# THE THEORY OF PROBABILISTIC DATABASES

*Roger Cavallo*
*Michael Pittarelli*

Department of Computer and Information Science
State University of New York College of Technology
Utica, New York

## ABSTRACT

A theory of probabilistic databases is outlined. This theory is one component of an integrated approach to data-modelling that accomodates both probabilistic and relational data. In fact, many of the results presented here were developed in the context of a framework for structural modelling of systems. Much that is fundamental to relational database theory was also developed in this context, and previous to the introduction by Codd of the relational model of data.

Probabilistic databases can store types of information that cannot be represented using the relational model. Probabilistic databases may also be viewed as generalizations of relational databases; any relational database can be represented without loss of information by a probabilistic database. A number of relational database concepts are shown to have probabilistic counterparts. In many cases, it is preferable to deal with the probabilistic formulation of a concept even when applying it to a relational database. For example, we define a new project-join mapping for relational databases that is based on transforming a relational to a probabilistic database. This mapping is shown to have more fixed points than the standard one.

## INTRODUCTION

The initial presentation of ideas which led to the development of the relational database model is generally accepted to have been made by Codd [1970]. In the sense that some consideration is given to questions of model-utilization in the design of actual databases, this attribution seems to be justified. It is interesting that a major advantage of the relational approach stems from its generality and data-modelling power; in fact, it is only since the presentation of the relational model that a general agreement has evolved on distinguishing between *data-modelling* and actual *database management systems,* whereby data models are seen as providing the "conceptual basis for thinking about data-intensive applications" [Brodie, 1984]. But, in the sense that this conceptual basis is important, relation theory [Wiener, 1914; Bourbaki, 1954] and the relational approach to data-modelling, including a large number of specific concerns that have proven directly relevant to the theory of relational databases [Ashby, 1956, 1965], significantly predate Codd.

Since the main orientation of this paper is not historical, we only mention the priority of Ashby's system-theoretic consideration of concepts such as the lossless join and project-join mapping [Ashby, 1965; Maier, 1983, pp. 146-148], functional dependencies [Ashby, 1956; Madden and Ashby, 1972], and the study of injective properties of decomposition maps [Madden and Ashby, 1972; Maier, 1983] and also note the existence of other developments in the system-theoretic context that have reappeared in the context of relational database theory (e.g., a certain class of system referred to and studied as "$\gamma$-structures" [Cavallo and Klir, 1979a] has recently been introduced by Fagin [1983] as a database scheme which is "$\gamma$-acyclic").

The foregoing is not intended to imply that there is no difference between data-modelling concepts and their development in the context of database theory, but rather to motivate the development in what follows of a more general approach to data-modelling that also has roots in the work of Ashby. An integrated development of some of the fundamental ideas of this approach — which, in particular, incorporates consideration of the importance of information theoretic ideas — was given by Cavallo and Klir [1979a, 1981]. Use of these general modelling concepts to develop the data-modelling aspects of database theory will serve to extend the applicability of database theory beyond the relatively simplistic data-processing type of applications that are predominant. It should especially allow database theoretic contribution to the study of classical problems of science and engineering that involve experimentation and other forms of data collection (e.g., process monitoring, decision analysis, remote sensing, etc.).

Databases store information. The form of stored

information has traditionally been considered to be simple facts such as "Supplier X supplies part Y". From the point of view of data-modelling, many situations require more complex forms of information that can be used to answer such queries as

"How reliable is part Y when supplied by supplier X"

"Is the probability that a person of type X will purchase product Y greater if that person has also purchased product Z"

"If X is known, how much additional information about Y is provided by knowledge of Z".

Some aspects of the type of information referred to here have been dealt with in the study of so-called statistical databases [Denning, 1982, ch. 6] where the objective is to allow queries of a statistical nature to be made regarding a relational database.

Our objective is to provide a framework that generalises the relational database model and extends all the concepts that have been developed to deal with collections of yes/no facts to apply also to facts about which one is uncertain (probabilistic databases) or about which one has vague or "fuzzy" information (fuzzy databases). The three main aspects on which this generalisation are based are:

1) by considering a relational database to be primarily a set of mappings from logically possible tuples to the set {true,false} (i.e., to be a set of predicates) the extension to probabilistic and fuzzy databases is immediate as a generalisation of the mappings;

2) the importance in the relational database model of the concept of information and information preservation; while the concept of combinatorial information [Kolmogorov, 1965] is relevant to relational databases, there are well-developed theories of information that can be applied to probability or fuzzy measures and thus to probabilistic or fuzzy databases (these are a little less well-developed in the case of fuzzy measures - we primarily discuss probabilistic databases in this paper);

3) all the work done on relational database schemes as opposed to relational databases is immediately and directly applicable to schemes for probabilistic and fuzzy databases, requiring only that the appropriate information-theoretic concepts be correctly adapted.

As we have stated, a number of the main ideas associated with a unified view of modelling relational, probabilistic, and fuzzy systems have already been worked out and we incorporate this work into our development here of probabilistic database theory.

## 1. RELATIONAL AND PROBABILISTIC DATABASES

Usually a *relational database* (RDB) is defined as a finite collection of relations where each relation is a subset of the cartesian product of sets referred to as *domains*. Each domain is considered to be associated with an *attribute symbol* which has some significance in the context of the particular database application. For any relation, the set of attributes associated with the domains of that relation is called a *relation scheme* and the set of relation schemes is called a (relational) *database scheme*.

Formally, we define a *relational database* to be a set $RD=\{B_1,...,B_n\}$ where each element of RD is a *relational system*, $B=(V_i,\Delta_i,dom_i,r_i)$ where

- $V_i$ is a non-empty set of distinct symbols called *attributes;*
- $\Delta_i$ is a non-empty set of sets of values called *domains;*
- $dom_i: V_i \rightarrow \Delta_i$ is a function that associates a domain with each attribute; (The set of all possible tuples of $B_i$, $\underset{v \in V_i}{\times} dom_i(v)$, is referred to as $T_i$; the tuples in $T_i$ are often considered to be functions from the set of attributes to the union of the domains to avoid the need to order the components of a tuple. For our purposes we assume an ordering of the domains and assume that where relevant all uses of a tuple conform to this ordering.)
- $r_i: T_i \rightarrow \{0,1\}$ is a characteristic function that identifies a subset of $T_i$. This subset is a relation and we will often refer to this relation rather than to the full relational system.

Thus, each relational system $B_i$ has an associated *relation scheme* $V_i$ and the set $\{V_1,...,V_n\}$ is the *database scheme* on which RD is defined.

