A Relational Knowledge System

by

Shik Kam Michael Wong, Cory James Butz,
and Dan Wu

Technical Report CS-01-04
January 2001

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

ISSN 0828-3494
ISBN 0-7731-0417-8

Abstract

The objective of this paper is to demonstrate that the conventional relational database model is in fact a special case of a more general *relational knowledge system* (RKS). In managing uncertainty in AI, the Bayesian network model has become the standard approach. The RKS provides a natural unification of the Bayesian and relational database models. There are both theoretical and practical benefits that can be drawn from unifying these two apparently different systems. For example, similar to probabilistic networks one can use embedded multivalued dependencies to improve the database schema design. On the other hand, many important basic notions and theorems developed for relational databases can be generalized to Bayesian networks. Furthermore, probabilistic networks can take advantage of practical experience gained in designing conventional DBMS, particularly in the area of performance enhancement. In fact, an existing relational DBMS can be easily extended to become a probabilistic reasoning system.

1 Introduction

Our purpose is to convince the reader that the conventional relational database model is in fact a special case of a more general relational knowledge system. The Bayesian network model has become increasingly important for probabilistic reasoning [4]. Our general relational knowledge system (RKS) provides a natural unification of Bayesian networks and relational databases. There are both theoretical and practical benefits that can be drawn from unifying these two apparently different approaches. For example, we will explicitly demonstrate in this paper that, based on our experience with Bayesian networks, embedded multivalued dependencies can be used to improve database schema design. On the other hand, many important basic notions and theorems developed for relational databases can be easily generalized to Bayesian networks. Furthermore, probabilistic network design can take full advantage of the practical knowledge gained in designing a DBMS, particularly in the area of performance enhancement. In fact, an existing relational DBMS can be easily extended to become a probabilistic reasoning system.

Our relational knowledge system can be understood in terms of how a function $f(ABC\ldots)$ can be factorized (decomposed). For our exposition here, we only consider a *joint probability distribution* (jpd). For example, Figure 1 depicts two hierarchical factorizations of the jpd $p(ABCDEF)$ based on a set of *probabilistic conditional* *independencies* (CIs). The factorization of $p(ABCDEF)$ defined by case (i) is:

$$\begin{aligned} p(ABCDEF) &= \frac{p(AC) \cdot p(AB) \cdot p(BDE) \cdot p(DEF)}{p(A) \cdot p(B) \cdot p(DE)} \\ &= p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(DE|B) \cdot p(F|DE) \\ &= p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(D|B) \cdot p(E|BD) \cdot p(F|DE). \end{aligned}$$

The factorization of $p(ABCDEF)$ defined by case (ii) is:

$$\begin{aligned} p(ABCDEF) &= \frac{p(AB) \cdot p(AC) \cdot p(BD) \cdot p(CE) \cdot p(DEF)}{p(A) \cdot p(B) \cdot p(C) \cdot p(DE)} \\ &= p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(D|B) \cdot p(E|C) \cdot p(F|DE). \end{aligned}$$

Figure 2 depicts the schema of the corresponding factorization. In this discussion, we are mainly interested in hierarchical factorizations as they are equivalent to a set of *embedded* CIs. (It should be noted that in general there may not exist a *faithful* hierarchical factorization for an arbitrary set of embedded CIs.)

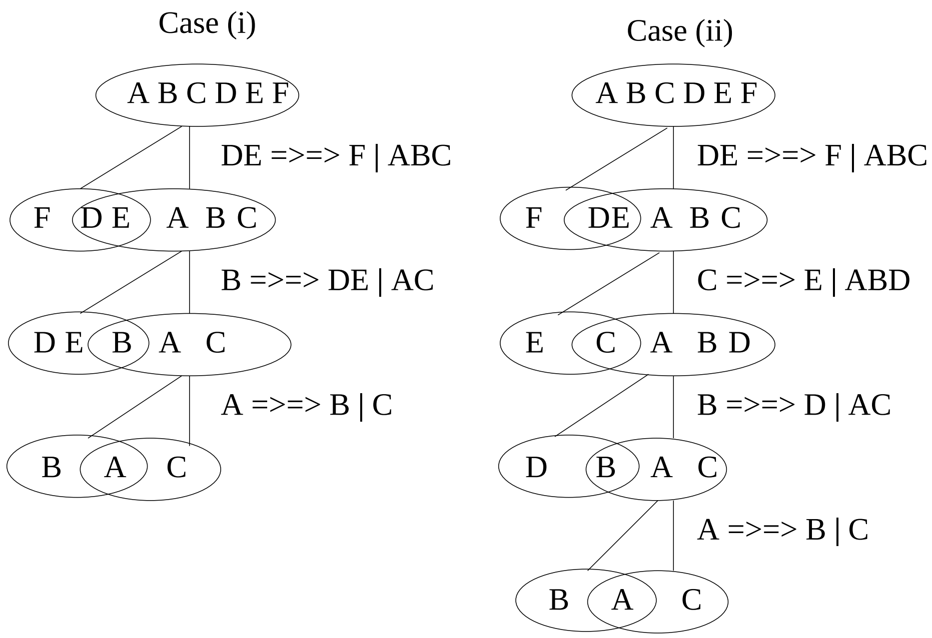

Figure 1: Two cases of hierarchical factorization.

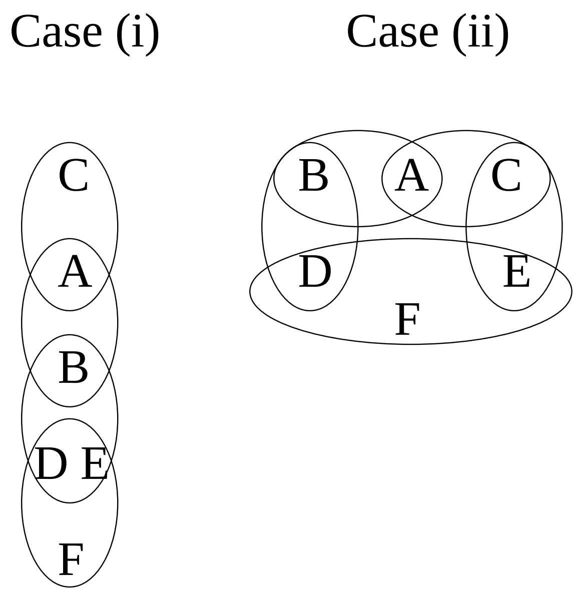

Figure 2: The resulting schemas of the two cases of hierarchical factorization of $p(ABCDEF)$.

2 Basic Notions

It is perhaps clearer to use some simple examples to illustrate the basic notions of our model.

Distributions and Relations

A joint probability distribution (jpd) can be considered as a generalized relation. For example, a jpd $p(ABCD)$ can be represented as a generalized relation $\mathbf{r}_p(ABCD)$, as shown in Figure 3. Notice that:

$$\sum_{ABCD} p(ABCD) \;=\; 1.0.$$

The standard relation $r(ABCD)$ corresponding to $\mathbf{r}_p(ABCD)$ is shown in Figure 4.

Marginalization and Projection

$\mathbf{r}_p(ABCD) =$

A	B	C	D	$p(ABCD)$
0	0	0	0	0.066
0	0	1	1	0.133
0	1	0	0	0.133
0	1	1	0	0.266
1	1	1	1	0.400

Figure 3: A jpd $p(ABCD)$ represented as a generalized relation $\mathbf{r}_p(ABCD)$.

