Lett.*, vol. 68, no. 3, pp. 133–139, 1998.

[27] A. Mendelzon, "On axiomatizing multivalued dependencies in relational databases," *Journal of the ACM*, vol. 26, no. 1, pp. 37–44, 1979.

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

[29] D. Parker and K. Parsaye-Ghomi, "Inference involving embedded multivalued dependencies and transitive dependencies," in *Proc. ACM-SIGMOD Int. Conf. Management of Data*, 1980, pp. 52–57.

[30] A. Paz, "Membership algorithm for marginal independencies," Univ. California, Tech. Rep. CSD-880 095, 1988.

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

[32] J. Pearl, D. Geiger, and T. Verma, "Conditional independence and its representations," *Kybernetica*, vol. 25, no. 2, pp. 33–44, 1989.

[33] J. Pearl and A. Paz, "Graphoids: Graph-based logic for reasoning about relevance relations," Univ. California, Tech. Rep. R-53-L, 1985.

[34] Y. Sagiv and F. Walecka, "Subset dependencies and a complete result for a subclass of embedded multivalued dependencies," *J. ACM*, vol. 20, no. 1, pp. 103–117, 1982.

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

[36] M. Studeny, "Multiinformation and the problem of characterization of conditional-independence relations," *Problems of Control and Information Theory*, vol. 18, no. 1, pp. 3–16, 1989.

[37] ——, "Conditional independence relations have no finite complete characterization," in *11th Prague Conf. Information Theory, Statistical Decision Foundation and Random Processes*. Norwell, MA, 1990, pp. 377–396.

[38] K. Tanaka, Y. Kambayashi, and S. Yajima, "Properties of embedded multivalued dependencies in relational databases," *Trans. IECE Jpn.*, vol. E62, no. 8, pp. 536–543, 1979.

[39] T. Verma and J. Pearl, "Causal networks: Semantics and expressiveness," in *4th Conf. Uncertainty in Artificial Intelligence*, St. Paul, MN, 1988, pp. 352–359.

[40] W. X. Wen, "From relational databases to belief networks," in *7th Conf. Uncertainty in Artificial Intelligence*. San Mateo, CA, 1991, pp. 406–413.

[41] S. K. M. Wong, "Testing implication of probabilistic dependencies," in *12th Conf. Uncertainty in Artificial Intelligence*. San Mateo, CA, 1996, pp. 545–553.

[42] ——, "An extended relational data model for probabilistic reasoning," *J. Intell. Inf. Syst.*, vol. 9, pp. 181–202, 1997.

[43] S. K. M. Wong and C. J. Butz, "Probabilistic reasoning in a distributed multi-agent environment," in *3rd Int. Conf. Multi-Agent Systems*. Piscataway, NJ, 1998, pp. 341–348.

[44] ——, "Contextual weak independence in bayesian networks," in *15th Conf. Uncertainty in Artificial Intelligence*. San Mateo, CA, 1999, pp. 670–679.

[45] S. K. M. Wong, C. J. Butz, and Y. Xiang, "A method for implementing a probabilistic model as a relational database," in *11th Conf. Uncertainty in Artificial Intelligence*. San Mateo, CA, 1995, pp. 556–564.

[46] S. K. M. Wong and Z. W. Wang, "On axiomatization of probabilistic conditional independence," in *10th Conf. Uncertainty in Artificial Intelligence*. San Mateo, CA, 1994, pp. 591–597.

[47] Y. Xiang, "A probabilistic framework for cooperative multi-agent distributed interpretation and optimization of communication," *Artif. Intell.*, vol. 87, pp. 295–342, 1996.

S. K. M. Wong received the B.Sc. degree from the University of Hong Kong in 1963, and the M.A. and Ph.D. degrees in theoretical physics from the University of Toronto, Toronto, ON, Canada, in 1964 and 1968, respectively.

Before he joined the Department of Computer Science at the University of Regina, Regina, SK, Canada, in 1982, he worked in various computer related industries. Currently, he is a Professor of Computer Science at the University of Regina. His research interests include uncertainty reasoning, information retrieval, database systems, and data mining.

C. J. Butz received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the University of Regina, Regina, SK, Canada, in 1994, 1996, and 2000, respectively.

In 2000, he joined the School of Information Technology and Engineering at the University of Ottawa, Ottawa, ON, Canada, as an Assistant Professor. His research interests include uncertainty reasoning, database systems, information retrieval, and data mining.

D. Wu received the B.Sc. degree in computer science from the Central China Normal University, Wuhan, China, in 1994, and the M.Eng. degree in information science from Peking University, Beijing, China, in 1997. He is currently a doctoral student at the University of Regina, Regina, SK, Canada. His research interests include uncertainty reasoning and database systems.