A *probabilistic system,* like a relational system, is a four-tuple $P=(V,\Delta,dom,p)$ but with its fourth component p a function of type $T \rightarrow [0,1]$ with the restriction that $\sum_{t \in T} p(t)=1$. We refer to p as a *distribution* (over V) and use the term interchangeably with probabilistic system. A *probabilistic database* (PDB) is a set PD of probabilistic systems. Probabilistic databases provide a means of representing types of information that cannot be captured by a relational database, and in such a way that all of the data-modelling concepts and mechanisms of relational database theory are applicable to these more complex modelling situations.

It is also the case that probabilistic databases generalise relational databases in the sense that any RDB can be represented by a PDB in such a way that important properties are preserved. A result of this is that useful concepts derived in the context of probability distribu-

tions may be applied in relational database theory. For example, by using probabilistic information theory with relational databases that have been transformed to probabilistic databases, cumbersome proofs of a number of significant results in relational database theory are simplified (see Section 3; Malvestuto, 1983).

## 2. INFORMATION AND CONSTRAINT

**2.1 INFORMATION CONTENT.** An important idea associated with relations when used in databases is that of information. Any relation, insofar as it is not the full cartesian product of its domains, exhibits constraint [Ashby, 1965]; it is in terms of this constraint that the information content of a relation is defined. Similarly, a distribution function exhibits constraint to the extent that it diverges from the uniform distribution over the set of tuples T.

Let H be the (Shannon) entropy of a discrete probability distribution,

$$H(q) = -\sum_{t\in T} q(t) \log q(t)$$

(by convention, 0 log 0 is 0; obviously, $H \geq 0$). Given a set of tuples T, H reaches its maximum value at u, where u is the uniform distribution over T, i.e., $u(t)=1/|T|$ for all $t\in T$.

**Definition:** Given a set of tuples T associated with a probabilistic system, the *information content* of a distribution p over T is given by $H(u)-H(p)$.

If p is a distribution associated with probabilistic system $P=(V,\Delta,dom,p)$ we often write H(V) instead of H(p). Similarly, (see section 3.2), given distributions, $p_i$ and $p_j$ over schemes $V_i$ and $V_j$, we write $H(V_i \mid V_j)$ for the conditional entropy of $V_i$ given $V_j$, defined as the average of the entropy of $V_i$ for each tuple of $\underset{v\in V_j}{\times} dom_j(v)$, weighted by $p_j(t)$ [see Khinchin, 1957, p.35].

Operations that are commonly performed on databases (e.g, project-join) may result in the replacement of a distribution p by a distribution q. To develop the idea of approximate satisfaction of join dependencies (database decompositions) we use a measure of the information lost by such a replacement. On the other hand we also want a measure of how accurately a PDB determines a distribution over some set of attributes (e. g., $\cup V_i$) when such a distribution is not represented in a single probabilistic system in the database. Before describing this measure we describe the two operations on databases and distributions that we use in this paper: projection and (probabilistic) join.

**2.2 PROJECTION.** Let P be a system with distribution p and scheme V and let $Z\subseteq V$. The *projection* of p onto Z results in the distribution

$$\downarrow_Z(p): \underset{v\in V}{\times} dom(v) \rightarrow [0,1]$$

where

$$\downarrow_Z(p)(b)=\sum_{a>b} p(a)$$

and

$$a\in \underset{v\in V}{\times} dom(v) > b\in \underset{v\in Z\subseteq V}{\times} dom(v) \text{ if } (a_v) = (b_v), \quad v\in Z$$

[Cavallo and Klir, 1979a]. We also refer to the result of the projection operation, $\downarrow_Z(p)$, as a projection. The definition is justified by observing that any b can be viewed as an event equivalent to the union of mutually exclusive and exhaustive subevents a. (When dealing with a relational system and characteristic function r, the definition is the same as for probabilistic systems except that the operator $\sum$ is replaced by max. $\downarrow_Z(r)$ corresponds to the notation in the relational database literature $\pi_Z(r)$, where, in the latter expression, r is the set of tuples represented by the characteristic function.) The system $(Z,\Delta,dom|Z,\downarrow_Z(p))$ will be referred to as a *subsystem* of P, and Z a *subscheme* of V (dom|Z is the restriction of dom to Z). The projection of a distribution p onto a database scheme $X=\{V_1,...,V_k\}$ is the set of subdistributions (the database instance) $\{\downarrow_{V_1}(p),...,\downarrow_{V_k}(p)\}$.

*Example:* The projection of the distribution

| $v_1$ | $v_2$ | $v_3$ | p(.) |
|---|---|---|---|
| 0 | 0 | 0 | 0.0 |
| 0 | 0 | 1 | 0.3 |
| 0 | 1 | 0 | 0.15 |
| 0 | 1 | 1 | 0.15 |
| 1 | 0 | 0 | 0.2 |
| 1 | 0 | 1 | 0.1 |
| 1 | 1 | 0 | 0.05 |
| 1 | 1 | 1 | 0.05 |

onto the database scheme $\{\{v_1,v_2\},\{v_2,v_3\}\}$ is the database instance

| $v_1$ | $v_2$ | $p_1(.)$ | $v_2$ | $v_3$ | $p_2(.)$ |
|---|---|---|---|---|---|
| 0 | 0 | 0.3 | 0 | 0 | 0.2 |
| 0 | 1 | 0.3 | 0 | 1 | 0.4 |
| 1 | 0 | 0.3 | 1 | 0 | 0.2 |
| 1 | 1 | 0.1 | 1 | 1 | 0.2 |

$p_1(v_1v_2=00)$, for example, is obtained as $p_1(v_1v_2=00)=p(v_1v_2v_3=000)+p(v_1v_2v_3=001)$, etc.

**2.3 RECONSTRUCTION FAMILY AND JOIN.** Let $p_1,..,p_n$ be distributions with schemes $V_1,..,V_n$, and let $X=\{V_1,..,V_n\}$.

**Definition:** The *reconstruction family* of database scheme X relative to distributions $p_1,..,p_n$, denoted $R_X$, is the set of distributions over $\cup V_i$ whose projections onto the schemes $V_1,..,V_n$ equal $p_1,..,p_n$ [Cavallo and Klir, 1981].

Any reconstruction family is the set of solutions p of a set of linear equations (which imply the equation $\sum p(.)=1$), subject to $p \geq 0$, and is therefore a bounded polyhedral set. For the database instance above, its reconstruction family, $R_{\{\{v_1,v_2\},\{v_2,v_3\}\}}$, is the set of all distributions satisfying the set of equations (subject to $p \geq 0$):

$$p(v_1v_2v_3=000)+p(v_1v_2v_3=001)=0.3$$

$$p(v_1v_2v_3=010)+p(v_1v_2v_3=011)=0.3$$

$$\cdots$$

$$p(v_1v_2v_3=011)+p(v_1v_2v_3=111)=0.2$$

**Definition:** The operation *probabilistic join* applied to $\{p_i\}$ results in the maximum entropy distribution from among the members of the reconstruction family.

