An information-theoretic analysis of worst-case redundancy in database design

Citation for published version:
Kolahi, S & Libkin, L 2010, 'An information-theoretic analysis of worst-case redundancy in database design', *ACM Transactions on Database Systems*, vol. 35, no. 1, 5, pp. 1-32.
https://doi.org/10.1145/1670243.1670248

Digital Object Identifier (DOI):
10.1145/1670243.1670248

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
ACM Transactions on Database Systems

An information-theoretic analysis of worst-case redundancy in database design

SOLMAZ KOLAHI
University of British Columbia
and
LEONID LIBKIN
University of Edinburgh

Normal forms that guide the process of database schema design have several key goals such as elimination of redundancies and preservation of integrity constraints, such as functional dependencies. It has long been known that complete elimination of redundancies and complete preservation of constraints cannot be achieved simultaneously. In this paper, we use a recently-introduced information-theoretic framework, and provide a quantitative analysis of the redundancy/integrity preservation tradeoff, and give techniques for comparing different schema designs in terms of the amount of redundancy they carry.

The main notion of the information-theoretic framework is that of an information content of each datum in an instance (which is a number in $[0, 1]$): the closer to 1, the less redundancy it carries. We start by providing a combinatorial criterion that lets us calculate, for a relational schema with functional dependencies, the lowest information content in its instances. This indicates how good the schema design is in terms of allowing redundant information. We then study the normal form 3NF, which tolerates some redundancy to guarantee preservation of functional dependencies. The main result provides a formal justification for normal form 3NF by showing that this normal form pays the smallest possible price, in terms of redundancy, for achieving dependency preservation. We also give techniques for quantitative comparison of different normal forms based on the redundancy they tolerate.

Categories and Subject Descriptors: H.2.1 [**Database Management**]: Logical Design—*normal forms*; *schema and subschema*; H.1.1 [**Models and Principles**]: Systems and Information Theory—*information theory*; *value of information*

General Terms: Design, Management, Theory

Additional Key Words and Phrases: Database design, functional dependency, redundancy, Third Normal Form (3NF)

1. INTRODUCTION

One of the most important factors in maintaining the integrity or correctness of a database is controlling the redundancy of data. Normal forms have long been

Authors' addresses: S. Kolahi, Department of Computer Science, University of British Columbia, 2366 Main Mall, Vancouver, BC, V6T 1Z4, Canada, e-mail: solmaz@cs.ubc.ca; L. Libkin, School of Informatics, University of Edinburgh, Informatics Forum, IF 5.33, 10 Crichton Street, Edinburgh, EH8 9AB, UK, e-mail: libkin@inf.ed.ac.uk.

studied as a means of reducing redundancies caused by data dependencies, such as functional and multivalued dependencies, in the process of schema design. In traditional normalization theory, a database is characterized as either redundant or non-redundant. However, between two databases that carry redundancy, one may be significantly worse than the other. Moreover, normalizing a database into a perfectly non-redundant design usually comes with a cost of slower query answering. Our goal in this paper is to provide richer guidelines that help a database designer have a better understanding of the amount of redundancy carried by data and make design decisions accordingly.

We use a recently introduced information-theoretic tool [Arenas and Libkin 2005] that actually *measures* the amount of redundancy in a database. Given a database instance I of a schema S with integrity constraints Σ, the information-theoretic measure, called *relative information content*, assigns a number $\text{RIC}_I(p \mid \Sigma)$ to every position p in the instance that contains a data value, where a position is specified by a tuple in the instance and an attribute name. This number ranges between 0 and 1 and shows how much redundancy is carried by position p. Intuitively, if $\text{RIC}_I(p \mid \Sigma) = 1$, then p carries the maximum possible amount of information: nothing about it can be inferred from the rest of the instance. Smaller values of $\text{RIC}_I(p \mid \Sigma)$ show that positions carry some amount of redundancy, as some information about them can be inferred. A *well-designed* schema is the one that guarantees maximum information content for data values in every instance.

Using this framework, normal forms, such as BCNF and 4NF, that completely eliminate the possibility of redundancies were justified by showing that these normal forms always ensure well-designed schemas. Perfectly non-redundant designs, however, are not always the best choice for a number of reasons. First, it may not be possible to normalize a database into a perfect normal form without losing some of the integrity constraints. For instance, normalizing a relation schema into BCNF may not be possible without losing some of the functional dependencies. Second, a normalization that completely eliminates redundancies may lead to producing too many relations, which will slow down the queries by requiring more joins. These are the reasons that a more forgiving normal form such as 3NF is more popular in practice. In fact, practical database design tips (e.g., in books [Greenwald et al. 2007; Stephens and Plew 2002; Dewson 2006]) usually refer to a "normalized" database schema as a schema that satisfies 3NF.

Our goal is to provide guidelines on how to choose the least-redundant design in case a non-redundant one is not achievable. This could be an important choice in the process of schema design, because the more redundant a database is, the more it is prone to anomalies and inconsistencies after a series of insertions or updates. The following example illustrates a situation, in which a precise redundancy analysis of schemas could be helpful in making a better design decision.

Example 1.1. Consider a relation schema $R(A, B, C, D, E)$ with functional dependencies $\Sigma = \{AB \to C, C \to B, D \to E\}$. One can easily think of instances of this schema that store redundant values in columns B, C, and E. To remove or lower these redundancies, any of the following designs could be considered, all of which ensure a lossless join and preserve the functional dependencies:

(1) $R_1(A, B, C, D)$, $R_2(D, E)$;

(2) $R_1(A,B,C)$, $R_2(A,B,D,E)$;
(3) $R_1(A,B,C)$, $R_2(A,B,D)$, $R_3(D,E)$.

Observe that here a perfectly non-redundant BCNF design that also preserves the functional dependencies does not exist. Furthermore, having more relations would impose the cost of additional joins for query answering. To choose any of these three designs or the original design with a single relation, the designer could use a quantitative analysis of the redundancy caused by functional dependencies in the instances of these schemas. A natural approach is thinking of a way to measure the highest or worst possible redundancy in all valid instances of these schemas. □

We introduce a measure of worst redundancy for database schemas and normal forms to be able to compare database designs with redundancy. The *guaranteed information content* of a schema with integrity constraints is the lowest information content ever found in the instances of the schema, and indicates how much redundancy the schema allows in the worst-case scenario. We give a combinatorial criterion that lets us calculate the guaranteed information content and thus show how good a schema is redundancy-wise. This could be a useful indicator of whether the schema needs to be further normalized in case the instances are potentially too redundant. We also study the complexity of calculating the guaranteed information content.

Guaranteed information content can also be defined for a normal form as the lowest information content ever found in instances of schemas that satisfy the normal form, and indicates how much redundancy the normal form tolerates. An application of such an analysis provides a justification for third normal form (3NF). The main property possessed by 3NF, but not by BCNF, is *dependency preservation*: for every schema, there always exists a lossless decomposition into 3NF that preserves all the functional dependencies. That is, the set of functional dependencies on the original schema is equivalent to the set of projected functional dependencies on the decomposed schemas. This is a very important property for integrity enforcement, as DBMSs provide a variety of mechanisms to ensure that integrity constraints are enforced during updates. Keeping all the constraints in the form of functional dependencies makes the integrity enforcement much faster since enforcing functional dependencies does not require joins across different relations.

Notice that it is not always possible to do a dependency-preserving BCNF normalization to achieve a well-designed schema (the smallest example is the 3NF schema $R(A,B,C)$ with FDs $\Sigma = \{AB \to C, C \to B\}$, which does not admit a lossless dependency-preserving BCNF decomposition). Consequently, to guarantee the integrity of the database, *some* redundancy must be tolerated. A natural question is then whether 3NF is the right choice of a dependency-preserving normal form. To be more precise, consider every possible normal form, defined as a set of restrictive conditions on FDs, such that every schema admits an FD-preserving decomposition that satisfies the normal form (clearly, BCNF would not be among these normal forms). Now if we apply the information-theoretic approach to measure the amount of redundancy introduced by these normal forms, will 3NF be the one with the least amount of redundancy? Our second main result gives a positive answer to this question.

Our last goal is to provide quantitative techniques for comparing different normal forms. The motivation comes from the following question: if we know that in BCNF designs, the value of $\text{RIC}_I(p \mid \Sigma)$ is always 1, can we find a constant $c < 1$ so that $\text{RIC}_I(p \mid \Sigma) > c$ for all 3NF designs? A strong negative answer was given by [Kolahi 2007] which showed that or every $\varepsilon > 0$, one can find a 3NF schema with a set Σ of functional dependencies, an instance of that schema, and a position p such that $\text{RIC}_I(p \mid \Sigma) < \varepsilon$. However, this is not particularly surprising: it has long been known [Ling et al. 1981; Zaniolo 1982; Biskup and Meyer 1987] that for some schemas already in 3NF, better 3NF designs can be produced by the standard synthesis algorithm. Hence, an arbitrary 3NF schema may have quite a bit of extra redundancy.

This gives rise to the following question: what can be said about arbitrary 3NF schemas, not only the good ones that ensure the minimum price of dependency preservation? Can they be as bad as arbitrary schemas? How do they compare to "good" 3NF designs? To answer these questions, we compare dependency-preserving normal forms based on their guaranteed information content or the maximum redundancy that they tolerate in the instances of schemas that satisfy those normal forms. Our next main result formally confirms that some 3NF schemas may have more redundancy than others, but it also shows that arbitrary 3NF schemas have at least twice the information content compared to unnormalized schemas.

The rest of the paper is organized as follows: in Section 2, we summarize necessary background information on relational normalization, information theory, and the definition and known applications of the information content measure. In Section 3, we give the definition of guaranteed information content for a schema and show how we can calculate it. We define and calculate guaranteed information content for normal forms and provide a justification for 3NF in Section 4. We give concluding remarks in Section 5.

Remark Some of the results of Section 4 have previously appeared in a conference proceedings version [Kolahi and Libkin 2006]. Specifically, in [Kolahi and Libkin 2006], we presented results about the notion of guaranteed information content. The results about the analogous notion based on the average information content, as well as results in Section 3 are new and have not been previously published.

2. BACKGROUND

2.1 Relational databases and normal forms

A *relation schema* consists of a relation name R and a set $U = \{A_1, \ldots, A_m\}$ of attribute names. We sometimes write $R(A_1, \ldots, A_m)$ and refer to U as $sort(R)$. A *database schema* is a set of relation schemas $S = \{R_1, \ldots, R_\ell\}$. In this paper, we assume that elements of database instances come from a countably-infinite domain; to be concrete, we assume it to be $\mathbb{N}^+$, the set of positive integers. Therefore, an instance I of a database schema S assigns to each m-attribute relation R in S a finite set $I(R)$ of tuples, where a tuple is a function $t : sort(R) \to \mathbb{N}^+$ (equivalently, it is an element of ${\mathbb{N}^+}^m$). We let $adom(I)$ stand for the *active domain* of I: the set of all elements of $\mathbb{N}^+$ that occur in I. The size of $I(R)$ is defined as $\|I(R)\| = |sort(R)| \cdot |I(R)|$, and the size of I is $\|I\| = \sum_{R \in S} \|I(R)\|$.

Given an instance I, a *position* in I is a triple (R, t, A), where R is a relation name in S, t is a tuple in $I(R)$, and A is an attribute of $sort(R)$. With each position we associate a value stored there, that is, $t[A]$, when t is viewed as a finite function. The set of all positions is denoted by $Pos(I)$. Note that $\|I\|$ equals the cardinality of $Pos(I)$.

Schemas may contain *integrity constraints*, in which case we refer to schemas (S, Σ), where S is a set of relation names and Σ is a set of constraints. We usually write (R, Σ) instead of the more formal $(\{R\}, \Sigma)$ in case of one relation. In this paper, we are only interested in *functional dependencies (FDs)* on a relation R, which are expressions of the form $X \to Y$, where both X and Y are nonempty subsets of $sort(R)$. An instance $I(R)$ satisfies $X \to Y$, written as $I(R) \models X \to Y$, if for every two tuples $t_1, t_2 \in I(R)$, $t_1[X] = t_2[X]$ implies $t_1[Y] = t_2[Y]$. We let $inst(S, \Sigma)$ stand for the set of all instances of S satisfying Σ and $inst_k(S, \Sigma)$ for the set of instances $I \in inst(S, \Sigma)$ with $adom(I) \subseteq [1, k]$.

We say that a functional dependency $X \to Y$ is *trivial* if $Y \subseteq X$. A key dependency is a functional dependency of the form $X \to sort(R)$. Then we say that X is a *superkey* for the relation. If there is no superkey Y such that $Y \subsetneq X$ then we say that X is a *candidate key* or just a *key.* If Σ is a set of FDs, then Σ^+ denotes the set of all FDs $X \to Y$ implied by it ($\Sigma \models X \to Y$). Given a set of attributes $X \subseteq sort(R)$, the *closure* of X, written as X^+, is defined as the set of attributes $\{A \mid \Sigma \models X \to A\}$.

We now review the most basic definitions of relational normalization theory, and refer the reader to surveys [Beeri et al. 1978; Kanellakis 1990; Biskup 1995] and texts [Abiteboul et al. 1995; Kifer et al. 2006; Levene et al. 1999] for additional information.