$r(ABCD) =$

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

Figure 4: The standard relation corresponding to the generalized relation $\mathbf{r}_p(ABCD)$ in Figure 3.

Marginalization is a generalized projection. The marginalization of $p(ABCD)$ onto A is the marginal distribution $p(A)$ defined as:

$$p(A) \;=\; \sum_{BCD} p(ABCD).$$

The generalized relation $\mathbf{r}_p(A)$ representing $p(A)$ is shown in Figure 5. The corresponding projection $r(A)$ corresponding to $\mathbf{r}_p(A)$ is shown in Figure 6.

Conditional Independence and Multivalued Dependency

(Probabilistic) conditional independency (CI) [4] is a generalized multivalued dependency (MVD) [2]. The conditional independence of A and D given BC, and its

$\mathbf{r}_p(A) =$

A	$p(A)$
0	0.6
1	0.4

Figure 5: The generalized relation $\mathbf{r}_p(A)$ representing $p(A)$.

$$r(A) = \begin{array}{|c|} \hline A \\ \hline 0 \\ 1 \\ \hline \end{array}$$

Figure 6: The standard relation $r(A)$ corresponding to the generalized relation $\mathbf{r}_p(A)$ in Figure 5.

denotation are respectively given below:

$$p(ABCD) = \frac{p(ABC) \cdot p(BCD)}{p(BC)}; \qquad BC \Rightarrow\Rightarrow A|D.$$

A conditional independence does not necessarily have to involve all of the attributes in a distribution. For example,

$$p(ABC) = \frac{p(AB) \cdot p(AC)}{p(A)}; \qquad A \Rightarrow\Rightarrow B|C.$$

If the CI $BC \Rightarrow\Rightarrow A|D$ holds in $\mathbf{r}_p(ABCD)$, then the corresponding MVD $BC \rightarrow\rightarrow A|D$ holds in $r(ABCD)$. That is,

$$BC \Rightarrow\Rightarrow A|D \implies BC \rightarrow\rightarrow A|D.$$

However, it can be shown that multivalued dependency (MVD) is a necessary but not a sufficient condition for conditional independence (CI) [5]. In other words,

$$BC \Rightarrow\Rightarrow A|D \not\Longleftarrow BC \rightarrow\rightarrow A|D.$$

Another important point to notice is that the CI $A \Rightarrow\Rightarrow B|C$ holds in $\mathbf{r}_p(ABC)$ but not in $\mathbf{r}_p(ABCD)$. That is, $A \Rightarrow\Rightarrow B|C$ is valid with respect to the context ABC but not with respect to context $ABCD$. Thus, we call $A \Rightarrow\Rightarrow B|C$ an *embedded* CI. Similarly, the MVD $A \rightarrow\rightarrow B|C$ holds in $r(ABC)$ but not in $r(ABCD)$. We then call $A \rightarrow\rightarrow B|C$ an *embedded* MVD.

3 Hierarchical Factorization

In this section, we use a comprehensive example to illustrate the ideas so far. Consider the jpd $p(ABCD)$ shown at the top of Figure 7. The distribution can be hierarchically

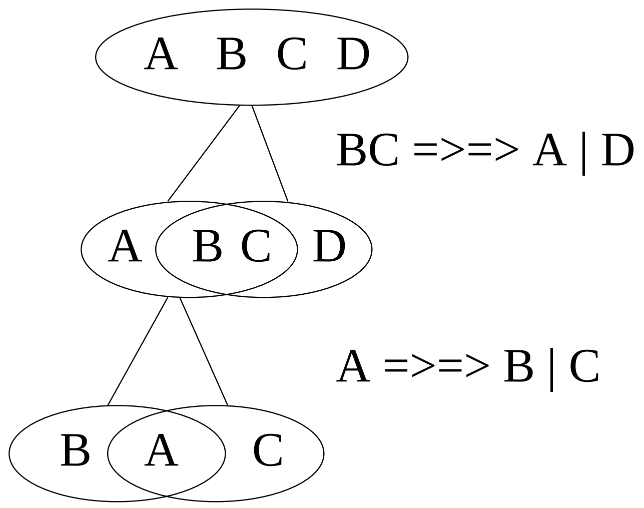

Figure 7: An example of hierarchical factorization.

factorized as:

$$\begin{aligned} p(ABCD) &= \frac{p(ABC) \cdot p(BCD)}{p(BC)} \\ &= \frac{\frac{p(AB) \cdot p(AC)}{p(A)} \cdot p(BCD)}{p(BC)} \\ &= p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(D|BC). \end{aligned}$$

That the embedded CI $A \Rightarrow\Rightarrow B|C$ holds in $\mathbf{r}(ABCD)$ is illustrated in Figure 8, where $\otimes$ is a generalized natural join operation [5]. Similarly, Figure 9 indicates that $\mathbf{r}(ABCD)$ satisfies the CI $BC \Rightarrow\Rightarrow A|D$. That is,

$$\mathbf{r}(ABCD) = \mathbf{r}(ABC) \otimes \mathbf{r}(BCD).$$

This means that the above factorization can be expressed as:

$$\mathbf{r}(ABCD) = (\mathbf{r}(AB) \otimes \mathbf{r}(AC)) \otimes \mathbf{r}(BCD).$$

$\mathbf{r}_p(ABCD) =$

A	B	C	D	$p(ABCD)$
0	0	0	0	0.066
0	0	1	1	0.133
0	1	0	0	0.133
0	1	1	0	0.266
1	1	1	0	0.400

$\mathbf{r}(A) =$

A	$p(A)$
0	0.6
1	0.4

, $\mathbf{r}(AB) =$

A	B	$p(AB)$
0	0	0.2
0	1	0.4
1	1	0.4

, $\mathbf{r}(AC) =$

A	C	$p(AC)$
0	0	0.2
0	1	0.4
1	1	0.4

$\mathbf{r}(ABC) =$

A	B	C	$p(ABC)$
0	0	0	0.066
0	0	1	0.133
0	1	0	0.133
0	1	1	0.266
1	1	1	0.400

$= \mathbf{r}(AB) \otimes \mathbf{r}(AC) =$

A	B	C	$p(AB) \cdot p(AC)/p(A)$
0	0	0	$0.2 \cdot 0.2/0.6$
0	0	0	$0.2 \cdot 0.4/0.6$
0	1	0	$0.4 \cdot 0.2/0.6$
0	1	1	$0.4 \cdot 0.4/0.6$
1	1	1	$0.4 \cdot 0.4/0.4$

Figure 8: The embedded CI $A \Rightarrow\Rightarrow B|C$ is satisfied by generalized relation $\mathbf{r}_p(ABCD)$.

$\mathbf{r}(BC) =$

B	C	$p(BC)$
0	0	0.066
0	1	0.133
1	0	0.133
1	1	0.666

, $\mathbf{r}(BCD) =$

B	C	D	$p(BCD)$
0	0	0	0.066
0	1	1	0.133
1	0	0	0.133
1	1	0	0.666

$\mathbf{r}(ABCD) =$

A	B	C	D	$p(ABC) \cdot p(BCD)/p(BC)$
0	0	0	0	$(0.066 \cdot 0.066)/0.066 = 0.066$
0	0	1	1	$(0.133 \cdot 0.133)/0.133 = 0.133$
0	1	0	0	$(0.133 \cdot 0.133)/0.133 = 0.133$
0	1	1	0	0.266
1	1	1	1	0.400

Figure 9: The CI $BC \Rightarrow\Rightarrow A|D$ is satisfied by generalized relation $\mathbf{r}_p(ABCD)$.