We denote the result of the join operation by $*\{p_i\}$, and refer to it also as a join. Thus,

$$H(*\{p_i\})=\max\{H(p) | p \text{ is over } \cup V_i \text{ and } \downarrow_{V_i}(p)=p_i\}.$$

(Note that for relational systems the definition of (natural) join is the same as that for probabilistic systems except that maximum entropy is replaced by maximum cardinality; any other member of the reconstruction family is a subset of the join [Ashby, 1965].) When $*\{p_i\}$ exists, it is unique, and its existence implies that $\downarrow_{V_j \cap V_i}(p_j)=\downarrow_{V_j \cap V_i}(p_k)$ for all j, k. Procedures for calculating $*\{p_i\}$ have been developed and studied in a number of contexts [Brown, 1959; Lewis, 1959; Bishop et al, 1975]. In the Appendix we use the computational definition to prove certain equivalences between relational and probabilistic data dependencies.

*Example:* The maximum entropy element of the reconstruction family in our running example is the distribution

| $v_1$ | $v_2$ | $v_3$ | p(.) |
|---|---|---|---|
| 0 | 0 | 0 | 0.1 |
| 0 | 0 | 1 | 0.2 |
| 0 | 1 | 0 | 0.15 |
| 0 | 1 | 1 | 0.15 |
| 1 | 0 | 0 | 0.1 |
| 1 | 0 | 1 | 0.2 |
| 1 | 1 | 0 | 0.05 |
| 1 | 1 | 1 | 0.05 |

and is easily calculated using techniques described in [Cavallo and Klir, 1981].

There are a number of information-theoretic arguments that can be made for choosing, as the result of the join, the maximum entropy distribution from among the set of distributions that project onto the $V_i$ [see Jaynes, 1979; Cavallo and Klir, 1981]. In addition to these, if the conversion to probabilities, as described below, is made, the natural join of a set of relations, when transformed to a probability distribution, gives the maximum entropy among all the (transformed) relations that project onto the given set of relations.

**2.4 DATABASE TRANSFORMATION.** A relational system is converted to a probabilistic system, and conversely, by means of two mappings, trans and $\text{trans}^{-1}$, defined as

$$\text{trans}:[T \rightarrow \{0,1\}] \rightarrow [T \rightarrow [0,1]]$$

where

$$\text{trans}(r)(t) = r(t) / \sum_{a \in T} r(a)$$

(it is assumed that for some $a \in T$, $r(a) > 0$; i.e., the relation is not empty) and

$$\text{trans}^{-1}:[T \rightarrow [0,1]] \rightarrow [T \rightarrow \{0,1\}]$$

where

$$\text{trans}^{-1}(p)(t)= \lceil p(t) \rceil .$$

It is easily proved that $\text{trans}^{-1}(\text{trans}(r))=r$, i.e., that $\text{trans}^{-1}$ is a left inverse of trans. Demonstrations that the probabilistic characterization of a relational system preserves important properties are found in section 3 and in the appendix. Here we use trans to define a new relational project-join mapping that has more fixed points than the standard one.

**2.5 PROJECT-JOIN.** It is convenient to separately define the project-join mapping for both probabilistic and relational systems.

**Definition:** Let X be a database scheme. The *project-join mapping* defined by X, applied to p, is $PPJ(X,p) = \underset{V \in X}{*}\{\downarrow_V(p)\}$.

We abbreviate the result of project-join as $p^*$ when X is clear from the context. Likewise, for a relational system, $RPJ(X,r)=\underset{V \in X}{\bowtie}\{\downarrow_V(p)\}$, abbreviated $r^{\bowtie}$.

We may define another project-join mapping for a relational system as $\text{trans}^{-1}(PPJ(X,\text{trans}(r)))$. It is sometimes the case that, although $r \neq r^{\bowtie}$, i.e., $r \neq RPJ(X,r)$, r does equal $\text{trans}^{-1}(PPJ(X,\text{trans}(r)))$. Thus, the transformation described above allows lossless decompositions of relational databases over a larger set of database schemes.

*Example:* For the system represented by the table

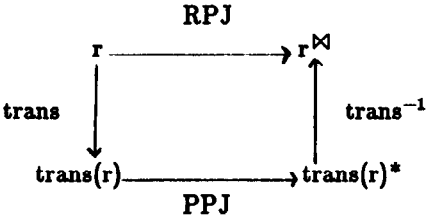
| $v_1$ | $v_2$ | $v_3$ | p(.) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

if $X = \{\{v_1,v_2\},\{v_2,v_3\},\{v_1,v_3\}\}$, $RPJ(X,r) \neq r$, but $r = trans^{-1}(PPJ(X,trans(r)))$. In fact, since X is the least refined element in the lattice of database schemes Y over $V=\{v_1,v_2,v_3\}$ such that $Y \neq \{V\}$ (see section 3.1), there is no nontrivial lossless decomposition of r, if we are restricted to the standard relational project-join mapping (this follows immediately from results in [Ashby, 1965]; see also [Cavallo, 1980]). However, r is a fixed point of the project-join mapping defined as the composition of the mappings trans, PPJ, and $trans^{-1}$.

This situation cannot arise when X is a $\gamma$-structure (is not $\gamma$-acyclic) [Cavallo and Klir, 1979a; Fagin, 1983]. When X is a $\gamma$-structure,

$$RPJ(X,r) = trans^{-1}(PPJ(X,trans(r))),$$

that is, the diagram below commutes:




The proof follows easily from the computational definition of $p^*$ for $\gamma$-structures.

**2.6 INFORMATION LOSS.** When replacing one distribution with another, as for example by using PPJ, the resulting information loss can be measured by the *directed divergence* from p to q [Kullback, 1959; Aczel and Daroczy, 1975],

$$d(p,q)=\sum_{t\in T} p(t)\log(p(t)/q(t)).$$

In the case that q is the uniform distribution u, then d(p,u) is easily shown to be equal to H(u)−H(p), (i.e., the directed divergence from p to u in fact measures the information content of p). We also have the result that, denoting PPJ(X,p) by $p^*$, $d(p,p^*)=H(p^*)-H(p)$ [Higashi, 1984]. The difference in the amount of information contained in p and that contained in $p^*$ is $(H(u)-H(p))-(H(u)-H(p^*)) = H(p^*)-H(p)$. Thus we have:

**Fact 1:** Given a probabilistic system P with distribution p and a database scheme X, the difference in information content when P is replaced by the database with scheme X and distributions the appropriate projections of p, is equal to the directed divergence from p to PPJ(X,p), i.e., H(PPJ(X,p))−H(p).

