On the Implication Problem in Granular Knowledge Systems

C.J. Butz and J. Liu
Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: {butz,liuji114}@cs.uregina.ca

***Abstract*— Previous work seemed to suggest that the logical implication of non-numeric constraints in database systems exactly coincides with that of numeric constraints in probabilistic expert systems, provided that restrictions are imposed on the given set of constraints. In this paper, we dispel this suggestion by showing that the logical implication differs in database systems and probabilistic expert systems even with a restriction imposed on the given set of constraints. Our restriction is a granular representation, called hierarchical Markov networks, which have shown great promise as a new representation of Bayesian networks. This work is then significant as it provides a lower upper-bound on where the logical implication of non-numeric and numeric constraints diverge.**

I. Introduction

In the design of both database systems [1], [7] and probabilistic expert systems [8], [9], a crucial issue to consider is the implication problem. The *implication problem* [11] is to test whether a given set Σ of constraints logically implies another independency σ. In this investigation, constraints in the database setting are considered to be *embedded multivalued dependencies* (EMVD) [3], while constraints in the probabilistic setting are considered to be *probabilistic condition independencies* (CI) [11]. The implication problem has been separately studied in both relational databases, including [1], [7], and in Bayesian networks [4], [5], [8], [10].

In a comprehensive study of the implication problem for probabilistic conditional independencies [11], it is emphasized that *Bayesian networks* (BNs) and relational databases coincide on *solvable* classes of independencies. That study suggested that the implication problem for these two closely related systems differs only in *unsolvable* classes of independencies. This meant that there is no *real* difference between Bayesian networks and relational databases [13], in the sense that only *solvable* classes of independencies are useful in the design and implementation of these knowledge systems.

In [12], we introduced a new kind of probabilistic network, called a *hierarchical Markov network* (HMN). A HMN is a rooted hierarchy of MNs. We gave an algorithm to transform a BN into a canonical HMN. The main result of [12] is that the constructed HMN is *unique* and *equivalent* to the input BN. Since HMNs are a *faithful* representation of BNs, a query may be optimized using independencies in a HMN that otherwise would have gone unrepresented in a traditional approach. The above remarks seem to suggest that HMNs are a very natural representation of probabilistic knowledge with several desirable properties. It should be made clear, however, that BNs and HMNs are not adversarial representations. Instead BNs and HMNs form an enviable pair for the acquisition and inference of probabilistic knowledge.

In this paper, we show that the logical implication of EMVD and CI differs even with a restriction imposed on the given set of constraints. Our restriction is that the input set of constraints define a HMN. This work is then significant as it provides a lower upper-bound on where the logical implication of non-numeric and numeric constraints diverge compared to the work in [11].

This paper is organized as follows. In Section II, we review relational databases and Bayesian networks. The implication problem is presented in Section III. In Section IV, we show that the implication of EMVD and CI differs in HMNs. The conclusion is given in Section V.

II. Background Knowledge

We begin our discussion by reviewing relational databases and Bayesian networks.

A. Relational Databases

To clarify the notations, we give a brief review of the standard relational database model [7]. The relational concepts presented here are then generalized to express the probabilistic network concepts in Section II-B.

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

$$r(R) = \begin{array}{|cccc|}\hline A_1 & A_2 & \ldots & A_m \\ \hline t_1(A_1) & t_1(A_2) & \ldots & t_1(A_m) \\ t_2(A_1) & t_2(A_2) & \ldots & t_2(A_m) \\ \vdots & \vdots & \vdots & \vdots \\ t_s(A_1) & t_s(A_2) & \ldots & t_s(A_m) \\ \hline \end{array}$$

Fig. 1. A relation r on the scheme $R = \{A_1, A_2, \ldots, A_m\}$.

$$r(ABCD) = \begin{array}{|cccc|}\hline A & B & C & D \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ \hline \end{array}$$

Fig. 2. A relation r on scheme $R = ABCD$.

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 $R = \{A_1, A_2, \ldots, A_m\}$ is written as simply $A_1 A_2 \ldots A_m$. A relation r on scheme R is denoted by either $r(R)$ or $r(A_1 A_2 \ldots A_m)$. The singleton set $\{A\}$ is written as A and the concatenation XY is used to denote set union $X \cup Y$. For example, a relation $r(R)$ on $R = ABCD$ is shown in Figure 2, where the domain of each attribute in R is $\{0, 1\}$.