A schema (R, Σ) is in BCNF if for every nontrivial functional dependency $X \to Y \in \Sigma^+$, X is a superkey. A database schema S is in BCNF if every relation in S is in BCNF. We say that an attribute $A \in sort(R)$ is *prime* if it is an element of some key of R. A schema (R, Σ) is in 3NF if for every nontrivial functional dependency $X \to A \in \Sigma^+$, X is a superkey or A is prime. We say that a database schema S is in 3NF if every relation schema in S is in 3NF.

Given a database schema $S = (R, \Sigma)$ and some normal form $\mathcal{NF}$, an *$\mathcal{NF}$-decomposition* is another schema $S' = \{(R_1, \Sigma_1), \dots, (R_\ell, \Sigma_\ell)\}$ such that for every $i \in [1, \ell]$, $(R_i[U_i], \Sigma_i)$ satisfies normal form $\mathcal{NF}$. The decomposition is *lossless* [Aho et al. 1979] if for every instance I of S there is an instance I' of S' such that for every $i \in [1, \ell]$, $I'(R_i) = \pi_{sort(R_i)}(I)$, and $I = I'(R_1) \bowtie \dots \bowtie I'(R_\ell)$. This property ensures that any instance of the original schema can be reconstructed by joining the instances of the decomposed schema. We say that S' is a *dependency-preserving* decomposition of S if $\left(\bigcup_{i=1}^{\ell} \Sigma_i\right)^+ = \Sigma^+$. That is, $\bigcup_{i=1}^{\ell} \Sigma_i$ and Σ are equivalent. This property ensures that the constraints remain in the form of functional dependencies after the decomposition, which makes the integrity enforcement more efficient since enforcing FDs does not require joins across different relations.

It is known that every schema can be decomposed into lossless BCNF and 3NF schemas. However, only 3NF decompositions are guaranteed to be dependency-preserving. That is, for some schemas no lossless BCNF decomposition exists that is also dependency-preserving. The smallest example of such schemas is $R(A, B, C)$ and $\Sigma = \{AB \to C, C \to B\}$.

2.2 Information theory

Entropy is a fundamental concept in information theory that is defined to measure the amount of information provided by a certain event. Assume that an event can have n different outcomes $s_1, \ldots, s_n$, each with probability $p_i, i \in [1, n]$, Then the *entropy* of the probability distribution $\mathcal{A} = (\{s_1, \ldots, s_n\}, P_{\mathcal{A}})$ is defined as

$$H(\mathcal{A}) \;=\; \sum_{i=1}^{n} P_{\mathcal{A}}(s_i) \log \frac{1}{P_{\mathcal{A}}(s_i)},$$

which shows how much information is gained on average by knowing that one of the $s_1, \ldots, s_n$ outcomes has occurred. For probabilities that are zero, we adopt the convention that $0 \log \frac{1}{0} = 0$, since we have $\lim_{x \to 0} x \log \frac{1}{x} = 0$. It is known that $0 \le H(\mathcal{A}) \le \log n$, with $H(\mathcal{A}) = \log n$ only for the uniform distribution $P_{\mathcal{A}}(s_i) = 1/n$ [Cover and Thomas 1991].

For two probability spaces $\mathcal{A} = (\{s_1, \ldots, s_n\}, P_{\mathcal{A}})$, $\mathcal{B} = (\{s'_1, \ldots, s'_m\}, P_{\mathcal{B}})$, and probabilities $P(s'_j, s_i)$ of all the events (s'_j, s_i) ($P_{\mathcal{A}}$ and $P_{\mathcal{B}}$ may not be independent), the *conditional entropy* of $\mathcal{B}$ given $\mathcal{A}$, denoted by $H(\mathcal{B} \mid \mathcal{A})$, gives the average amount of information provided by $\mathcal{B}$ if $\mathcal{A}$ is known [Cover and Thomas 1991]. If $P(s'_j \mid s_i) = P(s'_j, s_i)/P_{\mathcal{A}}(s_i)$ are conditional probabilities, then

$$H(\mathcal{B} \mid \mathcal{A}) = \sum_{i=1}^{n} \Bigg(P_{\mathcal{A}}(s_i) \sum_{j=1}^{m} P(s'_j \mid s_i) \log \frac{1}{P(s'_j \mid s_i)} \Bigg).$$

2.3 Information theory and normalization

An information-theoretic framework was recently proposed [Arenas and Libkin 2005] to justify relational normal forms and to provide a test of “goodness” of normal forms for other data models. This framework is completely independent of the notions of update or query languages, and is based on the intrinsic properties of the data. Unlike previously proposed information-theoretic measures [Lee 1987; Cavallo and Pittarelli 1987; Dalkilic and Robertson 2000; Levene and Loizou 2003], this measure takes into account both data and schema constraints.

Given a database schema S, a set of constraints Σ, and an instance I of (S, Σ), the information-theoretic measure assigns a number to every position p in the instance that contains a data value, by calculating a conditional entropy of a certain probability distribution and then normalizing to the interval $[0, 1]$. This number, which is called *relative information content* with respect to constraints Σ and is written as $\text{RIC}_I(p \mid \Sigma)$, ranges between 0 and 1 and shows how much redundancy is carried by position p. Intuitively, if $\text{RIC}_I(p \mid \Sigma) = 1$, then p carries the maximum possible amount of information: nothing about it can be inferred from the rest of the instance. Smaller values of $\text{RIC}_I(p \mid \Sigma)$ show that positions carry some amount of redundancy, as some information about them can be inferred. Next we give a formal definition of this measure.

Relative Information Content. We now present the formal definition of the information content measure as defined in [Arenas and Libkin 2005]. Fix a schema S and a set Σ of constraints, and let $I \in \mathit{inst}(S, \Sigma)$ with $\|I\| = n$. Recall that the set of positions in I, denoted by $\mathit{Pos}(I)$, is defined as the set $\{(R, t, A) \mid R \in S,\ t \in I(R),$

and $A \in sort(R)\}$. We now want to define $\text{R}\textsc{ic}_I(p \mid \Sigma)$, the relative information content of a position $p \in Pos(I)$ with respect to the set of constraints Σ, and we want this value to be normalized to the interval $[0, 1]$. We shall first define, for all k, a measure $\text{R}\textsc{ic}_I^k(p \mid \Sigma)$ that works when instances are taken from the set $inst_k(S, \Sigma)$. Since the maximum value of entropy for a discrete distribution on k elements is $\log k$, we then take the limit of the ratio $\frac{\text{R}\textsc{ic}_I^k(p|\Sigma)}{\log k}$ as $k \to \infty$ to get a number in $[0, 1]$ that does not depend on k.

This is a measure of the amount of redundancy, so intuitively, we want to measure how much, on average, the value of position p is determined by any set of positions in I. For that, we take a set $X \subseteq Pos(I) - \{p\}$, and assume that the values in those positions X are lost, and then someone restores them from $[1, k]$. Then, we measure how much information about the value in p is provided by this restoration by calculating the entropy of a suitably chosen distribution of all distinct instances that could be obtained as an outcome of the restoration. The average such measure over all sets $X \subseteq Pos(I) - \{p\}$ is defined as $\text{R}\textsc{ic}_I^k(p \mid \Sigma)$.

We now define this formally. Fix an instance I. The reference to the instance I will be removed from probabilities calculated below to reduce the clutter. We (arbitrarily) assign position numbers $1, \ldots, n$ to the positions of I (where $n = \|I\| = |Pos(I)|$) and fix an n-element set of variables $\{v_i \mid 1 \leq i \leq n\}$. Now fix a position $p \in Pos(I)$, and let $\Omega(I, p)$ be the set of all 2^{n-1} vectors $(a_1, \ldots, a_{p-1}, a_{p+1}, \ldots, a_n)$ such that for every $i \in [1, n] - \{p\}$, a_i is either v_i or the value in the i-th position of I. We make this into a probability space $\mathcal{A}(I, p) = (\Omega(I, p), P_u)$ with the uniform distribution $P_u(\bar{a}) = 2^{1-n}$.

We next define conditional probabilities $P_k(a \mid \bar{a})$, for $a \in [1, k]$, that show how likely a is to occur in position p, if values are removed from I according to the tuple $\bar{a} \in \Omega(I, p)$. Let $I_{(a,\bar{a})}$ be obtained from I by putting a in position p, and a_i in position $i \neq p$. A substitution is a map $\sigma : \bar{a} \to [1, k]$ that assigns a value to each a_i which is a variable, and leaves other a_i's intact. We let $\text{SAT}_\Sigma^k(I_{(a,\bar{a})})$ be the set of all substitutions σ such that $\sigma(I_{(a,\bar{a})}) \models \Sigma$ and $|\sigma(I_{(a,\bar{a})})| = |I|$ (the latter ensures that no two tuples collapse as the result of applying σ). Then $P_k(a \mid \bar{a})$ is defined as:

$$P_k(a \mid \bar{a}) \;=\; \frac{|\text{SAT}_\Sigma^k(I_{(a,\bar{a})})|}{\sum_{b \in [1,k]} |\text{SAT}_\Sigma^k(I_{(b,\bar{a})})|}.$$

With this, we define $\text{R}\textsc{ic}_I^k(p \mid \Sigma)$ as

$$\sum_{\bar{a} \in \Omega(I,p)} \left(\frac{1}{2^{n-1}} \sum_{a \in [1,k]} P_k(a \mid \bar{a}) \log \frac{1}{P_k(a \mid \bar{a})} \right).$$

Since $\sum_{a \in [1,k]} P_k(a \mid \bar{a}) \log \frac{1}{P_k(a|\bar{a})}$ measures the amount of information in p, given constraints Σ and some missing values in I, represented by the variables in $\bar{a}$, the measure $\text{R}\textsc{ic}_I^k(p \mid \Sigma)$ is the average such amount over all $\bar{a} \in \Omega(I, p)$.

To see that $\text{R}\textsc{ic}_I^k(p \mid \Sigma)$ is a conditional entropy, we define a probability distribution on $[1, k]$ as follows:

$$P'_k(a) \;=\; \frac{1}{2^{n-1}} \sum_{\bar{a} \in \Omega(I,p)} P_k(a \mid \bar{a}) \;.$$

I_1

A	B	C	D
1	2	3	4
1	2	3	5

$\textsc{Ric}_{I_1}(p_1 \mid \Sigma_1) = 0.875$
$\textsc{Ric}_{I_1}(p_1 \mid \Sigma_2) = 0.781$

I_2

A	B	C	D
1	2	3	4
1	2	3	5
1	2	3	6

$\textsc{Ric}_{I_2}(p_2 \mid \Sigma_1) = 0.781$
$\textsc{Ric}_{I_2}(p_2 \mid \Sigma_2) = 0.629$

I_3

A	B	C	D
1	2	3	4
1	2	3	5
1	2	3	6
1	2	3	7

$\textsc{Ric}_{I_3}(p_3 \mid \Sigma_1) = 0.711$
$\textsc{Ric}_{I_3}(p_3 \mid \Sigma_2) = 0.522$

Fig. 1. Information content vs. redundancy, where $\Sigma_1 = \{A \rightarrow C\}$ and $\Sigma_2 = \{A \rightarrow C, B \rightarrow C\}$.

Intuitively, this probability shows how likely an element from $[1, k]$ is to satisfy Σ when put in position p, given all possible interactions between p and sets of positions in I. If $\mathcal{B}^k_\Sigma(I, p)$ is the probability space $([1, k], P'_k)$, then $\textsc{Ric}^k_I(p \mid \Sigma)$ is the conditional entropy:

$$\textsc{Ric}^k_I(p \mid \Sigma) \;=\; H(\mathcal{B}^k_\Sigma(I, p) \mid \mathcal{A}(I, p)).$$

Since the domain of $\mathcal{B}^k_\Sigma(I, p)$ is $[1, k]$, we have $0 \leq \textsc{Ric}^k_I(p \mid \Sigma) \leq \log k$. To normalize this, we consider the ratio $\textsc{Ric}^k_I(p \mid \Sigma)/\log k$. Note that when k, the domain size, increases, the values $P_k(a \mid \bar{a})$ and $\textsc{Ric}^k_I(p \mid \Sigma)$ would change as the number of valid substitutions in $\text{SAT}^k_\Sigma(I_{(a,\bar{a})})$ would increase. It was however shown [Arenas and Libkin 2005] that for most reasonable constraints Σ (certainly for all constraints definable in first-order logic, such as functional, multi-valued, and join dependencies), the sequence of ratios $\textsc{Ric}^k_I(p \mid \Sigma)/\log k$ converges as $k \rightarrow \infty$, and we thus define

$$\textsc{Ric}_I(p \mid \Sigma) \;=\; \lim_{k\to\infty} \frac{\textsc{Ric}^k_I(p \mid \Sigma)}{\log k}.$$

If $\textsc{Ric}_I(p \mid \Sigma) = 1$, the information carried by position p is at a maximum, and there is no redundancy in position p. If the data in position p is redundant, and the value in this position can be inferred from the rest of the instance and constraints, then $\textsc{Ric}_I(p \mid \Sigma)$ gets a value in $[0, 1)$ to show how redundant the value in position p is.