In addition to this, for a large class of database schemes, if $p^*$ results from applying a project-join mapping to p, the quantity $H(p^*)$ can be calculated and the strength of the join dependency (see section 3.1) evaluated, without determining $p^*$ (that is, without performing the relatively expensive probabilistic join for a PDB or natural join for an RDB, if the transformation described above is carried out). For the most simple of such database schemes, for example, (denote X∪Y by XY) when $X\cap Y=X\cap Z=Y\cap Z=\emptyset$, if q is the join of XY and XZ then H(q)=H(XY)+H(XZ)−H(X).

## 3. PROBABILISTIC DATA DEPENDENCIES

In standard simple relational database applications, data dependencies can often be inferred from the meaning of the attributes, as determined by the application. In complex situations associated with scientific databases, it is often the case that dependencies are not known beforehand, and an important analogue of database design is the determination of which dependencies exist or of the relative strength of various dependencies when they do not exist in an absolute sense. Three of the most commonly dealt with are: join, functional, and multivalued dependencies. Here we define corresponding data dependencies for probabilistic databases. They generalize the relational concepts in two senses:

1) they apply to relational databases, as well as to probabilistic databases (if the simple transformation described in section 2 is carried out)
2) it is straightforward, by application of results from information theory, to speak of approximate satisfaction of probabilistic dependencies.

Both relational and probabilistic dependencies, in their exact or approximate forms, may be viewed as constraints that restrict the set of allowable functions r or p of systems in a database. Since information can be identified with constraint, the concepts of (probabilistic) information theory are fundamental to a theory of (probabilistic) data dependencies. As will be seen, they also provide a natural way to characterize and reason about relational dependencies.

**3.1 JOIN DEPENDENCIES.** A join dependency holds when a relational or probabilistic system can be decomposed into a collection of (sub)systems such that the system is equal to the join of its subsystems.

**Definition:** A probabilistic system $P=(V,\Delta,dom,p)$ satisfies the join dependency $*[V_1,...,V_n]$ iff $p=*\{\downarrow_{V_i}(p)\}$. If $X=\{V_1,..., V_n\}$, then $*[V_1,...,V_n]$ may be abbreviated *[X].

For any particular database instance over a scheme X, the join dependency *[X] is satisfied only by the maximum entropy element of the reconstruction family $R_X$. For any other element, p, of $R_X$, some error will result when it is replaced by the maximum entropy element, PPJ(X,p). The maximum error (information loss) that could result by applying the project-join mapping to any $p\in R_X$ we call the *information radius* of $R_X$,

$$ir(R_X) = \max_{p \in R_X} d(p,p^*).$$

This maximum is achieved for a vertex of $R_X$, and therefore is easily calculated (see appendix). For the example of section 2, $ir(R_X)=0.400$.

For any database application, accuracy of the facts that are represented is important. The problem of measurement error is obviously more severe for probabilistic data than it is for relational data. The statement "tuple t is sometimes observed" is more trustworthy, by virtue of its being less informative, than the statement "the probability of observing tuple t is x". If, as is the case when values p(t) are determined by sampling, p is an approximation to the probabilities of occurrence of tuples, a dependency may hold among the attributes of the system (if one accepts that the notion of "true" probabilities makes any sense), and yet the definition as given above will not be satisfied. Regardless, it is often meaningful and useful to speak of a dependency holding approximately. (A notion of approximate satisfaction of dependency constraints could also be developed for (strictly) relational databases, although this seems not to have been done; for example, a join is considered to be either "lossless" or "lossy". What is being lost via a lossy join is constraint, i.e., the information that certain tuples cannot be observed.) With probabilistic databases, the amount of information lost can be quantified in terms of entropy.

Information-theoretic definitions of join dependencies and approximate join dependencies follow.

**Definition:** p satisfies $*[V_1,...,V_n]$ iff $d(p,p^*)=0$ (recall that we use $p^*$ for $*\{\downarrow_{V_i}(p)\}$ when the context makes clear what the distributions are, and d is the measure of information loss introduced in section 2).

**Definition:** Database scheme $X=\{V_i,...,V_j\}$ is a *refinement* of scheme $Y=\{V_k,...,V_l\}$, denoted $X \leq Y$, iff for each $V_x \in X$ there exists a $V_y \in Y$ such that $V_x \subseteq V_y$ [Cavallo and Klir, 1979a].

*Example:* $\{\{v_1,v_2\},\{v_2,v_3,v_4\}\}$ is a refinement of $\{\{v_1,v_2,v_3\},\{v_1,v_2,v_4\},\{v_2,v_3,v_4\}\}$.

In the following we restrict our attention to database schemes $X=\{V_1,...,V_m\}$ which satisfy the following two properties: (i) $\cup V_i=V$; (ii) $i \neq j$ implies $V_i$ is not a subset of $V_j$. The second property ensures that no information is included that can be obtained by a single application of the project operation. Such schemes are known as reduced hypergraphs and have been studied by Cavallo and Klir [1979a] and Fagin [1983]. The refinement relation on database schemes defines a lattice with the universal scheme {V} as universal upper bound and the most refined scheme $\{\{v_1\},...,\{v_n\}\}$ as universal lower bound.

**Fact 2:** $X \leq Y \Rightarrow d(p, \overset{*}{_{V \in X}}\{\downarrow_V(p)\}) \geq d(p, \overset{*}{_{V \in Y}}\{\downarrow_V(p)\})$.

In words, if X is a refinement of Y (i.e., X is more "decomposed" than Y), the information loss when a distribution p is projected onto scheme X and the resulting systems are joined is greater than or equal to that produced by the project-join onto scheme Y (this obviously implies that, if p satisfies the join dependency *[X], then p satisfies *[Y]). The proof of Fact 2 follows immediately from:

1) $\downarrow_V(p)$ represents a set of linear equalities and inequalities;
2) $R_X$ represents the set of all solutions to the linear system determined by the projection onto X;
3) if $X \leq Y$, then each equation determined by the projection of p onto X is a linear combination of equations in the system determined by the projection of p onto Y; thus, all solutions to the latter system are also solutions to the first, i.e., $R_Y \subseteq R_X$, and the maximum entropy member of $R_X$ must have entropy at least as large as the maximum entropy member of $R_Y$.

Analogously, we note for relational systems,

**Fact 3:** $X \leq Y \Rightarrow RPJ(Y,r) \subseteq RPJ(X,r)$.

Let M denote the most refined scheme over V. It follows from Fact 2 that $d(p, \overset{*}{_{V \in X}}\{\downarrow_V(p)\})$ reaches its maximum when X=M, and from Fact 3 that $RPJ(M,r) \supseteq RPJ(X,r)$, where X satisfies $\cup V_i=V$.