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

$$\pi_X(r) = \{\ t(X) \mid t \in r\ \}. \tag{1}$$

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

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

We can introduce the key notion of EMVD [3].

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:

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

Example 1: Consider the relation $r(ABCD)$ in Figure 2. As shown in Figure 3, relation $r(ABCD)$ 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. 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,

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

Fig. 3. Relation $r(ABCD)$ in Figure 2 satisfies the EMVD $B \rightarrow\rightarrow A|C$, since $\pi_{ABC}(r) = \pi_{AB}(r) \bowtie \pi_{BC}(r)$.

for any two tuples t_1 and t_2 in r with $t_1(X) = t_2(X)$, there exists a tuple t_3 in r with:

$$t_3(XY) = t_1(XY) \text{ and } t_3(Z) = t_2(Z). \tag{2}$$

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

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

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

$$r(R) = \pi_{XY}(r) \bowtie \pi_{XZ}(r). \tag{3}$$

Example 2: The relation $r(ABC)$ in Figure 4 satisfies the MVD $B \rightarrow\rightarrow A$, since $r(ABC) = \pi_{AB}(r) \bowtie \pi_{BC}(r)$. □

$$r(ABC) = \begin{array}{|ccc|}\hline A & B & C \\ \hline 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|cc|}\hline A & B \\ \hline 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ \hline \end{array} \bowtie \begin{array}{|cc|}\hline B & C \\ \hline 0 & 0 \\ 0 & 1 \\ 1 & 1 \\ \hline \end{array}$$

Fig. 4. Relation $r(ABC)$ satisfies the MVD $B \rightarrow\rightarrow A$.

There is a 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 [2]. In these situations, the acyclic hypergraph is called a *perfect-map* [2]. 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 $R = A_1 A_2 A_3 A_4 A_5 A_6$:

$$\begin{aligned} C \ = \{ & A_2 A_3 \rightarrow\rightarrow A_1,\ A_2 A_3 \rightarrow\rightarrow A_4,\ A_2 A_3 \rightarrow\rightarrow A_5 A_6, \\ & A_5 \rightarrow\rightarrow A_1 A_2 A_3 A_4,\ A_5 \rightarrow\rightarrow A_6,\ A_2 A_3 A_5 \rightarrow\rightarrow A_1\}. \end{aligned} \tag{4}$$

This set of MVDs can be *faithfully* represented by the acyclic hypergraph $\mathcal{R}$ in Figure 5. According to the separation method for inferring MVDs from an acyclic hypergraph, every MVD in C can be inferred from $\mathcal{R}$. Obviously, every MVD logically implied by C can then be inferred from $\mathcal{R}$, and every MVD inferred from $\mathcal{R}$ is logically implied by C. Thus, the acyclic hypergraph $\mathcal{R}$ in Figure 5 is a *perfect-map* of the set C of MVDs in Equation (4). □

It is important to realize that there are some sets of MVDs which cannot be faithfully represented by a single acyclic hypergraph.

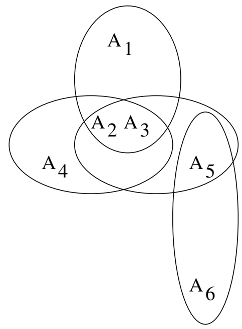

Fig. 5. An acyclic hypergraph $\mathcal{H} = \{A_1A_2A_3,\ A_2A_3A_4,\ A_2A_3A_5,\ A_5A_6\}$.

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

$$C \;=\; \{\ A_1 \rightarrow\rightarrow A_2,\ A_3 \rightarrow\rightarrow A_2\ \}. \tag{5}$$

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 Equation (5) is a member of the MVD class, but is not a member of the conflict-free MVD class.

B. Bayesian Networks

Let us now review some basic notions in Bayesian networks [8], [9].