Example 2.1. Consider relation $R(A, B, C, D)$, two sets of FDs $\Sigma_1 = \{A \rightarrow C\}$ and $\Sigma_2 = \{A \rightarrow C, B \rightarrow C\}$, and three instances I_1, I_2, and I_3 in Fig. 1 that are in both $inst(R, \Sigma_1)$ and $inst(R, \Sigma_2)$. Let p_1, p_2, p_3 denote the position of the gray cells in the instances. We observe that the information content of the gray cell decreases as it becomes more redundant by adding tuples that could determine the value of attribute C in that position. We also see how the information content changes when we have an additional constraint $B \rightarrow C$ that makes the gray cell even more redundant. This example intuitively shows how the value of information content varies between 1 and 0 as a decreasing function of redundancy. □

Justifying Perfect Normal Forms. Ideally, we want databases in which every position carries the maximum amount of information. The notion of being *well-designed* is accordingly defined as follows [Arenas and Libkin 2005]:

Definition 2.2. A database schema S with a set of constraints Σ is *well-designed* if for every instance $I \in \mathit{inst}(S, \Sigma)$ and every position $p \in \mathit{Pos}(I)$, $\textsc{Ric}_I(p \mid \Sigma) = 1$.

In other words, well-designed databases are the ones that allow absolutely no redundancy in any position. It is known [Arenas and Libkin 2005] that this definition corresponds exactly to the definition of having no redundancy by Vincent [Vincent 1999], which calls a data value v in instance I redundant if replacing v with any other value would violate the constraints. Using the notion of being well-designed, well-known normal forms, such as BCNF and 4NF, have been justified, and the corresponding normalization algorithms have been proved to always produce a well-designed database.

Although some of these normal forms were previously justified by showing that they eliminate the possibility of redundancy or update anomalies [Bernstein and Goodman 1980; LeDoux and Parker 1982; Fagin 1979; 1981; Vincent 1999; Levene and Vincent 2000], the information-theoretic technique provided the first justification for each step of the normalization algorithms [Arenas and Libkin 2005] by showing that these steps never decrease the amount of information content in any position of an instance. Furthermore, since the information-theoretic framework enables us to measure the amount of redundancy in databases that are not well-designed, it can be used to justify normal forms, such as 3NF, that do not completely eliminate redundancies, as we will see in Section 4.3. The next theorem summarizes some of the most important results. For more details, the reader is referred to [Arenas and Libkin 2005].

Theorem 2.3. **[Arenas and Libkin 2005]** *Let Σ be a set of integrity constraints over a database schema S.*

(*1*) *If Σ contains only functional and multivalued dependencies, then (S, Σ) is well-designed if and only if it is in 4NF.*

(*2*) *If Σ contains only functional dependencies, then (S, Σ) is well-designed if and only if it is in BCNF.*

3. QUALITY MEASURE FOR REDUNDANCY OF SCHEMAS

There is always a tradeoff between having a less redundant database and the efficiency of query answering: while doing a good normalization guarantees the least amount of redundancy, it may shred the original relation into too many relations, and this may affect the performance of query answering by requiring many joins. In order to find out if we really want to pay this price, we first need to know how bad an arbitrary schema is in terms of redundancy, and then decide whether a normalization is necessary in case the schema is allowing too much redundancy.

In traditional normalization theory, however, there is a yes or no answer to the question of whether a schema is good in terms of allowing redundant data. In this section, we show that there is a spectrum of redundancy ranging from too redundant to well-designed. In fact, we introduce a quality measure for a given relation schema with functional dependencies that calculates the minimum information content, or the maximum redundancy, for instances of that schema. This number could help a database designer decide whether further normalization is necessary in case database instances of the schema have the potential to carry too much redundancy.

Functional dependencies are one of the most popular integrity constraints that are taken into account in practical database design. However, there are other important integrity constraints, such as inclusion dependencies, that could affect the redundancy of data values. An inclusion dependency (IND), expressed as $R_i[X] \subseteq R_j[Y]$, states that for every database instance I and every tuple $t \in I(R_i)$, there should be a tuple in $t' \in I(R_j)$ such that $t[X] = t'[Y]$. Not only these constraints could introduce new causes of redundancy, but also new functional dependencies could be implied as a result of interacting FDs and INDs. This issue has been studied before, and it is known that interacting FDs and INDs are not desirable in a database design [Levene and Vincent 2000] for two reasons: a database design may be normalized with respect to the set of FDs, but not normalized with respect to the FDs implied by the set of FDs and INDs considered together. Moreover, the joint implication problem for FDs and INDs is undecidable.

Obviously, for a complete comparison of two different designs for a database, each consisting of multiple relations, one needs to consider inclusion dependencies. Our goal is this paper is, however, to study the redundancy introduced by FDs and the redundancy tolerated by FD-based normal forms. We, therefore, assume that schemas do not have inclusion or any other inter-relational dependencies, and then, without loss of generality, we can focus our attention to database schemas with only one relation.

The results of this section and Section 4 are mainly due to a fundamental lemma (Lemma 3.4), which basically shows that the calculation of the information-theoretic measure of redundancy for schemas that only contain functional dependencies reduces to a purely combinatorial analysis. This leads to an interesting observation: while we may need the full power of the information content measure to quantify redundancies caused by different kinds of integrity constraints, we do not need to deal with the complicated information-theoretic definition of this measure when analyzing the redundancy of schemas and normal forms in presense of only FDs, which is what mostly happens in practice.

3.1 Guaranteed information content of schemas

We now introduce a measure called *guaranteed information content* of a schema for a given attribute that shows how redundant the instances of a schema can potentially be for that attribute. This measure finds the information content of the most redundant position for an attribute by looking at the column corresponding to the attribute in all instances of the schema. We want to guarantee a certain amount of information content even for the most redundant instances of a schema, in which there are arbitrarily many distinct tuples showing a redundant fact due to a functional dependency. To be able to produce such instances, we assume that the domain of all attributes is an infinite set, e.g., the set of positive integers $\mathbb{N}^+$.

Definition 3.1. Let R be a relation schema and Σ be a set of functional dependencies defined over the attributes of R. For an attribute $A \in sort(R)$, we define the set of possible values of $\text{RIC}_I(p \mid \Sigma)$ for positions $p = (R, t, A)$ in instances $I \in inst(R, \Sigma)$:

$$\mathcal{POSS}^R_\Sigma(A) = \{\text{RIC}_I(p \mid \Sigma) \ \mid I \text{ is an instance of } (R, \Sigma), \\ p = (R, t, A) \text{ is in } Pos(I)\}.$$

Then the *guaranteed information content of schema* (R, Σ) *for attribute* A, $\mathrm{GIC}^R_\Sigma(A)$, is the infimum $\inf \mathcal{POSS}^R_\Sigma(A)$.

In other words, $\mathrm{GIC}^R_\Sigma(A)$ is the least amount of information content that may be found in A-columns of instances of R that satisfy FDs in Σ, and it can represent the worst-case of redundancy in column A over all possible instances.

The definition of $\mathrm{GIC}^R_\Sigma(A)$ itself does not even suggest that this value is computable. Our goal now is to present a purely combinatorial description of $\mathrm{GIC}^R_\Sigma(A)$ that immediately leads to an algorithm for calculating this value, and study the complexity of the problem of calculating it.

First we note that $\mathrm{GIC}^R_\Sigma(A) = 1$ for every attribute A that is not implied by any non-key set of attributes (i.e., $X \to A$ only if X is a superkey). The value of such an attribute in any instance can be replaced by an arbitrary constant without violating a functional dependency, and hence the information content is always 1 as there is absolutely no redundancy for that attribute.

So we need to show how to calculate $\mathrm{GIC}^R_\Sigma(A)$ when A can be implied by a non-key. It turns out that the structure of minimal non-key sets of attributes that imply A determines this value.

Recall that a *hypergraph* is a pair $\mathcal{H} = (U, \mathcal{F})$, where U is a set and $\mathcal{F}$ is a family of subsets of U. A *hitting set* of $\mathcal{H}$ is a set $V \subseteq U$ such that $V \cap X \neq \emptyset$ for all $X \in \mathcal{F}$. We use the notation $\#\mathrm{HS}(\mathcal{H})$ for the number of hitting sets of $\mathcal{H}$.

The calculation of the value of $\mathrm{GIC}^R_\Sigma(A)$ is based on computing the number of hitting sets of an implication hypergraph of Σ and A, which is a dual concept of a well-studied notion of generating sets for functional dependencies (see, e.g., [Beeri et al. 1984; Mannila and Räihä 1986]; those were primarily studied in connection with constructing Armstrong relations for families of constraints). The notion of 'dual' is the same as duality between keys and antikeys in relational schemas [Demetrovics and Thi 1987].

Definition 3.2. Given a set Σ of FDs over *sort*(R) and an attribute A, the *implication hypergraph of* Σ *and* A is a hypergraph $\mathcal{H} = (U, \mathcal{F})$ where

— $\mathcal{F} = \{X \mid X$ is a minimal non-key subset of *sort*(R) such that $X \to A \in \Sigma^+$ and $A \notin X\}$, and
— $U = \bigcup\{X \mid X \in \mathcal{F}\}$.

We refer to this hypergraph as $\mathrm{Imp}(\Sigma, A)$, or, if Σ is clear from the context, as $\mathrm{Imp}(A)$.

For instance, for schema $R_1(A, B, C, D, E)$ with FDs $\Sigma = \{AB \to E, D \to E\}$, the implication hypergraph of Σ and E is $\mathrm{Imp}(\Sigma, E) = (U, \mathcal{F})$, where $U = \{A, B, D\}$ and $\mathcal{F} = \{AB, D\}$. Moreover, we have $\#\mathrm{HS}(\mathrm{Imp}(\Sigma, E)) = |\{AD, BD, ABD\}| = 3$.

THEOREM 3.3. *Given a set* Σ *of FDs over sort*(R) *and an attribute* A*, let* $\mathrm{Imp}(\Sigma, A) = (U, \mathcal{F})$ *be the implication hypergraph of* Σ *and* A*. Then*

$$\mathrm{GIC}^R_\Sigma(A) \quad = \quad \frac{\#\mathrm{HS}(\mathrm{Imp}(\Sigma, A))}{2^{|U|}}.$$

In other words, $\mathrm{GIC}^R_\Sigma(A)$ is the ratio of hitting sets in the implication hypergraph (which is thus guaranteed to be in the $[0, 1]$ range).

We need a lemma to prove this theorem. Let Σ be a set of FDs over a relation schema R, $I \in \mathit{inst}(R, \Sigma)$, $p \in \mathit{Pos}(I)$. We say that $\bar{a} \in \Omega(I, p)$ *determines* p if there exists $k_0 > 0$ such that for every $k > k_0$, we have $P(a \mid \bar{a}) = 1$ for some $a \in \mathit{adom}(I)$, and $P(b \mid \bar{a}) = 0$ for every $b \in [1, k] - \{a\}$. In other words, $\bar{a}$ determines p if one can specify a single value for p, given the values present in $\bar{a}$ and constraints Σ. We write $\Omega_0(I, p)$ for the set of all $\bar{a} \in \Omega(I, p)$ that determine p, and $\Omega_1(I, p)$ for the set of all $\bar{a} \in \Omega(I, p)$ that do not determine p. Let $n = |\mathit{Pos}(I)|$. Then:

LEMMA 3.4. $\mathrm{RIC}_I(p \mid \Sigma) = |\Omega_1(I, p)|/2^{n-1}$.

PROOF. We show that the value of

$$\lim_{k\to\infty} \frac{1}{\log k} \sum_{a\in[1,k]} P_k(a \mid \bar{a}) \log \frac{1}{P_k(a \mid \bar{a})}$$

is 0 if $\bar{a} \in \Omega_0(I, p)$ and it is 1 if $\bar{a} \in \Omega_1(I, p)$. Assume that $\bar{a}$ determines p. By definition, there is a $k_0 > 0$ such that for every $k > k_0$, it is the case that $P_k(a \mid \bar{a}) = 1$ for some $a \in \mathit{adom}(I)$, and $P_k(b \mid \bar{a}) = 0$ for all $b \in [1, k] - \{a\}$. Hence for all $k > k_0$ we have:

$$\sum_{a\in[1,k]} P_k(a \mid \bar{a}) \log \frac{1}{P_k(a \mid \bar{a})} = 0.$$

Note that $P_k(a \mid \bar{a}) \log \frac{1}{P_k(a|\bar{a})} = 0$ when $P_k(a \mid \bar{a}) = 0$, by definition. Then

$$\lim_{k\to\infty} \frac{1}{\log k} \sum_{a\in[1,k]} P_k(a \mid \bar{a}) \log \frac{1}{P_k(a \mid \bar{a})} = 0.$$

Conversely, suppose $\bar{a}$ does not determine p. Then for every k_0 there is $k > k_0$ such that either $P_k(a \mid \bar{a}) = 0$ for all a, or $P_k(a_1 \mid \bar{a}), P_k(a_2 \mid \bar{a}) > 0$ for at least two different values a_1 and a_2. Since $I \models \Sigma$, we have $|\mathrm{SAT}^k_\Sigma(I_{(a,\bar{a})})| > 0$ for at least one $a \in \mathit{adom}(I)$, ruling out the first possibility. Since Σ contains only FDs, we conclude that $|\mathrm{SAT}^k_\Sigma(I_{(b,\bar{a})})| = |\mathrm{SAT}^k_\Sigma(I_{(b',\bar{a})})| > 0$ for all $b, b' \notin \mathit{adom}(I)$. Hence $P_k(b \mid \bar{a}) \leq 1/(k - n)$.