**Definition:**

$$JD(P,X) = \begin{cases} 1, \text{ if } d(p,u)=0 \\ \dfrac{d(p,u) - d(p,p^*)}{d(p,u)}, \text{ otherwise.} \end{cases}$$

$JD(P,X)$ is a normalized measure of approximate join dependency satisfaction for a probabilistic system P. The value of $JD(P,X)$ is the proportion of the information content of P preserved by the project-join mapping of p onto X. $JD(P,X)=1$ iff p satisfies *[X]. We say that P satisfies the join dependency *[X] to degree $\delta$ if $JD(P,X) \geq \delta$. The value of $JD(P,X)$ indicates the degree to which P may be viewed as decomposable into scheme X. Clearly, $X \leq Y$ implies $JD(P,X) \leq JD(P,Y)$.

**Fact 4:** $X \leq Y \Rightarrow ir(R_Y) \leq ir(R_X)$. (Proof in appendix.)

Given $ir(R_X)$ and $H(p^*)$, $\min_{p \in R_X} JD(p,X)$ is easily calculated. Of course, $\max_{p \in R_X} JD(p,X) = 1$, for any $R_X$.

The ability to detect the most refined scheme for which a join dependency or approximate join dependency of acceptable strength exists is of obvious value to the scientific user of a database interested in performing an "exploratory" analysis to detect relationships among

attributes that might suggest scientific hypotheses [Good, 1983]. This capability would also be useful in database design when faced with a large and/or unfamiliar set of attributes. A join dependency detected empirically, for a particular database instance, might hold for all instances; and the database designer, once the dependency is discovered, might be able to deduce that it holds [see Fagin et al, 1982]. On the other hand, the ability to mechanically detect the presence of dependencies can also be used to detect their absence, providing a check against erroneous assumptions that they exist.

Algorithms for detecting such dependencies in both relational and probabilistic data have been developed in context of a framework for structural modelling of systems [Cavallo and Klir, 1979a, 1981].

**3.2 FUNCTIONAL DEPENDENCIES.** Functional dependencies are also extremely easily dealt with in the probabilistic context [see also Nambiar, 1980; Malvestuto, 1983]. For relational databases , a set of attributes Y is *functionally dependent* on a set of attributes X, denoted $X \rightarrow Y$, iff it is the case that if tuples agree on attributes X, then they also agree on attributes Y.

**Definition:** For a probabilistic system $P=(V,\Delta,dom,p)$, with $X,Y \subseteq V$, and with the distributions over X and Y the appropriate projections of p,

$$X \rightarrow_p Y \text{ iff } H(Y|X)=0$$

where H(Y|X) is the conditional entropy of Y, given X (see section 2.1), which may also be defined as $H(Y|X)=H(Y \cup X)-H(X)$.

(In the following, let YX denote $Y \cup X$.) Intuitively, $H(Y|X)=0$ means that once the tuple values for attributes X are known, there is no uncertainty regarding possible tuple values for attributes Y: if a set of tuples $A \subseteq \underset{v \in X \cup Y}{\times} dom(v)$ agree on attributes X, then for at most one $t \in A$ is $\downarrow_{XY}(p)(t) > 0$.

*Example:* $\{1,2,4\} \rightarrow_p \{1,3\}$ in the system represented by the table

| $v_1$ | $v_2$ | $v_3$ | $v_4$ | p(.) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0.0 |
| . | . | . | . | . |
| 0 | 1 | 1 | 1 | 0.0 |
| 1 | 0 | 0 | 0 | 0.0 |
| 1 | 0 | 0 | 1 | 0.25 |
| 1 | 0 | 1 | 0 | 0.10 |
| 1 | 0 | 1 | 1 | 0.0 |
| 1 | 1 | 0 | 0 | 0.0 |
| 1 | 1 | 0 | 1 | 0.35 |
| 1 | 1 | 1 | 0 | 0.30 |
| 1 | 1 | 1 | 1 | 0.0 |

The value of H(Y|X) can be used to define approximate functional dependencies. The farther H(Y|X) is from zero, the weaker the dependency. The maximum value attainable by H(Y|X) is H(Y), when X gives no information about attributes Y. Hence, a reasonable measure, relative to a given system P, is

**Definition:**

$$FD(P;X,Y) = \begin{cases} 1, \text{ if } H(Y)=0 \\ \dfrac{H(Y)-H(Y|X)}{H(Y)}, \text{ otherwise.} \end{cases}$$

If a relation is represented as a probability distribution (Section 2.4), the probabilistic definition of functional dependency applies (see Appendix for proof). Furthermore, any of the commonly encountered inference rules for functional dependencies, loosely referred to as Armstrong's Axioms, can easily be proven sound, if formulated probabilistically, using the algebra of entropy [Malvestuto, 1983]. As an example, consider the augmentation rule: $X \rightarrow Y$ implies $XZ \rightarrow Y$. The proof is simple: Suppose $X \rightarrow_p Y$. Then $H(Y|X)=0$. Since $H(Y|X) \geq H(Y|XZ)$ [Khinchin, 1957, pp. 37-39], $H(Y|XZ)=0$, i.e., $XZ \rightarrow_p Y$.

**3.3 MULTIVALUED DEPENDENCIES.** Advantages of the probabilistic view of databases as presented here are especially apparent when discussing *multivalued dependencies*. Consider two textbook definitions for relational databases:

> Let R be a relation scheme, let X and Y be disjoint subsets of R, and let Z=R-(X∪Y). 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 $t_3(X)=t_1(X)$, $t_3(Y)=t_1(Y)$, and $t_3(Z)=t_2(Z)$ [Maier,1983, p. 124].

> Suppose we are given a relation scheme R and X and Y are subsets of R. Intuitively, we say that $X \rightarrow\rightarrow Y$, read "X multidetermines Y" ... if given values for the attributes of X there is a set of zero or more associated values for the attributes of Y, and this set of Y-values is not connected in any way to values of the attributes in R-X-Y (sic) [Ullman, 1982, p. 243].

After meditating for a while, one realises that what is meant is that $X \rightarrow\rightarrow Y$ if knowledge of $R-(Y-X)$ gives us no more information about Y than does knowledge of X alone. In terms of entropy, the statement is simple:

$$X \rightarrow\rightarrow_p Y \text{ iff } H(Y|X)=H(Y|R-(Y-X)).$$

(As shown in the appendix, the relational and probabilistic formulations are equivalent.)

It is also possible to define multivalued dependencies in terms of join dependencies:

$$X \rightarrow\!\!\!\rightarrow_p Y \text{ iff } *[R-(Y-X), XY];$$

i.e., $X \rightarrow\!\!\!\rightarrow_p Y$ iff the decomposition into the two components R−(Y−X) and XY, which are "coupled" by the set of attributes X, is lossless. In the paper that introduced multivalued dependencies, Fagin presented this alternate definition and proved it for disjoint X and Y [Fagin, 1977, p. 266]. For probabilistic databases, the general case is easily proved. If p is a distribution over any scheme R and X and Y are subsets of R with associated distributions $\downarrow_X(p)$ and $\downarrow_Y(p)$ then

$$\begin{aligned} H(R) &= H(R-(Y-X)) + H(Y-X \mid R-(Y-X)) \\ &= H(R-(Y-X)) + H(Y \mid R-(Y-X)). \end{aligned}$$

In case p over R satisfies *[R-(Y-X),XY], then

$$H(R) = H(R-(Y-X)) + H(Y \mid X)$$

So, $H(Y|X) = H(Y|R-(Y-X))$, i.e., $X \rightarrow\!\!\!\rightarrow_p Y$.
(The converse is proved similarly.)