Let $R = \{A_1, A_2, \ldots, A_m\}$ denote a finite set of discrete variables (attributes). Each variable A_i is associated with a finite domain D_{A_i}. Let D be the Cartesian product of the domains D_{A_i}, $1 \leq i \leq m$. A *joint probability distribution* (jpd) [6], [8] on D is a function p on D, $p : D \rightarrow [0, 1]$. That is, this function p assigns to each tuple $t \equiv \ < t(A_1), t(A_2), \ldots, t(A_m) > \ \in D$ a real number $0 \leq p(t) \leq 1$ and p is normalized, namely, $\sum_{t \in D} p(t) = 1$. For convenience, we write a joint probability distribution p as $p(A_1, A_2, \ldots, A_m)$ over the set R of variables. In particular, we use $p(a_1, a_2, \ldots, a_m)$ to denote a particular value of $p(t) = p(< t(A_1), t(A_2), \ldots, t(A_m) >)$. That is, $p(a_1, a_2, \ldots, a_m)$ denotes the probability value $p(< t(A_1), t(A_2), \ldots, t(A_m) >)$ of the function p for a particular *instantiation* of the variables $A_1, A_2, \ldots, A_m$. In general, a *potential* [6], [9] is a function q on D such that $q(t)$ is a nonnegative real number and $\sum_{t \in D} q(t)$ is positive, i.e., at least one $q(t) > 0$.

We now introduce the fundamental notion of *probabilistic conditional independency* [11]. 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

$$p(y \mid x, z) \;=\; p(y \mid x), \tag{6}$$

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

$$p(y, x, z) \;=\; \frac{p(y, x) \cdot p(x, z)}{p(x)}. \tag{7}$$

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 $p(A_1, A_2, \ldots, A_m)$ can always be written as:

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

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:

$$\begin{aligned} \mathbf{C} \;=\; & \{\ I(A_1, \emptyset, \emptyset),\ I(A_2, A_1, \emptyset),\ I(A_3, A_1, A_2), \\ & I(A_4, A_2A_3, A_1),\ I(A_5, A_2A_3, A_1A_4), \\ & I(A_6, A_5, A_1A_2A_3A_4)\ \}. \end{aligned} \tag{8}$$

Equivalently, we have:

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

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:

$$\begin{aligned} & p(A_1, A_2, A_3, A_4, A_5, A_6) \\ = \ & p(A_1) \cdot p(A_2|A_1) \cdot p(A_3|A_1) \cdot p(A_4|A_2, A_3) \cdot \\ & p(A_5|A_2, A_3) \cdot p(A_6|A_5). \end{aligned} \tag{9}$$

We can represent all of the probabilistic conditional independencies satisfied by this joint distribution by the *directed acyclic graph* (DAG) shown in Figure 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* [8]. □

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)

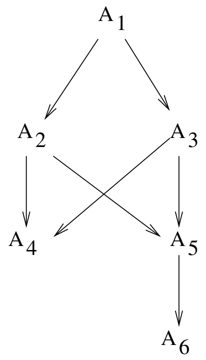

Fig. 6. A DAG D representing all of the probabilistic conditional independencies satisfied by the joint distribution defined by Equation 9

Markov network. A *Markov network* (MN) [6] 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 [6], [8] 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 [2].) For example, the DAG in Figure 6 can be transformed into the acyclic hypergraph depicted in Figure 5. *Local computation* procedures [13] can be applied to transform the conditional probability distributions into marginal distributions defined over the acyclic hypergraph. The joint probability distribution in Equation (9) can be rewritten, in terms of marginal distributions over the acyclic hypergraph in Figure 5, as:

$$p(A_1, A_2, A_3, A_4, A_5, A_6) \ = \ \frac{p(A_1, A_2, A_3) \cdot p(A_2, A_3, A_4) \cdot p(A_2, A_3, A_5) \cdot p(A_5, A_6)}{p(A_2, A_3) \cdot p(A_2, A_3) \cdot p(A_5)}. \quad (10)$$

The Markov network representation of probabilistic knowledge in Equation (10) is typically used for inference in many practical applications [9].

III. The Implication Problem

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

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

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

$$\sum \ \models \ \sigma. \quad (11)$$

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

(i) a member of $\sum$, or
(ii) 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 $\sum$. If the axioms are complete, to solve the implication problem we can simply compute $\sum^+$ and then test whether $\sigma \in \sum^+$.

Another approach for testing implication is to use a nonaxiomatic technique such as the *chase* algorithm [1], [7]. The chase algorithm in relational database model is a powerful tool to obtain many nontrivial results.