Next, expand $\bar{a}$ to $\bar{a}'$ by putting in a value for every position that is determined by $\bar{a}$ (which excludes p). Let r be the number of variables in $\bar{a}'$. Then for each $c \in [1, k]$ we have $|\mathrm{SAT}^k_\Sigma(I_{(c,\bar{a})})| \leq k^r$. Furthermore, for each $b \notin \mathit{adom}(I)$, any substitution σ that assigns to the r variables different values in $[1, k] - (\mathit{adom}(I) \cup \{b\})$ will be in $\mathrm{SAT}^k_\Sigma(I_{(b,\bar{a})})$; hence, we have $|\mathrm{SAT}^k_\Sigma(I_{(b,\bar{a})})| \geq (k - n - r)^r$. We thus have

$$P_k(b \mid \bar{a}) \geq \frac{(k - n - r)^r}{k \cdot k^r} = \frac{1}{k}\big(1 - \frac{n + r}{k}\big)^r.$$

Let $\pi_i = P_k(a_i \mid \bar{a})$ for each $a_i \in \mathit{adom}(I)$. Then

$$\begin{aligned}&\tfrac{1}{\log k} \textstyle\sum_{a\in[1,k]} P_k(a \mid \bar{a}) \log \frac{1}{P_k(a|\bar{a})} \geq \\ &\tfrac{1}{\log k}\Big(\sum_{a_i\in \mathit{adom}(I)} \pi_i \log \tfrac{1}{\pi_i} + (k - n) \cdot \tfrac{\log(k-n)}{k} \cdot \big(1 - \tfrac{n+r}{k}\big)^r\Big).\end{aligned}$$

Since n and r are fixed, this implies that $\lim_{k\to\infty} \frac{1}{\log k} \sum_{a\in[1,k]} P_k(a \mid \bar{a}) \log \frac{1}{P_k(a|\bar{a})} \neq 0$. By a known result [Arenas and Libkin 2005], this limit always exists, and if it is not 0, then it must be equal to 1, so we have

$$\begin{aligned}\textsc{Ric}_I(p \mid \Sigma) &= \lim_{k\to\infty} \tfrac{1}{\log k} \sum_{\bar{a}\in\Omega(I,p)} \tfrac{1}{2^{n-1}} \sum_{a\in[1,k]} P(a \mid \bar{a}) \log \tfrac{1}{P(a|\bar{a})} \\ &= \tfrac{1}{2^{n-1}} \sum_{\bar{a}\in\Omega_1(I,p)} \lim_{k\to\infty} \tfrac{1}{\log k} \sum_{a\in[1,k]} P(a \mid \bar{a}) \log \tfrac{1}{P(a|\bar{a})} \\ &= |\Omega_1(I,p)|/2^{n-1},\end{aligned}$$

which concludes the proof of Lemma 3.4. □

Proof of Theorem 3.3. Let $\text{Imp}(A) = (U, \mathcal{F})$ be the implication hypergraph of A, $X_1, \ldots, X_k$ denote the sets in $\mathcal{F}$, and S contain all the hitting sets of the family $\mathcal{F}$, i.e.,

$$S = \{Y \mid Y \subseteq X_1 \cup \ldots \cup X_k,\ Y \cap X_i \neq \emptyset \text{ for all } i \in [1,k]\}.$$

We denote the cardinality of the universe of $\text{Imp}(A)$ (that is, $\bigcup_i X_i$) by l. Then the proof consists of two parts. We prove that:

(a). for every $\varepsilon > 0$, there exists an instance $I \in \mathit{inst}(R,\Sigma)$ and a position $p = (R,t,A)$ in $\mathit{Pos}(I)$, such that $\textsc{Ric}_I(p \mid \Sigma) < |S| \cdot 2^{-l} + \varepsilon$;

(b). for every instance $I \in \mathit{inst}(R,\Sigma)$ and every position $p = (R,t,A)$ in $\mathit{Pos}(I)$, we have $\textsc{Ric}_I(p \mid \Sigma) \geq |S| \cdot 2^{-l}$.

(a) We construct instance I of R consisting of tuples $t_0, \ldots, t_q$, where $q = kr$, as follows:

—for all attributes $B \in \mathit{sort}(R)$, $t_0[B] = 1$;

—for a tuple t_i, $i \in [1,q]$, $t_i[B] = 1$ for all $B \in X_j^+$, where $j = \lceil i/r \rceil$, and for all other attributes $C \notin X_j^+$, $t_i[C] = v$, where v is a fresh value not used so far.

This instance consists of k groups, each having r tuples that agree with each other and also with tuple t_0 only on one of the sets of attributes X_j^+. Now consider position $p = (R, t_0, A)$.

Claim 3.5. *For the information content of position p we have*

$$\textsc{Ric}_I(p \mid \Sigma) \leq |S| \cdot 2^{-l} + \frac{2^{-l} - |S|}{2^l} \cdot (\frac{2^m - 1}{2^m})^r,$$

where $m = |\mathit{sort}(R)|$.

Proof. Let $\bar{a}$ be an arbitrary vector in $\Omega(I,p)$. Let $\bar{a}_{[t_0]}$ denote the subtuple in $\bar{a}$ corresponding to tuple $t_0 \in I$. If for all attributes in one of the sets Y in S, $\bar{a}_{[t_0]}$ contains variables, then $\bar{a}$ does not determine p no matter what the other positions in $\bar{a}$ contain. This is because none of the FDs implying attribute A could enforce a value for position p since Y contains at least one element from each X_j, $j \in [1,k]$. There are $|S| \cdot 2^{n-l-1}$ of such $\bar{a}$'s.

If for all attributes of some X_j, $j \in [1,k]$, subtuple $\bar{a}_{[t_0]}$ contains constants, which can happen for $(2^l - |S|) \cdot 2^{m-l-1}$ subtuples, then $\bar{a}$ does not determine p only if for any subtuple $\bar{a}_{[t_i]}$ corresponding to a tuple t_i, $i \in ((j-1)r, jr]$, $\bar{a}_{[t_i]}$ does not contain a constant for at least one attribute in X_jA. Therefore, $\bar{a}_{[t_i]}$ can have at most $2^m - 1$ shapes. The other $q - r$ subtuples in $\bar{a}$ can have at most 2^m shapes.

Putting everything together, $|\Omega_1(I,p)|$, or the total number of different $\bar{a}$'s in $\Omega(I,p)$ that do not determine p is at most

$$|S| \cdot 2^{n-l-1} + (2^l - |S|) \cdot (2^m - 1)^r \cdot (2^m)^{q-r} \cdot 2^{m-l-1},$$

and $n = m(q+1)$. Then by Lemma 3.4,

$$\begin{aligned} \textsc{Ric}_I(p \mid \Sigma) &\leq \frac{|S| \cdot 2^{n-l-1}}{2^{n-1}} + \frac{(2^l - |S|) \cdot (2^m - 1)^r \cdot (2^m)^{q-r} \cdot 2^{m-l-1}}{2^{n-1}} \\ &= |S| \cdot 2^{-l} + \frac{2^l - |S|}{2^l} \cdot (\frac{2^m - 1}{2^m})^r, \end{aligned}$$

which proves the claim. By taking $r > \log_{\frac{2^m}{2^m-1}}(\frac{2^l}{2^l-|S|})(\frac{1}{\varepsilon})$ we will have $\textsc{Ric}_I(p \mid \Sigma) < |S| \cdot 2^{-l} + \varepsilon$.

(b) Let I be an arbitrary instance in $inst(R,\Sigma)$, $p = (R, t_0, A) \in Pos(I)$, and $\bar{a} \in \Omega(I,p)$. Let $\bar{a}_{[t_0]}$ denote the subtuple of $\bar{a}$ corresponding to t_0. If for all attributes in one of the sets Y in S, $\bar{a}_{[t_0]}$ contains variables, then $\bar{a}$ does not determine p no matter what the other positions in $\bar{a}$ contain. There are $|S| \cdot 2^{n-l-1}$ of such $\bar{a}$'s. Therefore, $|\Omega_1(I,p)|$ is at least $|S| \cdot 2^{n-l-1}$, and thus $\textsc{Ric}_I(p \mid \Sigma) \geq |S| \cdot 2^{-l}$, which completes the proof of Theorem 3.3. □

The following example shows how the measure can be applied to a relation schema with functional dependency to determine the worst-case redundancy of the instances.

Example 3.6. Consider a schema $R_1(A,B,C,D,E)$ with FDs:

$$\begin{aligned} \Sigma_1 = \{\ & AB \rightarrow E, \\ & D \rightarrow E \quad \} \end{aligned}$$

and a schema $R_2(A,B,C,D,E)$ with FDs:

$$\begin{aligned} \Sigma_2 = \{\ & A \rightarrow B, \\ & AC \rightarrow E, \\ & BD \rightarrow E \ \} \end{aligned}$$

Just by looking at these two sets of FDs, it may not be immediately obvious which one of the designs (R_1, Σ_1) or (R_2, Σ_2) allows more redundancy in column E. Now using our combinatorial criterion, we can calculate and compare the values of $\text{GIC}^{R_1}_{\Sigma_1}(E)$ and $\text{GIC}^{R_2}_{\Sigma_2}(E)$.

First we use Theorem 3.3 to calculate the minimum information content, or the maximum redundancy, allowed in column E of instances of R_1 satisfying Σ_1:

$$\begin{aligned} \text{Imp}(\Sigma_1, E) &= (\{A,B,D\}, \{AB, D\}) \\ \#\text{HS}(\text{Imp}(\Sigma_1, E)) &= |\{AD, BD, ABD\}| = 3 \\ \text{GIC}^{R_1}_{\Sigma_1}(E) &= 3 \cdot 2^{-3} = \tfrac{3}{8} \end{aligned}$$

Now similar calculations for Σ_2 show:

$$\begin{aligned} \text{Imp}(\Sigma_2, E) &= (\{A,B,C,D\}, \{AC, BD, AD\}) \\ \#\text{HS}(\text{Imp}(\Sigma_2, E)) &= |\{AD, AB, CD, ABC, ABD, ACD, BCD, ABCD\}| = 8 \\ \text{GIC}^{R_2}_{\Sigma_2}(E) &= 8 \cdot 2^{-4} = \tfrac{1}{2} \end{aligned}$$

Hence, the first schema has a higher potential for storing redundant values in column E since it guarantees a lower information content for this attribute. □

3.2 Complexity of computing guaranteed information content

The following result shows that computing the worst-case redundancy for a schema with functional dependencies is computationally expensive. However, the complexity is with respect to the size of the schema and FDs, and has nothing to do with the size of database instances. Therefore, the guaranteed information content measure can still be useful in analyzing redundancy of schemas if we are dealing with schemas with reasonable sizes.

THEOREM 3.7. *Computing guaranteed information content of a schema is #P-hard.*

PROOF. We reduce from the #P-complete problem #MONOTONE 2-SAT [Valiant 1979], which is the problem of finding the number of satisfying assignments of a CNF formula, in which every clause is a disjunction of two positive literals. Let $C = C_1 \wedge \ldots \wedge C_k$ be such a CNF formula, where for every C_i, $i \in [1, k]$, $C_i = x_{i1} \vee x_{i2}$ for two variables $x_{i1}, x_{i2} \in X$ (we assume that every variable in X appears in at least one clause). We create relation schema R with attributes $sort(R) = X \cup \{y, z\}$ and a set of FDs Σ that contains an FD of the form $x_{i1}x_{i2} \rightarrow y$ for every clause $C_i = x_{i1} \vee x_{i2}$ in C. Guaranteed information content of schema (R, Σ) for attribute y, $\mathrm{GIC}^R_\Sigma(y)$, can be computed by finding $\#\mathrm{HS}(\mathrm{Imp}(y))$, which is exactly the number of satisfying assignments of C. More precisely, $\mathrm{GIC}^R_\Sigma(y) = N \cdot 2^{-|X|}$ if and only if the number of satisfying assignments of C is N. □

4. QUALITY MEASURE FOR REDUNDANCY OF NORMAL FORMS

The information content measure was recently used [Arenas and Libkin 2005] to justify *perfect* normal forms like BCNF that completely eliminate redundancies. In practice, however, one usually settles for 3NF which, unlike BCNF, may not perfectly eliminate all redundancies but always guarantees dependency preservation.

Our goal in this section is to provide an information-theoretic measure that evaluates non-perfect normal forms: the ones that tolerate some redundancy. Our measure, called guaranteed information content of normal forms, is based on the highest amount of redundancy or, equivalently, the lowest amount of information content that normal forms allow in a single position of a database as well as in the entire database on average.

The main result of this section provides a formal justification for one of the most popular and commonly-used normal forms, 3NF. The main property possessed by 3NF, but not by BCNF, is *dependency preservation*: for every schema, there always exists a lossless decomposition into 3NF that preserves all the functional dependencies. To guarantee dependency preservation, 3NF has to pay a price by tolerating some redundancy. We will compute the minimum price, in terms of redundancy, that is needed for a normal form to guarantee dependency preservation and show that a "good" 3NF normalization achieves this minimum redundancy.