4 The Semi-graphoid Inference Axioms

To facilitate our discussion on the hierarchical factorizations of a jpd, it is convenient to first introduce *inference rules* for the probabilistic conditional independencies. (It is well known that there exists inference rules for multivalued dependencies. [3])

The *semi-graphoid* (SG) inference axioms are:

$$(SGa) \quad X \Rightarrow\Rightarrow Y \Longrightarrow XW \Rightarrow\Rightarrow YW, XW \Rightarrow\Rightarrow Y, \ W \subseteq R$$

$$(SGb) \quad X \Rightarrow\Rightarrow Y|ZW \Longrightarrow X \Rightarrow\Rightarrow Y|Z,$$

$$(SGc) \quad [X \Rightarrow\Rightarrow Z(\text{w.r.t. context}XYZ)\text{and}XY \Rightarrow\Rightarrow Z] \Longrightarrow X \Rightarrow\Rightarrow Z - Y.$$

(SGa),(SGb) and (SGc) are respectively called *augmention*, *projection*, and *contraction*. (Here we assume that all trivial CIs, namely, $X \Rightarrow\Rightarrow Y$ with $Y \subseteq X$, and $X \Rightarrow\Rightarrow R$ are included as input.)

For convenience, the above rules can be expressed in terms of disjoint sets as follows:

$$(SG1) \quad X \Rightarrow\Rightarrow Y|ZW \Longrightarrow X \Rightarrow\Rightarrow ZW|Y$$

$$(SG2) \quad X \Rightarrow\Rightarrow Y|ZW \Longrightarrow XW \Rightarrow\Rightarrow Z|Y$$

$$(SG3) \quad X \Rightarrow\Rightarrow Y|ZW \Longrightarrow X \Rightarrow\Rightarrow Z|Y$$

$$(SG4) \quad X \Rightarrow\Rightarrow Y|Z, XY \Rightarrow\Rightarrow W|Z \Longrightarrow X \Rightarrow\Rightarrow YW|Z.$$

(SG1),(SG2),(SG3) and (SG4) are respectively called *symmetry*, *weak union*, *projection*, and *contraction*.

As in relational databases, we can use these rules to infer new CIs from an input set of CIs. An important difference between the SG inference rules and the standard MVD rules is that *embedded* independencies are explicitly used in the SG rules. In contrast, only MVDs with a *fixed* context are involved in the standard relational database model. We argue that such an approach is unnecessarily restrictive.

5 Case (i) - Revisited

Let us now resume the analysis of the case (i) hierarchical factorization of a jpd (see Figure 1).

Recall that in this case *no* keys are being split in the factorization. It is therefore clear that the resulting *schema* is represented by an acyclic hypergraph $\mathcal{R}$ [1]. In this example, we obtain the schema:

$$\mathcal{R} = \{R_1 = AC, R_2 = AB, R_3 = BDE, R_4 = DEF\},$$

as shown in the Figure 10.

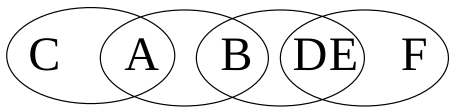

Figure 10: The resulting schema for case (i).

Obviously, $\mathcal{R}$ is acyclic. The corresponding factorization of the jpd is given by:

$$\begin{aligned}
p(ABCDEF) &= \frac{p(ABCDE) \cdot p(DEF)}{p(DE)} \\
&= [\frac{p(ABC) \cdot p(BDE)}{p(B)}] \cdot \frac{p(DEF)}{p(DE)} \\
&= [\frac{p(AB) \cdot p(AC)}{p(A)}] \cdot \frac{p(BDE) \cdot p(DEF)}{p(B) \cdot p(DE)} \\
&= \frac{p(AB) \cdot p(AC) \cdot p(BDE) \cdot p(DEF)}{p(A) \cdot p(B) \cdot p(DE)} \\
&= p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(DE|B) \cdot p(F|DE) \\
&= p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(D|B) \cdot p(E|BD) \cdot p(F|DE).
\end{aligned}$$

The set $\mathbf{M}$ of CIs used in this factorization is:

$$\mathbf{M} \;=\; \{DE \Rightarrow\Rightarrow F|ABC,\ B \Rightarrow\Rightarrow DE|AC,\ A \Rightarrow\Rightarrow B|C\}.$$

We call $\mathcal{J}_{\mathbf{M}} = \{DE, B, A\}$ the *keys* of $\mathbf{M}$.

Applying the SG inference rules recursively, it can be shown that $\mathbf{M}$ is logically equivalent to a set $\mathbf{M}'$ of (full) CIs with respect to the largest context $ABCDEF$. For example, by applying the SG inference rules, we obtain:

$$B \Rightarrow\Rightarrow AC|DE \quad \Longrightarrow \quad \left.\begin{array}{l} AB \Rightarrow\Rightarrow DE|C \\ A \Rightarrow\Rightarrow B|C \end{array}\right\} \quad \Longrightarrow \quad A \Rightarrow\Rightarrow BDE|C.$$

$$DE \Rightarrow\Rightarrow ABC|F \quad \Longrightarrow \quad \left.\begin{array}{l} ABDE \Rightarrow\Rightarrow F|C \\ A \Rightarrow\Rightarrow BDE|C \end{array}\right\} \quad \Longrightarrow \quad A \Rightarrow\Rightarrow BDEF|C.$$

Thus, the CI $A \Rightarrow\Rightarrow B|C$ can be *expanded* to a larger context, i.e., $ABCDEF$. Similarly, we can expand the CI $B \Rightarrow\Rightarrow AC|DE$ to the context $ABCDEF$ as follows:

$$DE \Rightarrow\Rightarrow F|ABC \quad \Longrightarrow \quad \left.\begin{array}{l} BDE \Rightarrow\Rightarrow F|AC \\ B \Rightarrow\Rightarrow DE|AC \end{array}\right\} \quad \Longrightarrow \quad B \Rightarrow\Rightarrow DEF|AC.$$

Let

$$\mathbf{M}_c = \{A \Rightarrow\Rightarrow BDEF|C,\ B \Rightarrow\Rightarrow DEF|AC,\ DE \Rightarrow\Rightarrow F|AC\}.$$

Obviously, $\mathbf{M}$ is equivalent to $\mathbf{M}_c$, i.e., $\mathbf{M} \equiv \mathbf{M}_c$. In fact, $\mathbf{M}_c$ provides a *canonical* representation of such a hierarchical factorization in which no key is allowed to be split in any step of the factorization. (We call $\mathbf{M}_c$ the canonical basis of the hierarchical factorization.)

Let us summarize our observations for this first kind of hierarchical factorization:

(i) The resulting *schema* $\mathcal{R}$ is an acyclic hypergraph.

(ii) $\mathbf{M}_c \equiv \mathbf{M}$

(iii) $\mathbf{M}_c$ and $\mathbf{M}$ have the same set of keys, i.e., $\mathcal{J}_{\mathbf{M}_c} = \mathcal{J}_{\mathbf{M}}$.

(iv) $\mathcal{J}_{\mathbf{M}_c}$ is the set of J-keys of the acyclic hypergraph $\mathcal{R}$. That is, $\mathcal{R}$ is a *perfect-map* of $\mathbf{M}_c$.