For common notation with EMVDs, we now write the CI $I(Y, X, Z)$ as

$$X \ \Rightarrow\Rightarrow \ Y \mid Z.$$

For any set $\mathbf{C}$ of *probabilistic* dependencies, there is a *corresponding* set C of *data* dependencies, namely,

$$C \ = \ \{X \rightarrow\rightarrow Y|Z \mid X \Rightarrow\Rightarrow Y|Z \ \in \mathbf{C}\}.$$

Since we advocate that Bayesian networks are a *generalization* of the relational database model, an immediate question to answer is:

> *Do the implication problems coincide in these two database models?*

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

$$\mathbf{C} \models \mathbf{c} \iff C \models c, \quad (12)$$

holds, where $\mathbf{c}$ is a CI and c is the corresponding EMVD. In particular, we study this question with respect to four classes of constraints:

(1) fixed context (nonembedded),

(2) DAG,

(3) acyclic hypergraph,

(4) no restriction.

The restriction of fixed-context in class (1) means that every constraint involves the same set of variables (attributes). For example, $\{A \Rightarrow\Rightarrow B|C,\ B \Rightarrow\Rightarrow A|C\}$ have the same fixed context ABC, while $\{A \Rightarrow\Rightarrow B|C,\ B \Rightarrow\Rightarrow A|CD\}$ do not. The restriction in class (2) means that the input set of constraints can be represented by a single DAG. For instance, the set of CIs in Example 5 define a DAG, but $\{A \Rightarrow\Rightarrow B|C,$ $B \Rightarrow\Rightarrow A|CD\}$ do not. Similarly, the restriction imposed in class (3) is that there must exist an acyclic hypergraph to represent the input set of constraints. Examples of constraints in this class and not in this class were provided in the review of relational databases.

Our analysis in [11] showed that

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

holds for classes (1), (2) and (3). However, Studeny [10] demonstrated that Equation (12) does *not* hold when no restrictions are imposed on the input set of constraints. These results are summarized in Table I.

TABLE I

PREVIOUS WORK SEEMS TO IMPLY THAT THE LOGICAL IMPLICATION OF CI AND EMVD COINCIDES AS LONG AS A RESTRICTION IS IMPOSED ON THE INPUT SET OF CONSTRAINTS.

Restriction	Logical implication of CI and EMVD coincides
fixed context	yes
DAG	yes
acyclic hypergraph	yes
none	no

IV. Hierarchical Markov Networks

Hierarchical Markov networks (HMNs) [12] were recently introduced as a new kind of probabilistic network.

Let $\mathcal{H}$ be a MN on $R = \{A_1 A_2 \cdots A_m\}$. The *context* of $\mathcal{H}$, denoted $context(\mathcal{H})$, is the set of nodes on which $\mathcal{H}$ is defined, i.e., $context(\mathcal{H}) = R$. Let $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_l$ be a set of MNs. We call $\mathcal{H}_j$ a *descendant* of $\mathcal{H}_i$, if $context(\mathcal{H}_j) \subseteq context(\mathcal{H}_i)$. If $\mathcal{H}_j$ is a descendant of $\mathcal{H}_i$, then we call $\mathcal{H}_i$ an *ancestor* of $\mathcal{H}_j$. We call $\mathcal{H}_j$ a *child* of $\mathcal{H}_i$, if $\mathcal{H}_j$ is a descendant of $\mathcal{H}_i$, and there does not exist a $\mathcal{H}_k$ such that $context(\mathcal{H}_j) \subseteq context(\mathcal{H}_k) \subseteq context(\mathcal{H}_i)$. If $\mathcal{H}_j$ is a child of $\mathcal{H}_i$, then we call $\mathcal{H}_i$ a *parent* of $\mathcal{H}_j$.

A *hierarchical Markov network* (HMN), denoted $\mathbf{H} = \{\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_l\}$, is a hierarchy of MNs satisfying the following three conditions:

(i) there is only one *root* MN in $\mathbf{H}$,

(ii) if $\mathcal{H}_j$ is a child of $\mathcal{H}_i$, then there exists a hyperedge $h_i \in \mathcal{H}_i$ such that $context(\mathcal{H}_j) \subseteq h_i$, and

(iii) if $\mathcal{H}_j$ and $\mathcal{H}_k$ are two distinct children of $\mathcal{H}_i$, then $\mathcal{H}_j$ and $\mathcal{H}_k$ are contained by *distinct* hyperedges of $\mathcal{H}_i$.