4.1 Guaranteed information content of normal forms

To be able to compare normal forms with respect to the lowest amount of information content that they allow, we again shall use the notion of guaranteed information content (see Definition 3.1). This time, it needs to be specified for a condition $\mathcal{C}$ (i.e., it is the smallest number $g \in [0,1]$ that can be found as the information content of some position in instances that satisfy $\mathcal{C}$). Furthermore, as we are using this notion to *compare* different conditions, we do not want the minimum over all instances satisfying $\mathcal{C}$, but rather instances with the same number of attributes. As explained earlier, even for conditions such as 3NF, the values of $\textsc{Ric}_I(p \mid \Sigma)$ can be arbitrarily low [Kolahi 2007], but the cause of this is relations with many attributes. Hence we do not want to compare values of $\textsc{Ric}_I(p \mid \Sigma)$ in relations over different sets of attributes. Thus, we shall use a modified definition of guaranteed information content, specialized to instances of m-attribute relations that satisfy $\mathcal{C}$.

Definition 4.1. Let $\mathcal{C}$ be a condition on relation schemas with functional dependencies. We define the set of possible values of $\textsc{Ric}_I(p \mid \Sigma)$ for m-attribute instances I of schemas satisfying $\mathcal{C}$:

$$\mathcal{POSS}_{\mathcal{C}}(m) = \{\textsc{Ric}_I(p \mid \Sigma) \ \begin{array}{l} | I \text{ is an instance of } (R,\Sigma), \\ R \text{ has } m \text{ attributes}, \\ (R,\Sigma) \text{ satisfies } \mathcal{C}\}. \end{array}$$

Then the *guaranteed information content*, $\text{GIC}_{\mathcal{C}}(m)$, is $\inf \mathcal{POSS}_{\mathcal{C}}(m)$.

For instance, we know that BCNF corresponds exactly to maximum information content for all positions in all instances [Arenas and Libkin 2005]. We can now formulate this fact as $\text{GIC}_{\text{BCNF}}(m) = 1$, for all $m > 0$.

Guaranteed information content of a normal form measures the redundancy of the most redundant position across all instances of schemas that satisfy the normal form. Next, we introduce another measure that looks for instances that have the highest average redundancy over all instances of schemas satisfying a normal form.

4.2 Guaranteed average information content of normal forms

To compare normal forms with respect to the lowest average information content that they allow for an instance, we define a measure called *guaranteed average information content* for a condition $\mathcal{C}$ as the smallest number $g \in [0,1]$ that can be found as the average information content of some instance that satisfies $\mathcal{C}$. We are particularly interested in the lowest average information content of instances of m-attribute relations that satisfy $\mathcal{C}$.

Definition 4.2. Let $\mathcal{C}$ be a condition on relation schemas with functional dependencies. Let $\text{AVG}(I \mid \Sigma)$ denote the average of the numbers in $\{\textsc{Ric}_I(p \mid \Sigma) \mid p \in Pos(I)\}$. We define the set of possible values of $\text{AVG}(I \mid \Sigma)$ for m-attribute instances I of schemas satisfying $\mathcal{C}$:

$$\mathcal{POSSA}_{\mathcal{C}}(m) = \{\text{AVG}(I \mid \Sigma) \ \begin{array}{l} | I \text{ is an instance of } (R,\Sigma), \\ R \text{ has } m \text{ attributes}, \\ (R,\Sigma) \text{ satisfies } \mathcal{C}\}. \end{array}$$

Then the *guaranteed average information content*, $\text{GAVG}_{\mathcal{C}}(m)$, is $\inf \mathcal{POSSA}_{\mathcal{C}}(m)$.

Again, we can clearly say that $\mathrm{GAVG}_{\mathrm{BCNF}}(m) = 1$, for all $m > 0$.

Next we will use the measures that we introduced to evaluate normal forms by the amount of redundancy they tolerate.

4.3 Price of dependency preservation: justifying 3NF

We say that a normal form $\mathcal{NF}$ is *dependency-preserving* if every relation schema admits a dependency-preserving $\mathcal{NF}$-decomposition. That is, for every relation schema (R, Σ), where Σ is a set of FDs, there is a lossless decomposition of (R, Σ) into schemas $(R_1, \Sigma_1), \ldots, (R_\ell, \Sigma_\ell)$, $\ell \geq 1$, such that each (R_i, Σ_i) satisfies $\mathcal{NF}$ and $\left(\bigcup_{i=1}^{l} \Sigma_i\right)^+ = \Sigma^+$.

For a dependency-preserving normal form $\mathcal{NF}$, we look at the set $\mathcal{G}(\mathcal{NF})$ of values $c \in [0,1]$ such that for an arbitrary schema we can always guarantee an $\mathcal{NF}$-decomposition in which the information content in *all* positions is at least c. Again, to be able to produce highly-redundant instances, we assume that the domain of all attributes is an infinite set, e.g., the set of positive integers $\mathbb{N}^+$. Formally, $\mathcal{G}(\mathcal{NF})$ is the set

$$\begin{aligned}\{c \in [0,1] \quad | \; & \forall\ (R, \Sigma),\ \forall\ I \in inst(R, \Sigma), \\ & \exists\ \mathcal{NF}\text{-decomposition } \{(R_j, \Sigma_j)\}_{j=1}^{\ell} \text{ s.t.} \\ & \forall j \leq \ell\ \forall p \in Pos(I_j),\ \textsc{Ric}_{I_j}(p \mid \Sigma_j) \geq c\},\end{aligned}$$

where I_j refers to $\pi_{sort(R_j)}(I)$. Using this, we define the price of dependency preservation for $\mathcal{NF}$ as the smallest amount of information content that is necessarily lost due to redundancies: that is, the smallest amount of redundancy one has to tolerate in order to have dependency preservation.

Definition 4.3. For every dependency-preserving normal form $\mathcal{NF}$, the price of dependency preservation $\textsc{Price}(\mathcal{NF})$ is defined as $1 - \sup \mathcal{G}(\mathcal{NF})$.

Clearly, $\textsc{Price}(\mathcal{NF}) \leq 1$. Since the FD-based normal form that achieves the maximum value 1 of $\textsc{Ric}_I(p \mid \Sigma)$ in all relations is BCNF [Arenas and Libkin 2005], and BCNF does *not* ensure dependency preservation, $\textsc{Price}(\mathcal{NF}) > 0$ for any dependency-preserving normal form $\mathcal{NF}$.

Now we are ready to present the main result of this section. Intuitively, it shows that each normal form needs to pay at least half of the maximum redundancy to achieve dependency preservation, and this is exactly what 3NF pays.

THEOREM 4.4. $\textsc{Price}(\text{3NF}) = 1/2$. *Furthermore, if $\mathcal{NF}$ is a dependency-preserving normal form, then* $\textsc{Price}(\mathcal{NF}) \geq 1/2$.

In the rest of this section we prove this theorem. We say that a schema (R, Σ) is *indecomposable* if it has no lossless dependency-preserving decomposition into smaller relations. That is, there is no $\mathcal{NF}$-decomposition $\{(R_1, \Sigma_1), \ldots, (R_\ell, \Sigma_\ell)\}$, with $|sort(R_j)| < |sort(R)|$ for every $j \in [1, \ell]$, that is both lossless and dependency-preserving. We are only interested in indecomposable schemas that are not in BCNF since BCNF already guarantees zero redundancy. The proof of Theorem 4.4 relies on two properties of indecomposable schemas presented in propositions below. We say that a candidate key X is *elementary* [Zaniolo 1982] if there is an attribute $A \notin X$ such that $X' \rightarrow A \notin \Sigma^+$ for all $X' \subsetneq X$.

```
Input: A relation schema $(R[U], \Sigma)$, where $\Sigma$ is a set of functional dependencies.
Output: A database schema $S$ in 3NF.
find a minimal cover $\Sigma'$ of $\Sigma$;
for each $X \to A$ in $\Sigma'$ do
    include the relation schema $(R_i[XA], \{X \to A\})$ in the output schema:
    $S := S \cup (R_i[XA], \{X \to A\})$, where $R_i$ is a fresh relation name;
if there is no $(R_i[U_i], \Sigma'_i)$ such that $U_i$ is a superkey for $R[U]$ then
    choose a key $X$ of $R[U]$;
    $S := S \cup \{(R_j[X], \emptyset)\}$;
return $S$.
```

Fig. 2. An algorithm for synthesizing 3NF schemas.

Proposition 4.5. *Let R be a relation schema with attributes $\{A_1, \ldots, A_m\}$, and let Σ be a non-empty set of FDs over R. Then (R, Σ) is indecomposable if and only if it has an $(m-1)$-attribute elementary key.*

Proof. If (R, Σ) contains an $(m-1)$-attribute elementary candidate key, then every decomposition of it would lose this key; hence, it is indecomposable. Conversely, suppose (R, Σ) is indecomposable, and there is no elementary candidate key with $m-1$ attributes. Let Σ_c be an arbitrary minimal cover for Σ. Then for every FD $X \to A \in \Sigma_c$, we have $X \cup \{A\} \subsetneq sort(R)$. Hence, the standard 3NF synthesis algorithm (see [Abiteboul et al. 1995]) shown in Fig. 2 will produce a dependency-preserving decomposition of (R, Σ), and this contradicts the assumption that (R, Σ) is indecomposable. □

Notice that the schemas produced by the synthesis algorithm in Fig. 2 (taken from [Abiteboul et al. 1995]) are indecomposable. The only difference between this algorithm and the 3NF synthesis algorithm, originally proposed by Bernstein [Bernstein 1976] and later extended [Biskup et al. 1979] to ensure the lossless decomposition property, is that this one does not group the functional dependencies according to their left-hand sides before synthesizing, and therefore produces smaller schemas, which are indecomposable.

Let $\mathcal{ID}$ denote the property of being indecomposable. Recall that $\text{GIC}_{\mathcal{ID}}(m)$ is the infimum of the set $\mathcal{POSS}_{\mathcal{ID}}(m)$ of possible values of $\text{Ric}_I(p \mid \Sigma)$ for m-attribute instances of indecomposable schemas (R, Σ). The following proposition shows that the information content in instances of indecomposable schemas can go arbitrarily close to 1/2 but not less than that.

Proposition 4.6. $\text{GIC}_{\mathcal{ID}}(m) = 1/2$ *for all* $m > 2$.

Proof. The proof consists of two parts. We prove that:

(a). for every $m > 2$ and $\varepsilon > 0$, there exists a schema (R, Σ), an instance $I \in inst(R, \Sigma)$, and a position $p \in Pos(I)$, such that $|sort(R)| = m$, (R, Σ) is indecomposable, and $\text{Ric}_I(p \mid \Sigma) < 1/2 + \varepsilon$;

(b). for every indecomposable schema (R, Σ), every instance $I \in inst(R, \Sigma)$, and every position $p \in Pos(I)$, we have $\text{Ric}_I(p \mid \Sigma) \geq 1/2$.

(a) Consider the relation schema $R(A_1, \ldots, A_m)$ with FDs $\Sigma = \{A_1 A_2 \ldots A_{m-1} \to A_m,\ \ A_m \to A_1\}$ and the instance I of this schema shown in Fig. 3. By Proposition 4.5, (R, Σ) is indecomposable. Let t_0 denote the first tuple depicted in Fig. 3, and let p denote the position of the gray cell.

A_1	A_2	A_3	$\ldots$	A_m
1	1	1	$\ldots$	1
1	2	1	$\ldots$	1
1	3	1	$\ldots$	1
$\vdots$	$\vdots$	$\vdots$		$\vdots$
1	k	1	$\ldots$	1

Fig. 3. A database instance for the proofs of Propositions 4.6 and 4.10.

Claim 4.7. *The information content of position p is*

$$\textsc{Ric}_I(p \mid \Sigma) = \frac{1}{2} + \frac{1}{2}\Big(\frac{3}{4}\Big)^{k-1}.$$

Proof. Let $\bar{a}$ be an arbitrary vector in $\Omega(I, p)$. Let $\bar{a}_{[t_0]}$ denote the subtuple in $\bar{a}$ corresponding to tuple $t_0 \in I$ and $\bar{a}_{[t_1]}$ denote the subtuple in $\bar{a}$ corresponding to an arbitrary tuple $t_1 \in I$. Each position in these subtuples contains either a variable (representing a missing value) or a constant, which equals the value that I has for that position.

Then $\bar{a}$ does not determine p if and only if

(1) the subtuple $\bar{a}_{[t_0]}$ has a variable in the position corresponding to attribute A_m; or

(2) the subtuple $\bar{a}_{[t_0]}$ has a constant in the position corresponding to attribute A_m, and for an arbitrary subtuple $\bar{a}_{[t_1]}$ in $\bar{a}$, $t_1 \neq t_0$:
  2.1. the subtuple $\bar{a}_{[t_1]}$ has a variable in the position corresponding to attribute A_m; or
  2.2. the subtuple $\bar{a}_{[t_0]}$ has a constant in the position corresponding to attribute A_m but a variable in the position corresponding to attribute A_1.

In Case 1, $\bar{a}$ can have either a variable or a constant in all other $n - 2$ positions. Therefore, we can have 2^{n-2} such $\bar{a}$'s. In Case 2, $\bar{a}_{[t_0]}$ can have either a constant or a variable in the positions corresponding to $A_2, \ldots, A_{m-1}$. Furthermore, in Case 2.1, every such subtuple $\bar{a}_{[t_1]}$ can have either a constant or a variable in the positions corresponding to attributes $A_1, \ldots, A_{m-1}$, and in Case 2.2, it can have either a constant or a variable in the positions corresponding to $A_2, \ldots, A_{m-1}$. Therefore, the total number of $\bar{a}$'s satisfying conditions of Case 2 is $2^{m-2}(2^{m-1} + 2^{m-2})^{k-1}$ since we have $k - 1$ tuples other than t_0 in the instance.