Property (iv) implies that the right sides of those CIs in $\mathbf{M}_c$ are *dependency bases* [1] of its left sides (i.e., the keys of $\mathbf{M}_c$). Furthermore, $\mathbf{M}_c$ is a set of *conflict-free* CIs satisfying the following properties [1]:

1. No key in $\mathcal{K}_{\mathbf{M}_c} = \{X, Y, \ldots\}$ is split by any CI in $\mathbf{M}_c$.

2. For any keys X and Y in $\mathcal{K}_{\mathbf{M}_c}$, $DEP(X) \cap DEP(Y) \subseteq DEP(X \cap Y)$, where $DEP(X)$, $DEP(Y)$ and $DEP(X \cap Y)$ represent the dependency bases of X, Y, and $X \cap Y$, respectively.

Thus, one can already see at this point that we are talking about something very similar to *acyclic join dependency* (AJD) [1] in the relational database model. We will come back to this and establish formally that AJD is indeed a special case of this type of factorization of a joint probability distribution.

We should mention a few more important facts before closing the discussion on case (i). First, the canonical basis $\mathbf{M}_c$ uniquely determines the acyclic schema $\mathcal{R}$. However, there are many different factorizations associated with a particular canonical basis. Consider for example the following set $\mathbf{M}'$ of CIs:

$$\mathbf{M}' \;=\; \{A \Rightarrow\Rightarrow BDEF|C,\ B \Rightarrow\Rightarrow DEF|A,\ DE \Rightarrow\Rightarrow F|B\}.$$

We can easily transform $\mathbf{M}'$ into $\mathbf{M}_c$ and show $\mathbf{M}' \equiv \mathbf{M}$. The factorization induced by $\mathbf{M}'$ is shown in Figure 11. Figure 11 clearly indicates that the resulting schema produced by $\mathbf{M}'$ is the same as the one shown in Figure 1, case (i). In fact, each different factorization associated with a given $\mathbf{M}_c$ specifies a particular *lossless join plan* in the database language. Each of these join plans is associated with a *hypertree construction ordering* [1] of the acyclic hypergraph $\mathcal{R}$.

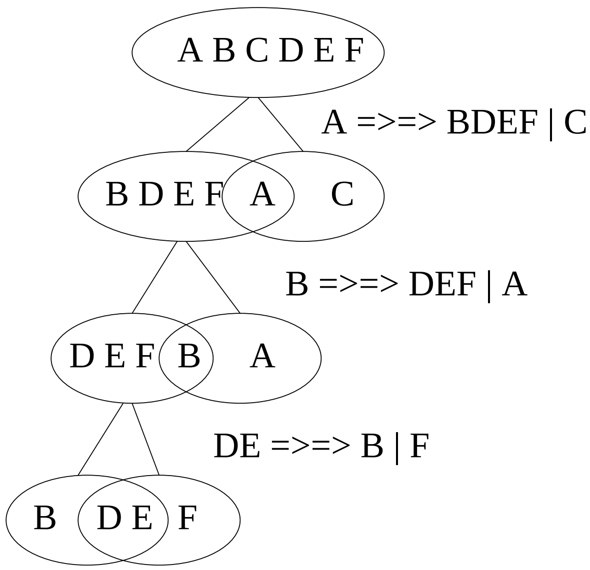

Figure 11: An example of hierarchical factorization.

It is important to note that the canonical basis $\mathbf{M}_c$ contains only *full* CIs (only those CIs defined on the largest context R). We know that $\mathbf{M}_c$ completely determines all of the possible factorizations of the jpd under consideration. Therefore, we do not really take advantage of "real" embedded CIs if they exist. For this reason, the type of factorizations allowed in this case is rather restrictive. We are going to consider a more general type of factorization in case (ii), where we can really use embedded CIs to factorize the jpd.

We call this type of factorization of a jpd a *Markov network* as opposed to a Bayesian network which is described in case (ii). As mentioned before, we will show that AJD in the conventional relational database model is a special case of a Markov network.

6 Case (ii) - Revisited

The only difference between case (i) and case (ii) is that in the factorization process a key in $\mathcal{K}_{\mathbf{M}}$ is allowed to be split at most once in case (ii), whereas no key is allowed to be split at all in case (i). However, this difference has a very significant impact on the type of CIs we can use in the factorization. In contrast to case (i), we can now really take advantage of embedded CIs to design our knowledge system - a probabilistic reasoning system.

Since we allow keys to be split in the factorization, the hypergraph $\mathcal{R}$ (in Figure 12),

$$\mathcal{R} \;=\; \{R_1 = AB, R_2 = AC, R_3 = BD, R_4 = CE, R_5 = DEF\},$$

of our example for case (ii) is not acyclic.

The set $\mathbf{M}$ of CIs used in this factorization is:

$$\mathbf{M} \;=\; \{DE \Rightarrow\Rightarrow F|ABC,\; C \Rightarrow\Rightarrow E|ABD,\; B \Rightarrow\Rightarrow D|AC,\; A \Rightarrow\Rightarrow B|C\}.$$

Let us summarize our observations at this point for this type of hierarchical factorization:

(i) The resulting schema $\mathcal{R}$ is a cyclic hypergraph. (The cyclicity is *primarily* due to the fact that a key is allowed to be split.)

(ii) In contrast to case (i), we can no longer transform $\mathbf{M}$ using the SG inference rules to obtain a set $\mathbf{M}'$ of CIs such that $\mathbf{M}' \equiv \mathbf{M}$ and $\mathbf{M}'$ contains only (full) CIs with respect to a fixed context.

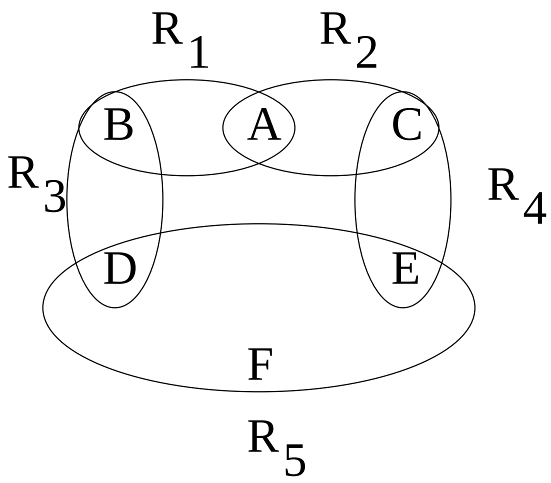

Figure 12: The resulting schema $\mathcal{R}$ for case (ii) is not acyclic.

(iii) We know that a *single* hypergraph (whether cyclic or acyclic) can represent only CIs with respect to a fixed context. Therefore, we can immediately conclude that no single hypergraph can be a perfect-map of **M**.

So, apparently we would have lost all the advantages of a Markov network which we have analyzed in case (i). What is the remedy then? Actually, there are two elegant "faithful" representations of such a factorization of a given jpd. These representations are a natural extension of the Markov networks. Although they are equivalent representations, each has its own strength. In fact, they complement each other very well. We will show later that this type of factorization (referred to as a Bayesian network in probabilistic reasoning) suggests a better method for schema design for relational databases. This new approach leads to a new kind of join dependency (referred to as *hierarchical join dependency* (HJD)), which includes AJD as a special case and has many additional advantages over the standard AJD.

We will first describe the representation which is characterized by a single *directed acyclic graph* (DAG). The second representation uses a hierarchy of hypergraphs to represent the CIs that are used in the factorization.