We call $\mathbf{H}$ a *hypertree hierarchy*.

Example 6: Let us verify that $\mathbf{H} = \{\mathcal{H}_0, \mathcal{H}_1,\ \mathcal{H}_2\}$, depicted in Figure 7, is a HMN, where

$$\begin{aligned} \mathcal{H}_0 &= \{h_1 = A_1A_3A_4,\ h_2 = A_2A_3A_4\}, \\ \mathcal{H}_1 &= \{h_{11} = A_1A_3,\ h_{12} = A_1A_4\}, \\ \mathcal{H}_2 &= \{h_{21} = A_2A_3,\ h_{22} = A_2A_4\}. \end{aligned}$$

MN $\mathcal{H}_0$ fulfills condition (i) as the root of $\mathbf{H}$. Condition (ii) is also satisfied. $\mathcal{H}_1$ is a child of $\mathcal{H}_0$ and $context(\mathcal{H}_1) = \{A_1A_3A_4\}$ is a subset of $h_1 \in \mathcal{H}_0$. $\mathcal{H}_2$ is another child of $\mathcal{H}_0$ and $context(\mathcal{H}_2) = \{A_2A_3A_4\}$ is contained in $h_2 \in \mathcal{H}_0$. Condition (iii) is satisfied since $\mathcal{H}_1$ can be assigned to hyperedge h_1 in $\mathcal{H}_0$, while $\mathcal{H}_2$ can be assigned to h_2 in $\mathcal{H}_0$.

Note that not every HMN can be faithfully represented by a single BN. The HMN $\mathbf{H} = \{\mathcal{H}_0,\ \mathcal{H}_1,\ \mathcal{H}_2\}$ in Example 6 cannot be faithfully represented by a BN. More formally,

$$\begin{aligned} CI(\mathbf{H}) &= CI(\mathcal{H}_0) \cup CI(\mathcal{H}_1) \cup CI(\mathcal{H}_2) \\ &= \{A_1 \rightarrow\rightarrow A_3 \mid A_4,\ A_2 \rightarrow\rightarrow A_3 \mid A_4, \\ &\quad\ A_3A_4 \rightarrow\rightarrow A_1 \mid A_2\} \end{aligned}$$

cannot be faithfully represented by a single DAG.

We now turn our attention to the logical implication of CI and EMVD in the hypertree hierarchy graphical representation.

Example 7: Suppose we wish to verify that $C \models c$, where C is defined as:

$$\{A_1 \rightarrow\rightarrow A_3 \mid A_4,\ A_2 \rightarrow\rightarrow A_3 \mid A_4,\ A_3A_4 \rightarrow\rightarrow A_1 \mid A_2\}$$

and $c = A_1A_2 \rightarrow\rightarrow A_3$. The initial tableau $T_{\mathcal{R}}$ is constructed according to c as shown in Figure 8 (left). We can apply the M-rule [11] corresponding to the EMVD $A_1 \rightarrow\rightarrow A_3 \mid A_4$ in C to joinable rows $w_1 = < a_1\ a_2\ a_3\ b_1 >$ and $w_2 = < a_1\ a_2\ b_2\ a_4 >$ to generate the new row $w_3 = < a_1\ b_3\ a_3\ a_4 >$ as shown in Figure 8 (right). Similarly, we can apply the M-rule corresponding to the EMVD $A_2 \rightarrow\rightarrow A_3 \mid A_4$ in C to joinable rows $w_1 = < a_1\ a_2\ a_3\ b_1 >$ and $w_2 = < a_1\ a_2\ b_2\ a_4 >$ to generate the new row $w_4 = < b_4\ a_2\ a_3\ a_4 >$ as shown in Figure 8 (right). Finally, we can obtain the row $< a_1\ a_2\ a_3\ a_4 >$ of all distinguished variables by applying the M-rule corresponding to the MVD $A_3A_4 \rightarrow\rightarrow A_1 \mid A_2$ in C to joinable rows w_3 and w_4. Therefore, $C \models c$. $\square$

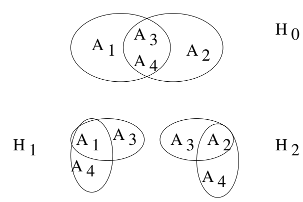

Fig. 7. A *hierarchical Markov network* $\mathbf{H}$.