Then $|\Omega_1(I, p)|$, or the total number of different $\bar{a}$'s in $\Omega(I, p)$ that do not determine p is

$$2^{n-2} + 2^{m-2}(2^{m-1} + 2^{m-2})^{k-1}.$$

By Lemma 3.4, $\textsc{Ric}_I(p \mid \Sigma)$ can be obtained by dividing this number by $2^{n-1} = 2^{mk-1}$:

$$\begin{aligned} \textsc{Ric}_I(p \mid \Sigma) &= \frac{2^{mk-2} + 2^{m-2}(2^{m-1} + 2^{m-2})^{k-1}}{2^{mk-1}} \\ &= \frac{1}{2} + \frac{1}{2}\Big(\frac{3}{4}\Big)^{k-1}, \end{aligned}$$

which proves the claim.

Thus for any $\varepsilon > 0$, there is an instance of the form shown in Fig. 3 and a position p in it such that the information content of p is less than $1/2 + \varepsilon$: one needs to choose $k > 1 + \log_{4/3}(1/(2\varepsilon))$ and apply Claim 4.7.

(b) We need an easy observation (that will also be used in the proofs of the next section). For a key X, an attribute $A \notin X$ such that A does not occur in the right-hand side of any nontrivial FD, we have $\text{RIC}_I(p \mid \Sigma) = 1$ for any instance I of (R, Σ) and any position p corresponding to attribute A. Indeed, in this case $|\text{SAT}^k_\Sigma(I_{(a,\bar{a})})| = |\text{SAT}^k_\Sigma(I_{(b,\bar{a})})|$ for arbitrary $a, b \in [1, k]$ and hence $P(a \mid \bar{a}) = 1/k$, and thus $\text{RIC}^k_I(p \mid \Sigma) = \log k$, and $\text{RIC}_I(p \mid \Sigma) = \lim_{k\to\infty} \text{RIC}^k_I(p \mid \Sigma)/\log k = 1$.

Now let Σ be an arbitrary non-empty set of FDs over $R(A_1, \ldots, A_m)$ such that (R, Σ) is indecomposable, and $A_1, \ldots, A_{m-1} \to A_m \in \Sigma$ be the FD of the form described in Proposition 4.5: that is, $A_1 \ldots A_{m-1}$ is an elementary candidate key. For any instance I of (R, Σ) and any position $p = (R, t, A_m) \in \mathit{Pos}(I)$ corresponding to attribute A_m, we have $\text{RIC}_I(p \mid \Sigma) = 1$ since p cannot have any redundancy due to a non-key FD.

Let $I \in \mathit{inst}(R, \Sigma)$, $p = (R, t_0, A_i) \in \mathit{Pos}(I)$, for some $i \in [1, m-1]$, and $\bar{a} \in \Omega(I, p)$. Let $\bar{a}_{[t_0]}$ denote the subtuple of $\bar{a}$ corresponding to t_0. It is easy to see that if $\bar{a}_{[t_0]}$ has a variable in the position corresponding to attribute A_m, then $\bar{a}$ does not determine p, no matter what the other positions in $\bar{a}$ contain. This is because there is no nontrivial FD $X \to A_i \in \Sigma^+$ such that $X \subseteq \{A_2, \ldots, A_{m-1}\}$. All other $n - 2$ positions in $\bar{a}$ can therefore contain either a constant or a variable, so there are at least 2^{n-2} $\bar{a}$'s that do not determine p. Then using Lemma 3.4, we conclude that the information content of p is at least $\frac{2^{n-2}}{2^{n-1}} = 1/2$. This proves Proposition 4.6. □

PROOF OF THEOREM 4.4. The second part of the proof follows from Proposition 4.6: the information content of a position in an indecomposable instance can be arbitrarily close to 1/2. Therefore, for every dependency-preserving normal form $\mathcal{NF}$ (which cannot further decompose an indecomposable instance), $\sup \mathcal{G}(\mathcal{NF})$ cannot exceed 1/2. Therefore, $\text{PRICE}(\mathcal{NF}) \geq 1/2$.

To prove the first part, we notice that, by Proposition 4.5 and basic properties of 3NF, every indecomposable (R, Σ) is in 3NF. Furthermore, if (R, Σ) is decomposable, then the 3NF synthesis algorithm of Fig. 2 will decompose (R, Σ) into indecomposable schemas. Therefore, for every (R, Σ) and every $I \in \mathit{inst}(R, \Sigma)$, one can find a 3NF-decomposition in which the information content of every position is at least 1/2 and sometimes exactly 1/2. That is, $\sup \mathcal{G}(\text{3NF}) = 1/2$, and $\text{PRICE}(\text{3NF}) = 1/2$. This concludes the proof. □

Notice that the proof of Theorem 4.4 implies that the guaranteed information content 1/2 (which witnesses $\text{PRICE}(\text{3NF}) = 1/2$) occurs in decompositions produced by the standard synthesis algorithm [Abiteboul et al. 1995], shown in Fig. 2, that generates a 3NF design from a minimal cover for Σ. Hence, our result not only justifies 3NF as the best dependency-preserving normal form, but also shows which 3NF decomposition algorithm guarantees the highest information content.

4.4 Comparing normal forms

In Section 4.3, we calculated the price of dependency preservation for normal forms and proved that one can always guarantee a 3NF decomposition whose price would be less than or equal to the price of other normal form decompositions. This good price can be achieved for schemas produced by the standard 3NF synthesis algorithm, as shown in Fig. 2. However, not every 3NF normalization would be of the same quality in terms of redundancy. It was noticed long ago that 3NF normalization algorithms can differ significantly [Ling et al. 1981; Zaniolo 1982; Biskup and Meyer 1987] in other aspects, such as the size of schemas they produce or the ability to remove redundant attributes. In this section, we use the information-theoretic framework to compare different dependency-preserving normal forms in terms of the amount of redundancy they allow in data.

The measure for this comparison is the *gain of normalization* function defined as

$$\textsc{Gain}_{\mathcal{NF}_1/\mathcal{NF}_2}(m) \;=\; \frac{\mathrm{GIC}_{\mathcal{NF}_1}(m)}{\mathrm{GIC}_{\mathcal{NF}_2}(m)},$$

where $\mathrm{GIC}_{\mathcal{NF}_1}(m), \mathrm{GIC}_{\mathcal{NF}_2}(m)$ are the smallest value of $\textsc{Ric}_I(p \mid \Sigma)$, as (R, Σ) ranges over schemas with m attributes satisfying normal forms $\mathcal{NF}_1$ and $\mathcal{NF}_1$ respectively (see Definition 4.1) .

We will substitute parameters $\mathcal{NF}_1$ or $\mathcal{NF}_2$ with All representing all schemas with no particular constraint, 3NF representing all schemas that satisfy the general definition of third normal form, and $3\mathrm{NF}^+$ representing the indecomposable schemas generated by the synthesis algorithm shown in Fig. 2.

We now prove that any 3NF schema, not necessarily indecomposable, is at least twice as good as some unnormalized schema. More precisely, the gain function for 3NF is the constant 2 for all $m > 2$ (the case of $m \leq 2$ is special, as any nontrivial FD over two attributes is a key, and hence all schemas are in BCNF). We also show that $3\mathrm{NF}^+$ schemas could be significantly better than arbitrary 3NF schemas. That is,

Theorem 4.8. *For every $m > 2$:*

— $\textsc{Gain}_{3\mathrm{NF}/\mathsf{All}}(m) = 2$;
— $\textsc{Gain}_{3\mathrm{NF}^+/3\mathrm{NF}}(m) = 2^{m-3}$;
— $\textsc{Gain}_{3\mathrm{NF}^+/\mathsf{All}}(m) = 2^{m-2}$.

In the proof of Theorem 4.4, we showed that $\mathrm{GIC}_{3\mathrm{NF}^+}(m) = \mathrm{GIC}_{\mathcal{ID}}(m) = 1/2$. Hence, the result will follow from these two propositions.

Proposition 4.9. $\mathrm{GIC}_{\mathsf{All}}(m) = 2^{1-m}$ *for all $m > 2$.*

Proposition 4.10. $\mathrm{GIC}_{3\mathrm{NF}}(m) = 2^{2-m}$ *for all $m > 2$.*

Proof of Proposition 4.9. We need to show that:

(a). for every $m > 2$ and $\varepsilon > 0$, there exists a schema (R, Σ) with $|sort(R)| = m$, an instance $I \in inst(R, \Sigma)$, and a position $p \in Pos(I)$ such that $\textsc{Ric}_I(p \mid \Sigma) < 2^{1-m} + \varepsilon$;

(b). for every (R, Σ) with $|sort(R)| = m$, every instance $I \in inst(R, \Sigma)$, and every position $p \in Pos(I)$, we have $\textsc{Ric}_I(p \mid \Sigma) \geq 2^{1-m}$.

A_1	A_2	A_3	$\ldots$	A_m
1	1	1	$\ldots$	1
1	2	1	$\ldots$	1
1	1	2	$\ldots$	1
⋮	⋮	⋮		⋮
1	1	1	$\ldots$	2
1	3	1	$\ldots$	1
1	1	3	$\ldots$	1
⋮	⋮	⋮		⋮
1	1	1	$\ldots$	3
⋮	⋮	⋮		⋮
1	k	1	$\ldots$	1
1	1	k	$\ldots$	1
⋮	⋮	⋮		⋮
1	1	1	$\ldots$	k

Fig. 4. A database instance for the proof of Proposition 4.9.

(a) Consider $R(A_1, \ldots, A_m)$ and $\Sigma = \{A_2 \rightarrow A_1, A_3 \rightarrow A_1, \ldots, A_m \rightarrow A_1\}$. Consider the instance $I \in \mathit{inst}(R, \Sigma)$ shown in Fig. 4. Let t_0 denote the first tuple depicted in this figure, and $p = (R, t_0, A_1)$ denote the position of the gray cell. Let t be the number of tuples minus 1, that is, $(m-1)(k-1)$.

Claim 4.11. *The information content of position p is*

$$\textsc{Ric}_I(p \mid \Sigma) = \frac{1}{2^{m+t-1}} \sum_{i=0}^{m-1} \binom{m-1}{i} (1 + 2^{-i})^t.$$

Proof. Let $\bar{a}$ be an arbitrary vector in $\Omega(I, p)$. Let $\bar{a}_{[t_0]}$ denote the subtuple of $\bar{a}$ corresponding to t_0, and suppose $\bar{a}_{[t_0]}$ has constants in positions corresponding to i attributes, and it has variables in the positions corresponding to the remaining $m - 1 - i$ attributes. Then $\bar{a}$ does not determine p if and only if for any arbitrary subtuple $\bar{a}_{[t_1]}$ of $\bar{a}$ corresponding to a tuple $t_1 \in I$, $t_1 \neq t_0$, we have:

(1) the subtuple $\bar{a}_{[t_1]}$ has a variable in the position corresponding to A_1; or

(2) the subtuple $\bar{a}_{[t_1]}$ has a constant in the position corresponding to A_1 but variables in the positions corresponding to the same i attributes for which $\bar{a}_{[t_0]}$ has constants.

In Case 1, $\bar{a}_{[t_1]}$ can have either a constant or a variable in every position corresponding to the other attributes $A_2, \ldots, A_m$, and therefore there are 2^{m-1} possibilities for such subtuples. In Case 2, $\bar{a}_{[t_1]}$ can have either a constant or a variable in every position corresponding to the other $m - 1 - i$ attributes, and therefore there are 2^{m-1-i} such subtuples. There are t tuples in I other than t_0, and i can range over $[0, m-1]$. Therefore, $|\Omega_1(I, p)|$ or the total number of different $\bar{a}$'s in $\Omega(I, p)$ that do not determine p is

$$\sum_{i=0}^{m-1} \binom{m-1}{i} (2^{m-1} + 2^{m-1-i})^t.$$

The information content of p is then obtained by dividing this number by $2^{n-1} = 2^{m(t+1)-1}$, which proves Claim 4.11:

$$\begin{aligned} \textsc{Ric}_I(p \mid \Sigma) &= \frac{1}{2^{m(t+1)-1}} \sum_{i=0}^{m-1} \binom{m-1}{i} (2^{m-1} + 2^{m-1-i})^t \\ &= \frac{1}{2^{m+t-1}} \sum_{i=0}^{m-1} \binom{m-1}{i} (1 + 2^{-i})^t. \end{aligned}$$

The following shows that as long as $t > \log_{4/3}(1/\varepsilon)$ (i.e., $k > (1+\log_{4/3}(1/\varepsilon))/(m-1)$), for the instance in Fig. 4 and position p of the gray cell the information content is less than $2^{1-m} + \varepsilon$:

$$\begin{aligned} \textsc{Ric}_I(p \mid \Sigma) &= \frac{1}{2^{m+t-1}} \sum_{i=0}^{m-1} \binom{m-1}{i} (1 + 2^{-i})^t \\ &= \frac{1}{2^{m+t-1}} \left(2^t + \sum_{i=1}^{m-1} \binom{m-1}{i} (1 + 2^{-i})^t \right) \\ &< 2^{1-m} + \frac{1}{2^{m+t-1}} \sum_{i=0}^{m-1} \binom{m-1}{i} (1 + 2^{-1})^t \\ &= 2^{1-m} + \frac{1}{2^{m+t-1}} \left(2^{m-1} (\frac{3}{2})^t \right) \\ &= 2^{1-m} + \left(\frac{3}{4}\right)^t \\ &< 2^{1-m} + \varepsilon. \end{aligned}$$

(b) Let Σ be an arbitrary set of FDs over a relational schema R, $I \in \mathit{inst}(R, \Sigma)$, $p = (R, t_0, A_1) \in \mathit{Pos}(I)$, and $\bar{a} \in \Omega(I, p)$. Let $\bar{a}_{[t_0]}$ denote the subtuple in $\bar{a}$ corresponding to t_0. It is easy to see that if $\bar{a}_{[t_0]}$ has variables in all positions corresponding to attributes $A_2, \ldots, A_m$, then $\bar{a}$ does not determine p, no matter what the other positions in $\bar{a}$ contain. All the other $n - m$ positions in $\bar{a}$ can therefore contain either a constant or a variable, so the number of $\bar{a}$'s that do not determine p is at least 2^{n-m}; that is, $|\Omega_1(I, p)| \geq 2^{n-m}$. Thus, using Lemma 3.4, the information content of p is at least $\frac{2^{n-m}}{2^{n-1}} = 2^{1-m}$. This proves Proposition 4.9. □

Proof of Proposition 4.10. We need to show that:

(a). for an arbitrary $\varepsilon > 0$ and every $m > 2$, there exists a 3NF schema (R, Σ) with $|\mathit{sort}(R)| = m$, an instance $I \in \mathit{inst}(R, \Sigma)$, and a position $p \in \mathit{Pos}(I)$ such that $\textsc{Ric}_I(p \mid \Sigma) < 2^{2-m} + \varepsilon$.