As with inference rules for functional dependencies, soundness of probabilistic versions of MVD inference rules is easily proved. For example, the complementation rule states that $X \rightarrow\!\!\!\rightarrow Y$ and $Z = R-X-Y$ imply $X \rightarrow\!\!\!\rightarrow Z$.

**Fact 5:** $X \rightarrow\!\!\!\rightarrow_p Y \Rightarrow X \rightarrow\!\!\!\rightarrow_p Z$ where $Z = R-X-Y$.

Proof: Suppose $X \rightarrow\!\!\!\rightarrow_p Y$. Then $H(Y|X) = H(Y|XZ)$.
Therefore, since $H(XYZ) = H(XZ) + H(Y \mid XZ)$

$$\text{and } H(Y \mid X) = H(XY) - H(X),$$

$$\begin{aligned} \text{then } H(XY) - H(X) &= H(Y \mid XZ) = H(XYZ) - H(XZ) \\ &= H(XY) + H(Z \mid XY) - H(XZ). \end{aligned}$$

$$\begin{aligned} \text{So, } -H(X) &= H(Z \mid XY) - H(XZ) \\ &\Rightarrow H(XZ) - H(X) = H(Z \mid XY) \\ &\Rightarrow H(Z \mid X) = H(Z \mid XY), \end{aligned}$$

i.e., $X \rightarrow\!\!\!\rightarrow_p Z$.

The degree of strength of an approximate MVD is reflected by the difference $H(Y|X) - H(Y|XZ)$, which is zero when $X \rightarrow\!\!\!\rightarrow_p Y$. This is actually the information loss when the original distribution p is replaced by the distribution $p^* = (\downarrow_{XY}(p)) * (\downarrow_{XZ}(p))$:

$$\begin{aligned} d(p,p^*) &= H(XZ) + H(Y \mid X) - H(XYZ) \text{ [Lewis,1959]} \\ &= H(Y \mid X) - (H(XYZ) - H(XZ)) \\ &= H(Y \mid X) - H(Y \mid XZ). \end{aligned}$$

This is what one would expect, given Fagin's theorem. A reasonable normalised measure of approximate MVD satisfaction follows.

**Definition:**

$$MVD(P;X,Y) = \begin{cases} 1, \text{ if } H(Y \mid X) = 0 \\ \dfrac{H(Y \mid XZ)}{H(Y \mid X)}, \text{ otherwise.} \end{cases}$$

**3.4 PROBABILISTIC DATA DEPENDENCIES AS CONSTRAINTS.** Usually, data dependencies are viewed as pre-existing constraints restricting admissible database instances to those satisfying the dependencies. Our main emphasis here has been on the discovery that dependencies do or do not exist for a particular instance, or of the degree to which they hold for an instance. However, if a dependency does not hold to an acceptable degree for a particular instance, it cannot hold to that degree for all possible instances; and, if a dependency is found to hold for a particular instance, it may be possible to demonstrate (on other grounds) that it holds for all instances.

This, of course, is not to say that approximate, probabilistic dependencies (APD) cannot be used, e.g., as integrity constraints. As shown, a full-strength APD is equivalent to its corresponding relational dependency. Further, an APD of less than full strength may also be useful as a constraint.

For example, it may be standard practice for a particular application to maintain a level of *JD* satisfaction of at least 0.85. This could arise in a setting in which a distributed monitoring scheme is in place for a set of attributes V (e.g., hardware monitoring, various types of surveillance, etc.) all of which it is not feasible to observe simultaneously for long periods of time, but for which observation over a scheme X is feasible. At the same time, as reflected by the *JD* constraint, it is desired to limit the resulting information loss to an acceptable amount. Periodic observation over the entire set V, and application of the project-join mapping PPJ(X,p) to the sampled distribution p, may be necessary to determine whether the JD constraint is satisfied by the current scheme. If not, an alternate scheme satisfying the *JD* constraint (but also satisfying the additional feasibility constraints) could be found using the data-modelling techniques referred to previously.

## CONCLUSION

Some aspects of a theory of probabilistic databases, applicable also to relational data, have been outlined. This theory is part of a unified approach to data modelling that integrates relational database theory, system theory, and multivariate statistical modelling techniques.

Two areas for further investigation are: the use of probabilistic dependencies as constraints, and the way in which they interact; and the concept of the degree to

which a distribution or relation is identifiable from a given database instance (to which the notion of the "information radius" of a reconstruction family is relevant). Developments in the latter area would be particularly useful for problems of inference and decision-making from the information contained in a database.

## APPENDIX

**A0.** In this appendix, if X and Y are sets, $[X \rightarrow Y]$ denotes the set of all functions from X to Y. Also, if a tuple t is an element of $\underset{v \in X \cup Y}{\times} \mathrm{dom}(v)$ we may denote it by xy where x represents elements of $\underset{v \in X}{\times} \mathrm{dom}(v)$ and y those of $\underset{v \in Y}{\times} \mathrm{dom}(v)$. Note that X and Y need not be disjoint. For example, if $X=\{v_1,v_2\}$ and $Y=\{v_2,v_3\}$, then for a tuple $t \in \underset{v \in X \cup Y}{\times} \mathrm{dom}(v)$, $t=(t_1,t_2,t_3)=xy$, where $x=(t_1,t_2)$ and $y=(t_2,t_3)$.

**A1.** If p satisfies $X \rightarrow_p Y$, then $\mathrm{trans}^{-1}(p)$ satisfies $X \rightarrow Y$.

Proof: Suppose p satisfies $X \rightarrow_p Y$. Then

$$0 = H(Y|X)$$

$$= -\sum_{xy \in \underset{v \in XY}{\times} dom(v)} (p{\downarrow}XY)(xy) \log((p{\downarrow}XY)(xy) / \sum_{\substack{y \in \underset{v \in Y}{\times} dom(v) \\ \text{such that } xy \in \underset{v \in XY}{\times} dom(v)}} (p{\downarrow}XY)(xy))$$

$$\Rightarrow (p{\downarrow}XY)(xy)=0 \text{ or } (p{\downarrow}XY)(xy)=\sum_y (p{\downarrow}XY)(xy), \text{ for all } xy.$$

This implies that for every set of tuples $A \subseteq \underset{v \in XY}{\times} \mathrm{dom}(v)$ that agree on attributes X, at most one $t \in A$ is such that $\downarrow_{XY}(p)(t)>0$. Since $\mathrm{trans}^{-1}(p)(t)=0$ iff $p(t)=0$, it follows that $\downarrow_{XY}(\mathrm{trans}^{-1}(p))(t)>0$ for at most one $t \in A$, where $\downarrow_Z(r)(t)=\max_{a>t} r(t)$. Therefore, $\mathrm{trans}^{-1}(p)$ satisfies $X \rightarrow Y$.