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

$$\mathbf{r}(A_1A_2A_3A_4) = \begin{array}{|ccccc|} A_1 & A_2 & A_3 & A_4 & A_p \\ \hline 0 & 0 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 1 & 0.2 \\ 0 & 0 & 1 & 0 & 0.2 \\ 0 & 0 & 1 & 1 & 0.1 \\ 0 & 1 & 1 & 1 & 0.1 \\ 1 & 0 & 1 & 1 & 0.1 \\ 1 & 1 & 1 & 1 & 0.1 \\ \hline \end{array}$$

Fig. 9. Relation $\mathbf{r}$ satisfies all of the CIs in $\mathbf{C}$ but does not the CI $\mathbf{c}$, where $\mathbf{C}$ and $\mathbf{c}$ are defined in Example 8. Therefore, $\mathbf{C} \not\models \mathbf{c}$.

The chase algorithm used in Example 7 has shown that $C \models c$. Now consider the corresponding set of CIs.

Example 8: Consider the set $\mathbf{C} = \{A_1 \Rightarrow\Rightarrow A_3 \mid A_4,\ A_2 \Rightarrow\Rightarrow A_3 \mid A_4,\ A_3A_4 \Rightarrow\Rightarrow A_1 \mid A_2\}$ and $\mathbf{c}$ is the CI $A_1A_2 \Rightarrow\Rightarrow A_3$. It is easily verified that relation $\mathbf{r}(A_1A_2A_3A_4)$ in Figure 9 satisfies all of the CIs in $\mathbf{C}$ but does not satisfy the CI $\mathbf{c}$. Therefore, $\mathbf{C} \not\models \mathbf{c}$. □

The important point in this section is that Examples 7 and 8 together indicate that

$$\mathbf{C} \models \mathbf{c} \;\not\Leftarrow\; C \models c. \quad (13)$$

That is, even with the restriction that the input set of constraints represent a hypertree hierarchy, the implication of EMVD does not coincide with the implication of CI. We summarize our finding in Table II.

TABLE II

UNLIKE TABLE I, THE LOGICAL IMPLICATION OF CI AND EMVD DOES NOT ALWAYS COINCIDE WHEN A RESTRICTION IS IMPOSED.

Restriction	Logical implication of CI and EMVD coincides
fixed context	yes
DAG	yes
acyclic hypergraph	yes
none	no
hypertree hierarchy	no

V. Conclusion

We have previously argued [11], [13] that there is no *real* difference between the Bayesian database model and the relational database model in a *practical* sense. In fact, we made the conjecture that the Bayesian database model generalizes the relational database model on *all* solvable classes of dependencies. Since we have explicitly shown in this paper that the implication problem for CI and EMVD does not coincide for sets defining a HMN, future work then needs to determine whether this class of constraints is solvable or not.

References

[1] S. Abiteboul, R. Hull and V. Vianu. *Foundations of Databases*. Addison-Wesley, New York, 1995.

[2] C. Beeri, R. Fagin, D. Maier and M. Yannakakis, "On the desirability of acyclic database schemes," *Journal of the Association for Computing Machinery*, vol. 30, 479–513, 1983.

[3] R. Fagin, "Multivalued dependencies and a new normal form for relational databases," *ACM Transactions on Database Systems*, vol. 2, no. 3, 262–278, 1977.

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

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

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

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

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

[9] G. Shafer. *Probabilistic Expert Systems*, Society for Industrial and Applied Mathematics, 1996.

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

[11] S.K.M. Wong, C.J. Butz and D. Wu. On the implication problem for probabilistic conditional independency. *IEEE Trans. Sys. Man. Cyb.*, A, 30(6):785–805, 2000.

[12] S.K.M. Wong, C.J. Butz and D. Wu. On undirected representations of Bayesian networks. *ACM SIGIR Workshop on Mathematical/Formal Models in Information Retrieval*, 52-59, 2001.

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