(b). for every (R, Σ) in 3NF with $|\mathit{sort}(R)| = m$, every instance $I \in \mathit{inst}(R, \Sigma)$, and every position $p \in \mathit{Pos}(I)$, we have $\textsc{Ric}_I(p \mid \Sigma) \geq 2^{2-m}$.

(a) Consider $R(A_1, \dots, A_m)$ and $\Sigma = \{A_1A_2 \to A_3 \dots A_m,\ A_3 \to A_1,\ \dots,\ A_m \to A_1\}$. Clearly (R, Σ) is in 3NF. Consider the instance $I \in \mathit{inst}(R, \Sigma)$ shown in Fig. 3. Let t_0 denote the first tuple depicted in this figure, and $p = (R, t_0, A_1)$ denote the position of the gray cell.

Claim 4.12. *The information content of position p is*

$$\mathrm{RIC}_I(p \mid \Sigma) = \frac{1}{2^{m+k-3}} \sum_{i=0}^{m-2} \binom{m-2}{i} (1 + 2^{-i})^{k-1}.$$

Proof. Let $\bar{a}$ be an arbitrary vector in $\Omega(I, p)$. Let $\bar{a}_{[t_0]}$ denote the subtuple in $\bar{a}$ corresponding to t_0, and suppose that $\bar{a}_{[t_0]}$ has constants in the positions corresponding to i attributes among $A_3, \dots A_m$, and it has variables in the positions corresponding to the remaining $m - 2 - i$ attributes. Then $\bar{a}$ does not determine p if and only if for any arbitrary subtuple $\bar{a}_{[t_1]}$ in $\bar{a}$ corresponding to a tuple $t_1 \in I$, $t_1 \neq t_0$, either

(1) the subtuple $\bar{a}_{[t_1]}$ has a variable in the position corresponding to A_1; or

(2) the subtuple $\bar{a}_{[t_1]}$ has a constant in the position corresponding to A_1 but variables in the positions corresponding to the same i attributes for which $\bar{a}_{[t_0]}$ has constants.

In Case 1, $\bar{a}_{[t_1]}$ can have either a constant or a variable in every position corresponding to attributes $A_2, \dots, A_m$, and hence there could be 2^{m-1} such subtuples for every $t_1 \neq t_0$. In Case 2, $\bar{a}_{[t_1]}$ can have either a constant or a variable in every position corresponding to the $m - 1 - i$ attributes, and therefore there are 2^{m-1-i} possible such subtuples. There are $k - 1$ subtuples like $\bar{a}_{[t_1]}$, and i can range over $[0, m-2]$. So far we have not said anything about values corresponding to A_2 in t_0, but since A_1A_2 is a candidate key, in both cases, $\bar{a}_{[t_0]}$ can have either a constant or a variable in that position. Putting it all together, we see that $|\Omega_1(I, p)|$, the total number of different $\bar{a}$'s in $\Omega(I, p)$ that do not determine p is

$$2 \cdot \sum_{i=0}^{m-2} \binom{m-2}{i} (2^{m-1} + 2^{m-1-i})^{k-1}.$$

The information content of p can be obtained by dividing this number by $2^{n-1} = 2^{mk-1}$:

$$\begin{aligned} \mathrm{RIC}_I(p \mid \Sigma) &= \frac{1}{2^{mk-2}} \sum_{i=0}^{m-2} \binom{m-2}{i} (2^{m-1} + 2^{m-1-i})^{k-1} \\ &= \frac{1}{2^{m+k-3}} \sum_{i=0}^{m-2} \binom{m-2}{i} (1 + 2^{-i})^{k-1}. \end{aligned}$$

This proves Claim 4.12.

Now we need to show that for any $\varepsilon > 0$ there is an instance of the form shown in Fig. 3 and a position p in it corresponding to the gray cell such that the information content of p is less than $2^{2-m} + \varepsilon$. Taking p to be the position used in Claim 4.12

we have

$$\begin{aligned}
\mathrm{R{\small IC}}_I(p \mid \Sigma) &= \frac{1}{2^{m+k-3}} \sum_{i=0}^{m-2} \binom{m-2}{i} (1+2^{-i})^{k-1} \\
&= \frac{1}{2^{m+k-3}} \left(2^{k-1} + \sum_{i=1}^{m-2} \binom{m-2}{i} (1+2^{-i})^{k-1} \right) \\
&< 2^{2-m} + \frac{1}{2^{m+k-3}} \sum_{i=0}^{m-2} \binom{m-2}{i} (1+2^{-1})^{k-1} \\
&= 2^{2-m} + \left(\frac{3}{4}\right)^{k-1} \\
&< 2^{2-m} + \varepsilon,
\end{aligned}$$

as long as $k > 1 + \log_{4/3}(1/\varepsilon)$.

(b) Let (R, Σ) be in 3NF, $I \in inst(R, \Sigma)$, $p = (R, t_0, A_1) \in Pos(I)$, and $\bar{a} \in \Omega(I, p)$. Let $\bar{a}_{[t_0]}$ denote the subtuple in $\bar{a}$ corresponding to t_0. We assume that A_1 is a prime attribute, but not a key itself, because otherwise $\mathrm{R{\small IC}}_I(p \mid \Sigma) = 1$ since p would not have any redundancy due to a non-key FD.

It is easy to see that if $\bar{a}_{[t_0]}$ has variables in all positions corresponding to attributes $A_2, \ldots, A_m$, then $\bar{a}$ does not determine p, no matter what the other positions in $\bar{a}$ contain. All the other $n - m$ positions in $\bar{a}$ can therefore contain either a constant or a variable, so there are at least 2^{n-m} $\bar{a}$'s that do not determine p. Since A_1 is prime and not a key by itself, there is at least another attribute A_k such that A_1, A_k belong to a candidate key. If $\bar{a}_{[t_0]}$ has a constant in the position corresponding to A_k and variables in all positions corresponding to the other attributes, then $\bar{a}$ does not determine p since the FD $A_k \to A_1 \notin \Sigma^+$. Thus, there are at least another 2^{n-m} $\bar{a}$'s that do not determine p. Then using Lemma 3.4, the information content of p is at least $\frac{2^{n-m}+2^{n-m}}{2^{n-1}} = 2^{2-m}$, which completes the proof of Proposition 4.10 and thus of Theorem 4.8. □

Combining the results of Theorem 4.8 with fact that $\mathrm{GIC}_{\mathrm{BCNF}}(m) = 1$, for all $m > 0$, we obtain the following comparisons of BCNF and 3NF:

COROLLARY 4.13. *For every $m > 2$:*

—$\mathrm{G{\small AIN}}_{\mathrm{BCNF/3NF^+}}(m) = 2$;
—$\mathrm{G{\small AIN}}_{\mathrm{BCNF/3NF}}(m) = 2^{m-2}$;
—$\mathrm{G{\small AIN}}_{\mathrm{BCNF/All}}(m) = 2^{m-1}$.

We have so far compared normal forms based on the amount of information content that they guarantee for each position in an instance, and concluded that a 3NF$^+$ normalization is the best according to this measure. Now we would like to extend this result, and show that doing a 3NF$^+$ normalization can also have a significant effect on the average information content of instances.

This time we compare normal forms based on *average gain of normalization* function defined as

$$\mathrm{G{\small AIN}A{\small VG}}_{\mathcal{NF}_1/\mathcal{NF}_2}(m) \;=\; \frac{\mathrm{GAVG}_{\mathcal{NF}_1}(m)}{\mathrm{GAVG}_{\mathcal{NF}_2}(m)},$$

where $\mathrm{GAVG}_{\mathcal{NF}_1}(m), \mathrm{GAVG}_{\mathcal{NF}_2}(m)$ are the smallest value of average information content of instances as (R, Σ) ranges over schemas with m attributes satisfying normal forms $\mathcal{NF}_1$ and $\mathcal{NF}_1$ respectively (see Definition 4.2) .

The following theorem shows how the lowest average information content changes when we go from an ordinary 3NF normalization to a 3NF$^+$ one.

THEOREM 4.14. *For every $m > 4$:*

$$\mathrm{GAINAVG}_{3\mathrm{NF}^+/3\mathrm{NF}}(m) \ > \ 2^{\frac{m}{2}-2}.$$

The proof of this theorem follows from the following two propositions.

PROPOSITION 4.15. $\mathrm{GAVG}_{3\mathrm{NF}^+}(m) \geq 1/2 + 1/(2m)$ *for all $m > 2$.*

PROPOSITION 4.16. $\mathrm{GAVG}_{3\mathrm{NF}}(m) \leq 2^{1-\frac{m}{2}}$ *for all $m > 4$.*

PROOF OF PROPOSITION 4.15. We need to show that for every (R, Σ) with $|sort(R)| = m$ and every instance $I \in inst(R, \Sigma)$, we have $\mathrm{AVG}(I \mid \Sigma) \geq 1/2 + 1/(2m)$. Let Σ be an arbitrary non-empty set of FDs over $R(A_1, \ldots, A_m)$ such that (R, Σ) is indecomposable, and $A_1, \ldots, A_{m-1} \to A_m \in \Sigma$ be the FD of the form described in Proposition 4.5. For an instance $I \in inst(R, \Sigma)$ and any position $p = (R, t, A_m) \in Pos(I)$, we have $\mathrm{RIC}_I(p \mid \Sigma) = 1$. If k is the total number of tuples in I, then there is k of these positions. For all other positions $p = (R, t, A_i)$, $i \in [1, m-1]$ corresponding to any of the attributes $A_1, \ldots, A_{m-1}$, we have $\mathrm{RIC}_I(p \mid \Sigma) \geq 1/2$, by Proposition 4.6, and there is $k \cdot (m-1)$ of these positions. If n denotes $\|I\|$, then

$$\begin{aligned} \mathrm{AVG}(I \mid \Sigma) \ &\geq \ \frac{k + k \cdot (m-1) \cdot \frac{1}{2}}{n} \\ &= \frac{1}{2} + \frac{1}{2m}, \end{aligned}$$

which proves the proposition. □

PROOF OF PROPOSITION 4.16. We show that for every $m > 4$ and $\varepsilon > 0$, there exists a schema (R, Σ) with $|sort(R)| = m$ and an instance $I \in inst(R, \Sigma)$, such that (R, Σ) is in 3NF, and $\mathrm{AVG}(I \mid \Sigma) < 2^{1-m/2} + \varepsilon$. Consider $R(A_1, \ldots, A_m)$, where m is an even integer and

$$\begin{aligned} \Sigma \ = \ \{&A_1 \to A_3, A_3 \to A_5, \ldots, A_{m-3} \to A_{m-1}, A_{m-1} \to A_1, \\ &A_2 \to A_4, A_4 \to A_6, \ldots, A_{m-2} \to A_m, A_m \to A_2\}. \end{aligned}$$

Consider the instance $I \in inst(R, \Sigma)$ shown in Fig. 5. Let t_0 denote the first tuple depicted in this figure, and $p = (R, t_0, A_1)$ denote the position of the gray cell.