6.1 Representation 1

Let us use the example in Figure 1 for case (ii) to illustrate the salient features of this method - the Bayesian network. Based on the set $\mathbf{M}$ of CIs,

$$\mathbf{M} = \{DE \Rightarrow\Rightarrow F|ABC,\ C \Rightarrow\Rightarrow E|ABD,\ B \Rightarrow\Rightarrow D|AC,\ A \Rightarrow\Rightarrow B|C\},$$

the factorization of the jpd can be written as (see Figure 1):

$$\begin{aligned} p(ABCDEF) &= \frac{p(ABCDE) \cdot p(DEF)}{p(DE)} \\ &= [\frac{p(ABCD) \cdot p(CE)}{p(C)}] \cdot \frac{p(DEF)}{p(DE)} \\ &= [\frac{p(ABC) \cdot p(BD)}{p(B)}] \cdot \frac{p(CE) \cdot p(DEF)}{p(C) \cdot p(DE)} \\ &= [\frac{p(AB) \cdot p(AC)}{p(A)}] \cdot \frac{p(BD) \cdot p(CE) \cdot p(DEF)}{p(B) \cdot p(C) \cdot p(DE)} \\ &= \frac{[\frac{p(AB) \cdot p(AC) \cdot p(BD) \cdot p(CE)}{p(A) \cdot p(B) \cdot p(C)}] \cdot p(DEF)}{p(DE)}. \end{aligned} \quad (1)$$

Note that the above factorization can be expressed as:

$$p(ABCDEF) = p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(B|D) \cdot p(E|C) \cdot p(F|DE). \quad (2)$$

It should be emphasized here that the factorization expressed by Equation (1) in terms of marginals is intrinsic but that expressed by Equation (2) in terms of conditional probabilities is not. For example, based on the intrinsic factorization in Equation (1), the jpd can be expressed as:

$$p(ABCDEF) = p(B) \cdot (p(D|B) \cdot p(A|B) \cdot p(C|A) \cdot p(E|C) \cdot p(F|DE). \quad (3)$$

There is an elegant graphical representation of this type of hierarchical factorization in terms of a *directed acyclic graph* (DAG). We outline this method below using the same example we used above.

For any conditional probability, say, $p(F|DE)$, we call those attributes (variables) on the right-hand side of the vertical bar in $p(\cdot|\cdot)$, i.e., DE, the parent(s) of the attribute on the left-hand side of the bar, i.e., F. We call F a child of DE. Every attribute in the jpd is represented by a node. For each conditional probability that

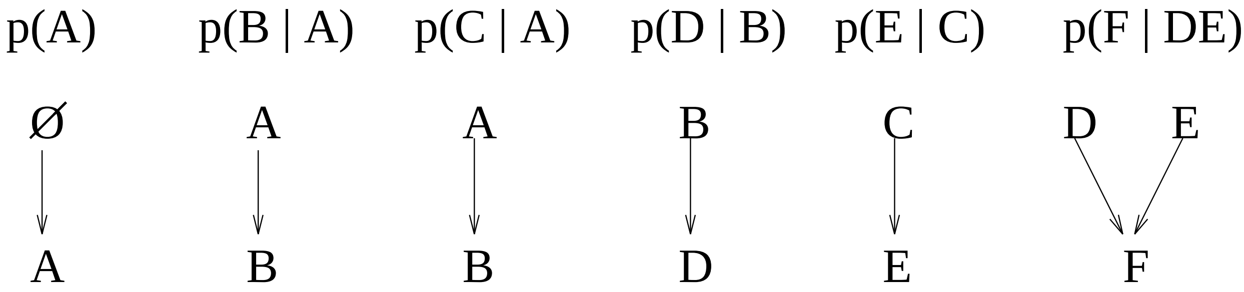

Figure 13: Sub-DAGs depicting each conditional independence.

appears in Equation (2), we construct a sub-DAG as follows: draw a directed edge from each parent node to the child node. We obtain the sub-DAGs shown in Figure 13.

The DAG representing Equation (2) is constructed by combining all of the above sub-DAGs, as illustrated in Figure 14. (Note that the resulting directed graph is acyclic because Equation (2) is associated with a hierarchical factorization.)

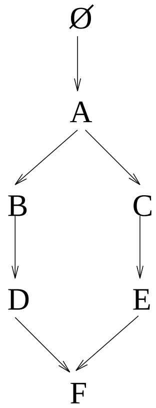

Figure 14: Depicting all of the conditional independencies.

Similarly, we can construct the DAG for Equation (3), as shown in Figure 15. Although the DAG in Figure 14 looks different from the DAG in Figure 15, they actually represent the same information. (Recall that both Equations (2) and (3) are derived from the *same* Equation (1).

It is perhaps worth mentioning here that Equation (3) and the corresponding DAG in Figure 15 are directly associated with the hierarchical factorization depicted

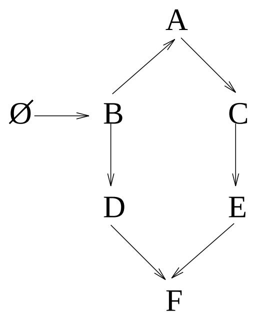

Figure 15: An equivalent depiction of the above conditional independencies.

in Figure 16.

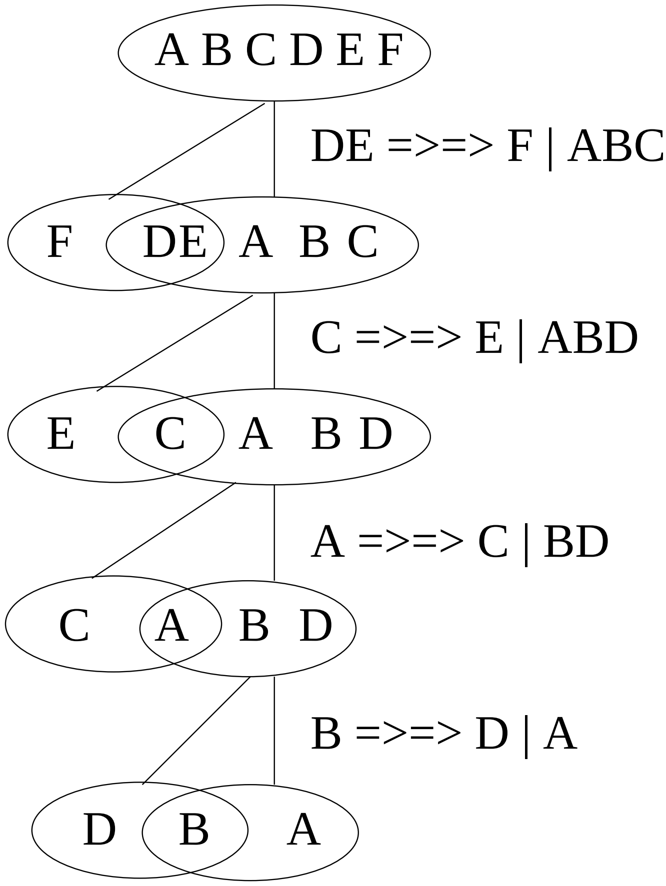

Figure 16: The hierarchical factorization according to the DAG in Figure 15.

Note that the set

$$\mathbf{M}' = \{DE \Rightarrow\Rightarrow F|ABC,\ C \Rightarrow\Rightarrow E|ABD,\ A \Rightarrow\Rightarrow C|BD,\ B \Rightarrow\Rightarrow D|A\}$$

used for this factorization is different from **M** used in the previous factorization. However, both of these hierarchical factorizations produce the same schema (shown in Figure 17).