(The proof that r satisfies $X \rightarrow Y$ implies trans(r) satisfies $X \rightarrow_p Y$ is similar.)

**A2.** In section 2.3, for clarity, we defined the probabilistic join in terms of its essential property: it maximizes entropy among the set of distributions that project onto the joined distributions. Alternatively, we could have given a computational definition of $p^*$ and proved that it was the maximum entropy distribution [Brown, 1959; Lewis, 1959]. In the computational definition, probabilities of the join are determined by multiplying probabilities or conditional probabilities associated with tuples of the operand distributions. A basic fact that can be derived from this is that if $\downarrow_Z(p)(b)=0$, then for any database scheme X where $Z \in X$, $PPJ(X,p)(t)=0$, for any $t>b$. The converse of this statement is not true in general, but does hold, e.g., when $|X| = 2$. The proof of the statement in A3 takes advantage of the computational definition, since the maximum entropy member of the reconstruction family is unique, regardless of the way it is determined.

**A3.** p satisfies $X \rightarrow\!\!\!\!\rightarrow_p Y$ implies $\mathrm{trans}^{-1}(p)$ satisfies $X \rightarrow\!\!\!\!\rightarrow Y$.

Proof: Suppose p satisfies $X \rightarrow\!\!\!\!\rightarrow_p Y$. Then p satisfies $^*[R-(Y-X),XY]$ (section 3.3); i.e., $p=p^*=\downarrow_{R-(Y-X)}(p)^*\downarrow_{XY}(p)$, which means that $p^*(t)=p(t)$, for all tuples t.

Case I: $p(t)=0$. Then $p^*(t)=0$ and, as discussed in reference to the computational definition of a two component probabilistic join, this implies that for some $a<t$, where $a \in \underset{v \in R-(Y-X)}{\times} \mathrm{dom}(v)$, $\downarrow_{R-(Y-X)}(p)(a)=0$ or for some $b<t$, where $b \in \underset{v \in XY}{\times} \mathrm{dom}(v)$, $\downarrow_{XY}(p)(b)=0$; then (since $\downarrow_Z(r)(c) = \max_{t>c}\{r(t)\}$ and $p(t)=0 \Rightarrow \mathrm{trans}^{-1}(p)(t)=0$), we must have $\downarrow_{R-(Y-X)}(\mathrm{trans}^{-1}(p))(a)=0$ or $\downarrow_{XY}(\mathrm{trans}^{-1}(p))(b)=0$, since $\sum_{n \in S} n=0 \Rightarrow \max_{n \in S}\{\lceil n \rceil\}=0$, if $0 \le n \le 1$ for all $n \in S$.

But this implies that $r^{\bowtie}(t)=0$, where $r^{\bowtie} = \downarrow_{R-(Y-X)}(r) \bowtie \downarrow_{XY}(r)$.

Case II: $p(t) \neq 0$. $p(t)>0 \Rightarrow p^*(t)>0 \Rightarrow$ for all $a<t$ and $b<t$, $\downarrow_{R-(Y-X)}(p)(a)>0$ and $\downarrow_{XY}(p)(b)>0$. Let $r=\mathrm{trans}^{-1}(p)$. $\downarrow_{R-(Y-X)}(p)(a)>0$ and $\downarrow_{XY}(p)(b)>0$ imply that $\downarrow_{R-(Y-X)}(r)(a)=1$ and $\downarrow_{XY}(r)(b)=1$, which imply that $r^{\bowtie}(t)=1=r(t)$.

So, from (I) and (II), $p^*=p \Rightarrow \mathrm{trans}^{-1}(p)^{\bowtie}=\mathrm{trans}^{-1}(p)$; i.e., if p satisfies $^*[R-(Y-X),XY]$, then $\mathrm{trans}^{-1}(p)$ satisfies $\bowtie[R-(Y-X),XY]$. By Fagin's theorem (section 3.3), this is equivalent to: p satisfies $X \rightarrow\!\!\!\!\rightarrow_p Y$ implies $\mathrm{trans}^{-1}(p)$ satisfies $X \rightarrow\!\!\!\!\rightarrow Y$.

(r satisfies $X \rightarrow\!\!\!\!\rightarrow Y$ implies trans(r) satisfies $X \rightarrow\!\!\!\!\rightarrow_p Y$ is easily proved.)

**A4.** r satisfies $\bowtie[X]$ implies trans(r) satisfies $^*[X]$.

Proof: Suppose r satisfies $\bowtie[X]$. Then $r = r^{\bowtie}$; in particular, $r(t)=0 \Rightarrow r^{\bowtie}(t)=0$. But this means that zeros are preserved by the project-join mapping on trans(r), i.e., $\mathrm{trans}(r)(t)=0 \Rightarrow \mathrm{trans}(r)^*(t)=0$. By definition, $\mathrm{trans}(r)^*$ is the distribution with maximum entropy among the set D of distributions whose projections onto sets of attributes in X equal those of trans(r). Let $W=\{p \mid p$ is a distribution over T and $\mathrm{trans}(r)(t)=0 \Rightarrow p(t)=0\}$. Then $H(\mathrm{trans}(r)) = \max\{H(p) \mid p \in W\}$. This follows from the fact that all non-zero components of trans(r) are equal to each other, for any r, and the following three properties of entropy:

1) the ordering of the components of a probability distribution does not affect its entropy

2) the entropy of an n+1-component distribution, q, whose n+1th component is zero is equal to that of an

n-component distribution whose components are equal to components 1,...,n of q

3) the distribution with maximum entropy among the set of all n-component distributions is the distribution whose components are all equal to 1/n [Acsel and Daroczy, 1975].

Since trans(r)*∈W, H(trans(r)*)≤H(trans(r)). But, since trans(r)∈D, H(trans(r))≤H(trans(r)*). Therefore, H(trans(r))=H(trans(r)*). But this implies that trans(r)=trans(r)*, since trans(r)∈D and the maximum entropy element of D is unique.

It follows immediately from this that

$$r=r^{\bowtie} => \text{trans}^{-1}(\text{trans}(r)^*)=r^{\bowtie}.$$

As we observed in section 2.5, it is not in general true that p satisfies *[X] implies $\text{trans}^{-1}(p)$ satisfies $\bowtie$[X].