CLAIM 4.17. *The information content of position p is*

$$\mathrm{RIC}_I(p \mid \Sigma) = \frac{1}{2^{\frac{m}{2}+k-2}} \sum_{i=0}^{\frac{m}{2}-1} \binom{\frac{m}{2}-1}{i} (1 + 2^{-i})^{k-1}.$$

PROOF. Let $\bar{a}$ be an arbitrary vector in $\Omega(I, p)$. Let $\bar{a}_{[t_0]}$ denote the subtuple in $\bar{a}$ corresponding to t_0, and suppose that $\bar{a}_{[t_0]}$ has constants in the positions corresponding to i attributes among $A_3, A_5, \ldots A_{m-1}$, and it has variables in the

A_1	A_2	A_3	A_4	$\dots$	A_{m-1}	A_m
1	1	1	1	$\dots$	1	1
1	2	1	2	$\dots$	1	2
⋮	⋮	⋮	⋮		⋮	⋮
1	k	1	k	$\dots$	1	k
2	1	2	1	$\dots$	2	1
2	2	2	2	$\dots$	2	2
⋮	⋮	⋮	⋮		⋮	⋮
2	k	2	k	$\dots$	2	k
⋮						
k	1	k	1	$\dots$	k	1
k	2	k	2	$\dots$	k	2
⋮	⋮	⋮	⋮		⋮	⋮
k	k	k	k	$\dots$	k	k

Fig. 5. A database instance for the proof of Proposition 4.16.

positions corresponding to the remaining $m/2 - i$ attributes with odd subscripts. Then $\bar{a}$ does not determine p if and only if for any arbitrary subtuple $\bar{a}_{[t_1]}$ in $\bar{a}$ corresponding to a tuple t_1 among the first k tuples in Fig. 5, $t_1 \neq t_0$, either

(1) the subtuple $\bar{a}_{[t_1]}$ has a variable in the position corresponding to A_1; or

(2) the subtuple $\bar{a}_{[t_1]}$ has a constant in the position corresponding to A_1, but it has variables in the positions corresponding to the same i attributes among $A_3, A_5, \dots A_{m-1}$ for which $\bar{a}_{[t_0]}$ has constants.

In Case 1, $\bar{a}_{[t_1]}$ can have either a constant or a variable in every position corresponding to attributes $A_2, \dots, A_m$, and hence there could be 2^{m-1} such subtuples for every $t_1 \neq t_0$. In Case 2, $\bar{a}_{[t_1]}$ can have either a constant or a variable in every position corresponding to the $m - 1 - i$ attributes, and therefore there are 2^{m-1-i} possible such subtuples. There are $k - 1$ subtuples like $\bar{a}_{[t_1]}$, and i can range over $[0, m/2 - 1]$. So far we have not said anything about positions corresponding to attributes $A_2, A_4, \dots, A_m$ in t_0 ($m/2$ positions) and all positions in tuples that are not among the first k tuples in Fig. 5 ($mk(k - 1)$ positions). Since these positions do not have anything to do with the value in position p, $\bar{a}$ can have either a constant or a variable for those positions and still do not determine p. Putting it all together, we see that $|\Omega_1(I, p)|$, the total number of different $\bar{a}$'s in $\Omega(I, p)$ that do not determine p is

$$2^{\frac{m}{2}} \cdot 2^{mk(k-1)} \cdot \sum_{i=0}^{\frac{m}{2}-1} \binom{\frac{m}{2}-1}{i} (2^{m-1} + 2^{m-1-i})^{k-1}.$$

The information content of p can be obtained by dividing this number by $2^{n-1} =$

2^{mk^2-1}:

$$\begin{aligned}
\text{RIC}_I(p \mid \Sigma) &= \frac{2^{\frac{m}{2}} \cdot 2^{mk(k-1)}}{2^{mk^2-1}} \sum_{i=0}^{\frac{m}{2}-1} \binom{\frac{m}{2}-1}{i} (2^{m-1} + 2^{m-1-i})^{k-1} \\
&= \frac{1}{2^{\frac{m}{2}+k-2}} \sum_{i=0}^{\frac{m}{2}-1} \binom{\frac{m}{2}-1}{i} (1 + 2^{-i})^{k-1}.
\end{aligned}$$

This proves Claim 4.17.

Now we proceed with the proof of Proposition 4.16. It can be easily observed that instance $I \in inst(R, \Sigma)$ shown in Fig. 5 is symmetric with respect to all positions, and therefore the information content of all positions is the same as the value that we calculated in Claim 4.17 since the FDs are also symmetric. Then we have

$$\begin{aligned}
\text{AVG}(I \mid \Sigma) &= \frac{1}{2^{\frac{m}{2}+k-2}} \sum_{i=0}^{\frac{m}{2}-1} \binom{\frac{m}{2}-1}{i} (1 + 2^{-i})^{k-1} \\
&= \frac{1}{2^{\frac{m}{2}+k-2}} \left(2^{k-1} + \sum_{i=1}^{\frac{m}{2}-1} \binom{\frac{m}{2}-1}{i} (1 + 2^{-i})^{k-1} \right) \\
&< 2^{1-\frac{m}{2}} + 2^{2-\frac{m}{2}-k} \cdot \sum_{i=1}^{\frac{m}{2}-1} \binom{\frac{m}{2}-1}{i} (1 + 2^{-1})^{k-1} \\
&= 2^{1-\frac{m}{2}} + 2^{1-\frac{m}{2}} \cdot (\frac{m}{2} - 1) \cdot (\frac{3}{4})^{k-1} \\
&< 2^{1-\frac{m}{2}} + \varepsilon
\end{aligned}$$

if we take $k > 1 + \log_{4/3}(\frac{m/2-1}{\varepsilon \cdot 2^{m/2-1}})$, which completes the proof of Proposition 4.16 and hence that of Theorem 4.14. □

5. CONCLUSIONS

Using the information-theoretic framework of [Arenas and Libkin 2005], we presented a measure for analyzing database designs based on how much redundant data the database can potentially store. For a relation schema with functional dependencies, guaranteed information content of the schema represents the highest redundancy, or equivalently the lowest information content, allowed by that schema for instances. We showed how this measure can be calculated for a given schema, which can be used to decide whether normalizing or decomposing the schema into smaller relations is necessary. We were motivated by two facts: first, normalizing a database that does not contain much redundancy is a poor design decision that leads to inefficient query answering; and second, a database with too much redundancy is highly prone to update anomalies and inconsistencies.

Our next result concerns the normalization of relational databases with functional dependencies. We showed that when preserving functional dependencies is critical, a minimum amount of redundancy must be tolerated, which we call the price of dependency preservation. To achieve this minimum redundancy, one has to normalize the database into a 3NF$^+$ design, which is the name we used for good

3NF designs produced by the main 3NF synthesis algorithm in [Abiteboul et al. 1995]. Doing an arbitrary 3NF normalization can also reduce the redundancy, however, by a small factor. In this analysis, normal forms were compared based on their guaranteed information content, which represents the highest redundancy allowed by the normal form for a data value. The results were extended for guaranteed average information content that represents the highest redundancy allowed by a normal form for values in a database on average.

We would like to extend these results in several ways. First, we would like to use the information-theoretic approach to see whether we can find a natural analog of 3NF for hierarchical databases such as XML. Such a normal form should guarantee a reasonable information content for XML documents satisfying the normal form, and all non-satisfying XML documents should be decomposable into this normal form in a dependency preserving manner. The challenge is that we do not yet have an adequate understanding of the notion of dependency preserving normalization for XML documents. The complicated structure of XML documents makes it nontrivial to define such a concept. The good news is that functional dependencies can be defined for XML in a natural way [Arenas and Libkin 2004], and the information-theoretic framework is applicable to normal forms defined for XML [Arenas and Libkin 2005].

Besides functional dependencies, there are other constraints, such as multivalued or inclusion dependencies, that can make a data value redundant. The information-theoretic framework is capable of representing the redundancy of data with respect to these constraints as well [Arenas and Libkin 2005]. It would be interesting to ask whether we can extend the notion of guaranteed information content for dependencies beyond FDs, and if we can calculate the potential redundancy of instances of a schema with inclusion or multivalued dependencies.

ACKNOWLEDGMENTS

We thank anonymous referees for very helpful comments. Kolahi was supported by a grant from NSERC. Libkin was supported by EPSRC grants E005039 and G049165, and EU FET-Open Project FoX (grant agreement FP7-ICT-233599).

REFERENCES

ABITEBOUL, S., HULL, R., AND VIANU, V. 1995. *Foundations of Databases.* Addison-Wesley.

AHO, A. V., BEERI, C., AND ULLMAN, J. D. 1979. The theory of joins in relational databases. *ACM Transactions on Database Systems 4,* 3, 297–314.

ARENAS, M. AND LIBKIN, L. 2004. A normal form for XML documents. *ACM Transactions on Database Systems 29*, 195–232.

ARENAS, M. AND LIBKIN, L. 2005. An information-theoretic approach to normal forms for relational and XML data. *Journal of the ACM 52,* 2, 246–283.

BEERI, C., BERNSTEIN, P. A., AND GOODMAN, N. 1978. A sophisticate's introduction to database normalization theory. In *Proceedings of the 4th International Conference on Very Large Data Bases.* 113–124.

BEERI, C., DOWD, M., FAGIN, R., AND STATMAN, R. 1984. On the structure of Armstrong relations for functional dependencies. *Journal of the ACM 31,* 1, 30–46.

BERNSTEIN, P. A. 1976. Synthesizing third normal form relations from functional dependencies. *ACM Transactions on Database Systems 1,* 4, 277–298.

BERNSTEIN, P. A. AND GOODMAN, N. 1980. What does boyce-codd normal form do? In *Proceedings of the 6th International Conference on Very Large Data Bases*. IEEE Computer Society, 245–259.

BISKUP, J. 1995. Achievements of relational database schema design theory revisited. In *Semantics in Databases*. 29–54.

BISKUP, J., DAYAL, U., AND BERNSTEIN, P. A. 1979. Synthesizing independent database schemas. In *Proceedings of the 1979 ACM SIGMOD International Conference on Management of Data*. ACM, 143–151.

BISKUP, J. AND MEYER, R. 1987. Design of relational database schemes by deleting attributes in the canonical decomposition. *J. Comput. Syst. Sci. 35,* 1, 1–22.

CAVALLO, R. AND PITTARELLI, M. 1987. The theory of probabilistic databases. In *Proceedings of the 13th International Conference on Very Large Data Bases*. 71–81.

COVER, T. M. AND THOMAS, J. A. 1991. *Elements of Information Theory*. John Wiley & Sons.

DALKILIC, M. M. AND ROBERTSON, E. L. 2000. Information dependencies. In *Proceedings of the 19th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. 245–253.

DEMETROVICS, J. AND THI, V. 1987. Keys, antikeys and prime attributes. *Annales Univ. Sci., Sect. Comp., Budapest 8*, 35–52.

DEWSON, R. 2006. *Beginning SQL Server 2005 for Developers: From Novice to Professional*. Apress.

FAGIN, R. 1979. Normal forms and relational database operators. In *Proceedings of the 1979 ACM SIGMOD International Conference on Management of Data*. 153–160.

FAGIN, R. 1981. A normal form for relational databases that is based on domians and keys. *ACM Transactions on Database Systems 6,* 3, 387–415.

GREENWALD, R., STACKOWIAK, R., AND STERN, J. 2007. *Oracle Essentials: Oracle Database 11g (4th Edition)*. O'Reilly Media.

KANELLAKIS, P. C. 1990. Elements of relational database theory. 1073–1156.

KIFER, M., BERNSTEIN, A., AND LEWIS, P. M. 2006. *Database systems : an application-oriented approach*. Addison-Wesley.

KOLAHI, S. 2007. Dependency-preserving normalization of relational and XML data. *Journal of Computer and System Sciences 73,* 4, 636–647.

KOLAHI, S. AND LIBKIN, L. 2006. On redundancy vs dependency preservation in normalization: an information-theoretic study of 3NF. In *Proceedings of the 25th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. 114–123.

LEDOUX, C. H. AND PARKER, D. S. 1982. Reflections on boyce-codd normal form. In *Proceedings of the 8th International Conference on Very Large Data Bases*. Morgan Kaufmann, 131–141.

LEE, T. T. 1987. An information-theoretic analysis of relational databases - part i: Data dependencies and information metric. *IEEE Transactions on Software Engineering 13,* 10, 1049–1061.

LEVENE, M., LEVENE, M., AND LOIZOU, G. 1999. *A Guided Tour of Relational Databases and Beyond*. Springer-Verlag, London, UK.

LEVENE, M. AND LOIZOU, G. 2003. Why is the snowflake schema a good data warehouse design? *Information Systems 28,* 3, 225–240.

LEVENE, M. AND VINCENT, M. W. 2000. Justification for inclusion dependency normal form. *IEEE Transactions on Knowledge and Data Engineering 12,* 2, 281–291.

LING, T. W., TOMPA, F. W., AND KAMEDA, T. 1981. An improved third normal form for relational databases. *ACM Transactions on Database Systems 6,* 2, 329–346.

MANNILA, H. AND RÄIHÄ, K.-J. 1986. Design by example: an application of Armstrong relations. *Journal of Computer and System Sciences 33,* 3, 126–141.

STEPHENS, R. K. AND PLEW, R. R. 2002. *Sams Teach Yourself SQL in 21 Days (4th Edition)*. Sams.

VALIANT, L. G. 1979. The complexity of enumeration and reliability problems. *SIAM J. Comput. 8,* 3, 410–421.

VINCENT, M. W. 1999. Semantic foundations of 4NF in relational database design. *Acta Informatica 36,* 3, 173–213.

ZANIOLO, C. 1982. A new normal form for the design of relational database schemata. *ACM Transactions on Database Systems 7,* 3, 489–499.