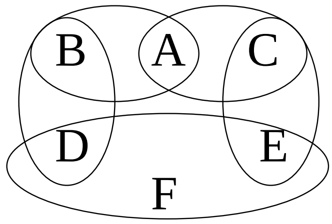

Figure 17: The resulting schema for case (ii).

One of the most important results in probabilistic reasoning (based on the Bayesian network model) is that the DAG constructed from a factorization in terms of the conditional probabilities (resulting from a hierarchical factorization of case (ii)) is a *perfect graphical representation* (a perfect-map) of such a factorization.

Like the "finger rule" for inferring CIs from a hypergraph, here we introduce a rule, called the *d-separation* method [4], for inferring CIs from the DAG constructed from a particular factorization of a jpd.

6.1.1 The d-separation rule

If X, Y and Z are three disjoint subsets of attributes (variables or nodes) in a DAG $\mathcal{D}$, then X is said to d-separate Y from Z, denoted $< Y|X|Z >_{\mathcal{D}}$, if along every path between a node in Y and a node in Z, there is a node ω in the path satisfying *one* of the following two conditions:

(i) ω has converging arrows, and none of its descendants (including ω) is in X,

(ii) ω does not have converging arrows and ω is in X.

If a path satisfies one of the above conditions, it is said to be *blocked*; otherwise, it is said to be *active*.

For example, consider the DAG in Figure 18. In this DAG, we say that $Y = B$ and $Z = C$ are d-separated by $X = A$. Why? Note that there are two possible paths between Y and Z, namely:

(a) $B \leftarrow A \rightarrow C$,

(b) $B \rightarrow D \leftarrow C$.

The path (a) is blocked by X because A does not have converging arrows and is in X. The second path (b) is also blocked because D and its descendant are not in X. That is, A d-separates B and C, namely, $< B|A|C >_{\mathcal{D}}$.

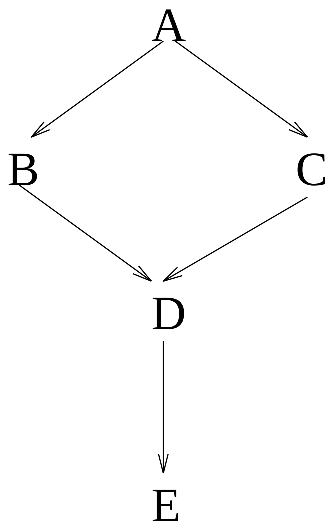

Figure 18: An example DAG.

However, $Y = B$ and $Z = C$ are *not* d-separated by $X' = AE$, because in path (b), $B \rightarrow D \leftarrow C$, D has converging arrows and its descendent E is in $X = AE$. Thus, AE does not d-separate B and C.

The d-separation method means that:

$$< Y|X|Z >_{\mathcal{D}} \implies X \Rightarrow\Rightarrow Y|Z.$$

This method provides a graphical rule for inferring CIs from a DAG representing a particular factorization.

6.1.2 Logical Implication

We say that a CI $X \Rightarrow\Rightarrow Y|Z$ is *logically implied* by a particular *form* $\mathcal{F}_p$ of factorization of a jpd, written

$$\mathcal{F}_p \quad \models \quad X \Rightarrow\Rightarrow Y|Z,$$

if whenever any jpd that can be factorized in the *same* form $\mathcal{F}_p$, it satisfies $X \Rightarrow\Rightarrow Y|Z$.

Based on these definitions, we can now state precisely that the DAG associated with a particular factorization of a jpd is a perfect graphical representation of the logical implication of CIs.

It can be show that $X \Rightarrow\Rightarrow Y|Z$ is logically implied by a particular factorization $\mathcal{F}_p$, i.e., $\mathcal{F}_p \models X \Rightarrow\Rightarrow Y|Z$, if and only if $< Y|X|Z >_{\mathcal{D}}$, where $\mathcal{D}$ is the corresponding DAG constructed from the factorization $\mathcal{F}_p$. We call $\mathcal{D}$ a *perfect-map* of the factorization $\mathcal{F}_p$.

Recall that every factorization is associated with a set $\mathbf{M}$ of CIs. In our example, the set,

$$\mathbf{M} \quad = \quad \{DE \Rightarrow\Rightarrow F|ABC,\ C \Rightarrow\Rightarrow E|ABD,\ B \Rightarrow\Rightarrow D|AC,\ A \Rightarrow\Rightarrow B|C\}$$

used in the factorization as shown in Figure 1, case (ii), is shown in Figure 19.

Based on this factorization, the jpd can be expressed as:

$$p(ABCDEF) \quad = \quad p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(D|B) \cdot p(E|C) \cdot p(F|DE).$$

The DAG $\mathcal{D}$ in Figure 14 is constructed from this expression $\mathcal{F}_p$.

It can be shown that the above set $\mathbf{M}$ of CIs is in fact a cover of those CIs that can be inferred from the DAG $\mathcal{D}$ using the d-separation rule. That is,

$$\mathbf{M}^+ \quad = \quad \{X \Rightarrow\Rightarrow Y|Z \quad | \quad < Y|X|Z >_{\mathcal{D}}\},$$

where the closure $\mathbf{M}^+$ is defined with respect to the SG inference rules. In other words, $\mathbf{M} \vdash X \Rightarrow\Rightarrow Y|Z$ if and only if $< Y|X|Z >_{\mathcal{D}}$, where $\mathbf{M} \vdash X \Rightarrow\Rightarrow Y|Z$ denotes that the CI $X \Rightarrow\Rightarrow Y|Z$ can be derived from $\mathbf{M}$ using the SG inference rules.

This observation means that we can compute algebraically those CIs that are logically implied by a particular factorization $\mathcal{F}_p$ using the SG inference rules. The

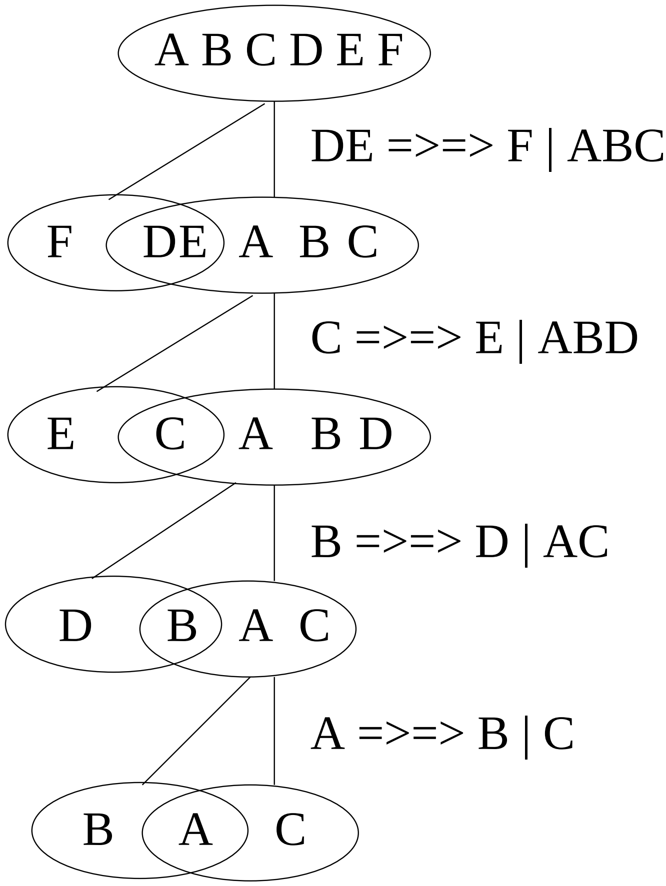

Figure 19: The hierarchical factorization according to case (ii).

d-separation rule, on the other hand, provides an equivalent graphical method for the implication problem. These two methods complement each other very well.