**A5. Definition:** The *information radius* of a reconstruction family $R_X$, $ir(R_X)$, is

$$ir(R_X)=\max_{p\in R_X} d(p,p_X^*),$$

where d is directed divergence and $p_X^*$ is the maximum entropy element of $R_X$; i.e., $p_X^*=PPJ(X,p)$, for any $p\in R_X$.

**Fact:** $\max_{p\in R_X} d(p,p_X^*)$ is achieved when p is a vertex of $R_X$.

**Proof:** For fixed q, d(p,q) is a convex function of p, over the set $P^n$ [Kumar et al, 1986]. Therefore, since $R_X\subseteq P^n$, $d(p,p_X^*)$ is a convex function of p over $R_X$. The maximum value of any convex function defined on a bounded polyhedral set is achieved for one or more vertices of the set.

**Fact:** $R_X\subseteq R_Y$ implies $ir(R_X)\leq ir(R_Y)$.

**Proof:** Assume $R_X\subseteq R_Y$. Let $v_X$ and $v_Y$ denote, respectively, the maximising distributions p for $d(p,p_X^*)$ and $d(p,p_Y^*)$. Since $R_X\subseteq R_Y$, $H(p_Y^*)\geq H(p_X^*)$. For any reconstruction family $R_Z$, $d(p,p_Z^*)=H(p_Z^*)-H(p)$ [Higashi, 1984]. Therefore,

$$\begin{aligned} d(v_Y,p_Y^*) &\leq d(v_X,p_Y^*) && [v_X\in R_Y;\ \text{def. } v_Y] \\ &= H(p_Y^*)-H(v_X) && [v_X\in R_Y] \\ &\geq H(p_X^*)-H(v_X) && [H(p_Y^*)\geq H(p_X^*)] \\ &= d(v_X,p_X^*). \end{aligned}$$

So, $d(v_Y,p_Y^*)\geq d(v_X,p_X^*)$; i.e., $ir(R_X)\leq ir(R_Y)$.

**Corollary:** $X\leq Y$ implies $ir(R_X)\geq ir(R_Y)$.

**Proof:** $X\leq Y$ implies $R_Y\subseteq R_X$, which implies $ir(R_X)\geq ir(R_Y)$.

## REFERENCES


Acsel, J. and Daroczy, Z., 1975. *On Measures of Information and Their Characterizations,* Academic Press, New York.

Ashby, W.R., 1956. *An Intoduction to Cybernetics,* Methuen, London.

Ashby, W.R., 1965. "Constraint Analysis of Many-dimensional Relations". In: *Progress in Biocybernetics,* v.2, N. Wiener and J.P. Schade, Eds., Elsevier, Amsterdam, pp. 10-18.

Bishop, Y., Fienberg, S. and Holland, P., 1975. *Discrete Multivariate Analysis,* MIT Press, Cambridge, MA.

Bourbaki, N., 1954. *Theorie des Ensembles,* Hermann Cie, Paris.

Brodie, M.L., 1984. "On the development of data Models", in: *On Conceptual Modelling,* M.L. Brodie, J. Myopoulos, and J. W. Schmidt, Eds., Springer-Verlag, N.Y.

Brown, D.T., 1959. "A Note on Approximations to Discrete Probability Distributions", *Information and Control,* v.2, n.4, pp. 386-392.

Cavallo, R.,1980. "Reconstructability and Identifiability in the Evaluation of Structure Hypotheses: an Issue in the Logic of Modelling". In: *Systems Science and Science,* SGSR, Louisville, pp. 647-654.

Cavallo, R. and Klir, G., 1979a. "Reconstructability Analysis of Multi-dimensional Relations: a Theoretical Basis for Computer-aided Determination of Acceptable Systems Models", *Int. J. General Systems,* v.5, n.3, pp. 143- 171.

Cavallo, R. and Klir, G., 1979b. "The Structure of Reconstructable Relations: a Comprehensive Study", *J. of Cybernetics,* v.9, n.4, pp. 399-413.

Cavallo, R. and Klir, G., 1981. "Reconstructability Analysis: Evaluation of Reconstruction Hypotheses", *Int. J. of General Systems,* v.7, n.1, pp. 7-32.

Codd, E.F., 1970. "A Relational Model of Data for Large Shared Data Banks", *CACM,* v.13, n.6, pp.377-387.

Denning, D.,1982. *Cryptography and Data Security,* Addison-Wesley, Reading, MA.

Fagin, R., 1977. "Multivalued Dependencies and a New Normal Form for Relational Databases", *ACM TODS,* v.2, n.3, 262-278.

Fagin, R., 1983. "Degrees of Acyclicity for Hypergraphs and Relational Database Schemes", *JACM,* v.30, n.3, pp.514-550.

Fagin, R., Mendelson, A. and Ullman, J., 1982. "A Simplified Universal Relation Assumption and Its Properties", *ACM TODS,* v.7, n.3, pp. 343-360.

Good, I.J., 1983. "The Philosophy of Exploratory Data Analysis", *Philosophy of Science,* v.50, pp. 283-295.

Higashi, M., 1984. *A Systems Modelling Methodology: Probabilistic and Possibilistic Approaches,* Ph.D dissertation, Systems Science Dept, SUNY-Binghamton.

Jaynes, E., 1979. "Where Do We Stand on Maximum

Entropy?". In: *The Maximum Entropy Formalism,* Levine, R. and Tribus, M., Eds., MIT Press, Cambridge, MA., pp. 15-118.

Khinchin, A., 1957. *Mathematical Foundations of Information Theory,* Dover, New York.

Kolmogorov, A.N., 1965. "Three Approaches to the Quantitative Definition of Information". In: *Problems of Information Transmission,* v.1, n.1, pp. 1-7.

Kullback, S., 1959. *Information Theory and Statistics,* Wiley, New York.

Kumar, U. et al, 1986. "Some Normalized Measures of Directed Divergence", *Int. J. General Systems,* v.13, n.1, pp. 5-16.

Lewis, P.M., 1959. "Approximating Probability Distributions to Reduce Storage Requirements", *Information and Control,* v.2, n.3, pp. 214-225.

Madden, R.F. and Ashby, W.R., 1972. "On the Identification of Many-dimensional Relations", *Int. J. Systems Science,* v.3, n.4, pp.343-356.

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

Malvestuto, F.M., 1983. "Theory of Random Observables in Relational Data Bases", *Information Systems,* v.8, n.4, pp. 281-289.

Nambiar, K., 1980. "Some Analytic Tools for the Design of Relational Database Systems", *Sixth Int. Conf. on VLDB,* IEEE, pp. 417-428.

Ullman, J., 1982. *Principles of Database Systems,* 2nd ed., Computer Science Press, Rockville, MD.

Wiener, N., 1914. "A Simplification of the Logic of Relations", *Proc. Cambridge Philosophical Society,* v.17, pp. 387-390.