In concluding the discussion on this type of factorization, we want to mention one key advantage of the graphical representation based on a DAG. This graphical method provides a very practical semantic model for designing a relational knowledge system. In particular, we automatically obtain the *schema* once the DAG is constructed. The construction of the DAG, of course, depends on our understanding of the dependency relationships among the various attributes. Another advantage of this representation is that the numerical input to the system is entirely determined by supplying the conditional probabilities independently. Since we are dealing with a product of conditional probabilities, consistency is automatically guaranteed. Besides, conditional probabilities are easier to supply than marginals.

6.2 Representation 2

In analyzing case (i), we noticed that there exists a unique canonical basis $\mathbf{M}_c$ logically equivalent to the CIs used in the hierarchical factorization. More importantly, perhaps, $\mathbf{M}_c$, a set of CIs with a fixed context, is conflict-free and it can be characterized by a *single* acyclic hypergraph. Representation 2, which we are about to describe, is an extension of these ideas introduced in case (i). In the more general representation, the canonical basis $\mathbf{M}_c$ may contain CIs with mixed contexts. Therefore, we may have to represent $\mathbf{M}_c$ by a number of hypergraphs - in fact, a *hierarchy* of acyclic hypergraphs.

Again, let us use an example (the same example we used before) to illustrate this alternative representation in terms of hypergraphs rather than using a DAG.

Recall the hierarchical decompositions redrawn in Figure 20.

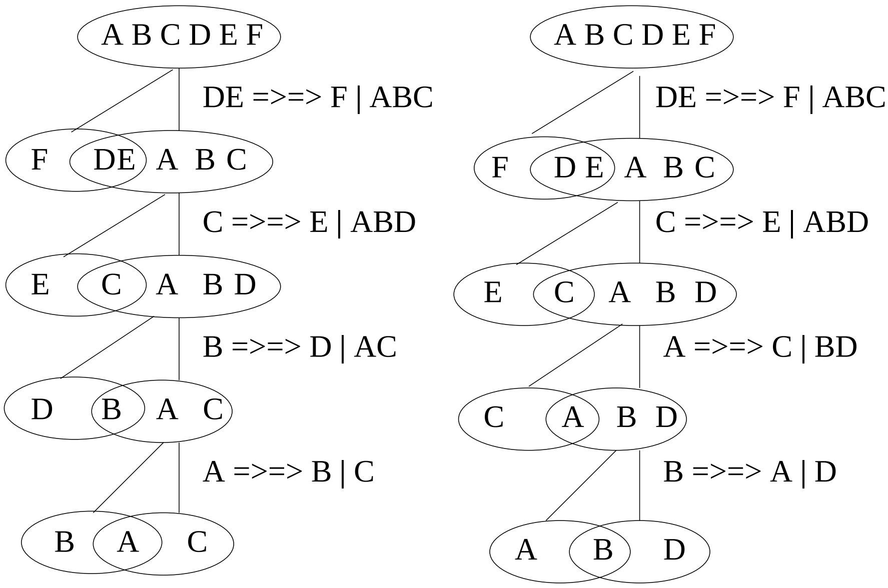

Figure 20: Two different, but equivalent, factorizations.

$$M = \{DE \Rightarrow\Rightarrow F|ABC,\ C \Rightarrow\Rightarrow E|ABD,\ B \Rightarrow\Rightarrow D|AC,\ A \Rightarrow\Rightarrow B|C\}$$

gives

$$p(ABCDEF) \quad = \quad p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(D|B) \cdot p(E|C) \cdot p(F|DE).$$

On the other hand,

$$M' \quad = \quad \{DE \Rightarrow\Rightarrow F|ABC,\ C \Rightarrow\Rightarrow E|ABD,\ A \Rightarrow\Rightarrow C|BD,\ B \Rightarrow\Rightarrow D|A\}$$

gives

$$p(ABCDEF) \quad = \quad p(B) \cdot p(D|B) \cdot p(A|B) \cdot p(C|A) \cdot p(E|C) \cdot p(F|DE).$$

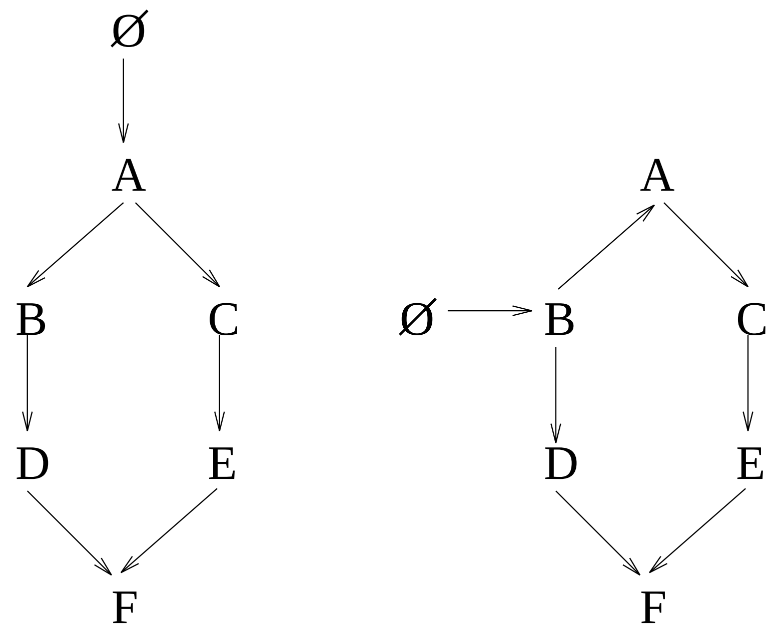

Figure 21: The respective DAGs according to the above factorizations.

Clearly, these factorizations *look* different. First of all, **M** and **M′** are different. The two expressions of the jpd $P(ABCDEF)$ in terms of conditional probabilities are different. Furthermore, the corresponding DAGs are different. However, the schema produced by these apparently different factorizations are the *same*.

In fact, the above factorizations are equivalent, i.e., one can be derived from the other and vice versa. This claim can be established from the following two observations:

(i) Note that:

$$\begin{aligned}\mathbf{M} &\equiv \mathbf{M}' \\ &\equiv \mathbf{M}_c \\ &= \{DE \Rightarrow\Rightarrow F|ABC,\ A \Rightarrow\Rightarrow BD|CE,\ B \Rightarrow\Rightarrow D|ACE,\ C \Rightarrow\Rightarrow E|ABD\},\end{aligned}$$

where $\mathbf{M}_c$ is called the canonical basis of this alternative representation in terms of hypergraphs (i.e., Representation 2). In other words, $\mathbf{M}$, $\mathbf{M}'$, and $\mathbf{M}_c$ have the same closure, namely:

$$\mathbf{M}^+ = (\mathbf{M}')^+ = (\mathbf{M}_c)^+,$$

where the closures are computed from the SG inference rules.

It can be shown that in general the CIs in $\mathbf{M}_c$ can be grouped into a hierarchy of sets of CIs. Here the hierarchy is defined in terms of the partial order of set containment on the context of the CIs.

In this example, for instance, $\mathbf{M}_c$ can be organized into two groups, namely,

$$\begin{aligned}\mathbf{M}_c &= \{\mathbf{M}_1,\ \mathbf{M}_2\} \\ &= \{\{DE \Rightarrow\Rightarrow F|ABC\},\ \{A \Rightarrow\Rightarrow BD|CE,\ B \Rightarrow\Rightarrow D|ACE,\ C \Rightarrow\Rightarrow E|ABD\}\}.\end{aligned}$$

Here $context_1 = ABCDEF$ and $context_2 = ABCDE$, and

$$context_1 \ \subseteq \ context_2.$$

It is important to note that each group of CIs in the hierarchy is *conflict-free*. This means that $\mathbf{M}_c$ can be represented by two acyclic hypergraphs $\mathcal{H}_1$ and $\mathcal{H}_2$ as follows:

$$\mathcal{H}_1 = \{DEF,\ ABCDE\} \ \equiv \ \mathbf{M}_1 = \{DE \Rightarrow\Rightarrow F|ABC\},$$

which induces the following factorization,

$$p(ABCDEF) \ = \ \frac{p(ABCDE) \cdot p(DEF)}{p(DE)},$$

and

$$\mathcal{H}_2 = \{AB, AC, BD, CE\} \ \equiv \ \mathbf{M}_2 = \{A \Rightarrow\Rightarrow BD|CE, B \Rightarrow\Rightarrow D|ACE, C \Rightarrow\Rightarrow E|ABD\},$$

which induces the following factorization,

$$p(ABCDE) \;=\; \frac{p(AB)\cdot p(AC)\cdot p(BD)\cdot p(CE)}{p(A)\cdot p(B)\cdot p(C)}.$$

(Note that the attributes of $\mathcal{H}_2$ are contained by one of the hyperedges of $\mathcal{H}_1$.)

These two factorizations together induce the combined factorization:

$$\begin{aligned} p(ABCDEF) &= \frac{p(ABCDE)\cdot p(DEF)}{p(DE)} \\ &= \frac{[\frac{p(AB)\cdot p(AC)\cdot p(BD)\cdot p(CE)}{p(A)\cdot p(B)\cdot p(C)}]\cdot p(DEF)}{p(DE)}, \end{aligned} \tag{4}$$

which we referred to previously as the *intrinsic* factorization of the jpd $p(ABCDEF)$. Thus, we have demonstrated that $\mathbf{M}_c$ is equivalent to a hierarchy of Markov networks. We call this representation the *canonical representation*.

(ii) The intrinsic factorization is expressed in terms of marginals. It is clear that we can derive from it many different factorizations expressed in terms of conditional probabilities. For example,

$$\begin{aligned} p(ABCDEF) &= \frac{[\frac{p(AB)\cdot p(AC)\cdot p(BD)\cdot p(CE)}{p(A)\cdot p(B)\cdot p(C)}]\cdot p(DEF)}{p(DE)} \\ &= p(A)\cdot p(B|A)\cdot p(C|A)\cdot p(B|D)\cdot p(E|C)\cdot p(F|DE) \\ &= p(B)\cdot p(D|B)\cdot p(A|B)\cdot p(C|A)\cdot p(E|C)\cdot p(F|DE) \\ &= p(C)\cdot p(A|C)\cdot p(E|C)\cdot p(B|A)\cdot p(D|B)\cdot p(F|DE) \\ &= p(D)\cdot p(B|D)\cdot p(A|B)\cdot p(C|A)\cdot p(E|C)\cdot p(F|DE). \end{aligned}$$

Thus, the factorizations given in Figures 10 and 11 are in fact equivalent because they can be derived from the intrinsic factorization in Equation (4). This means that factorizations in terms of conditional probabilities are not unique, but our canonical representation in terms of a hierarchy of Markov networks is unique.

We have already mentioned that the directed acyclic graph representation model provides a very useful semantic tool for designing a probabilistic reasoning system. It also greatly facilitates the acquisition of input knowledge. On the other hand, our canonical representation model is more suitable for probabilistic inference (answering queries in database language). We use a hierarchy of acyclic hypergraphs

Probabilistic Knowledge System	**Relatioinal Databaes System**
1. A joint probabilistic distribution $p(ABCDEF)$.	A relation $r(ABCDEF)$
2. The marginal $p(CD)$ of $p(ABCDEF)$	The projection $r(CD)$ of $r(ABCDEF)$
3. Conditional Independency (CI)	Multivalued Dependency (MVD)
4. Hierarchical Factorization of $p(ABCDEF)$	Hierarchical Decomposition of $r(ABCDEF)$
5. Markov Network (MN)	Acyclic join Dependency (AJD)
6. Hierarchical Markov Network (HMN)	Hierarchical Join Dependency (HJD)

Table 1: A summary of corresponding notions in probabilistic knowledge system and relational database system.

in the canonical model. This observation clearly indicates that such a representation is a direct extension of the Markov network (a generalization of the acyclic join dependency). (Recall that a Markov network is represented by a *single* acyclic hypergraph.) Nevertheless, these two representations complement each other very well; they together provide a powerful tool for modeling probabilistic knowledge as well as conventional relational database systems. We will argue in the next section that the probabilistic knowledge system we have described and the conventional database model have the same relational structure.

7 A Summarization of the Corresponding Notions

From the above discussion, it is intuitively clear that the probabilistic knowledge system can be viewed as a *numeric* relational database system. In fact, in the probabilistic model, there exists a numerical counter part of any notion in the conventional relational database model. We list in Table 1 some of the basic numeric and the corresponding non-numeric notions in these two models. (Note that the list in the table is far from complete.)

From Table 1, it is clear that the probabilistic knowledge system indeed *subsumes* the conventional relational database model. In particular, the inference rules in these two systems have the same *form* [6] . Thus, the implication problems of the probabilistic knowledge system and the relational database system *coincide*. This means that:

$$\mathbf{M} \models X \Rightarrow\Rightarrow Y \quad \text{iff} \quad M \models X \rightarrow\rightarrow Y,$$

where $\mathbf{M}$ is a set of CIs and M is the corresponding set of MVDs. That is, there is a one-to-one correspondence between the CIs in $\mathbf{M}$ and the MVDs in M. This observation clearly indicates that these two systems have the same inherent relational structure.

8 Conclusion

The recognition that there is a unified model for both the probabilistic knowledge system and the relational database system has many important consequences. One thing for sure is that one can apply the results of one system to the other and vice versa. Let us just describe one significant example of this nature before ending our discussion.

Consider again point (6) in the Table 1. We have demonstrated in our unified approach that a Bayesian network can be viewed as a hierarchy of Markov networks. What gives the additional advantage to Bayesian networks over Markov networks lies in the fact that embedded CIs are used in the schema design. (Recall that a Markov network only uses a restrictive type of CIs, i.e., CIs of a fixed context, in its representation.) Translated into the corresponding notions in relational databases, it is almost obvious that we should adopt the *Hierarchical Join Dependency* (HJD) (see (6) in the list) for database schema design. We have shown that the standard *Acyclic join Dependency* (AJD) is a special case of HJD. As in Bayesian networks, a HJD has many additional advantages over an AJD. More importantly, perhaps, is that based on our experience with Bayesian networks, adopting a HJD as a database schema should not have a significant negative impact on system performance.

References

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

[2] R. Fagin. Multivalued dependencies and a new normal form for relational databases. *ACM Transactions on Database Systems*, 2(3):262–278, September 1977.

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

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

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

[6] S.K.M. Wong, C.J. Butz, and D. Wu. On the implication problem for probabilistic conditional independency. *IEEE Transactions on System, Man, Cybernetics, Part A: Systems and Humans*, 30(6):785–805, 2000.