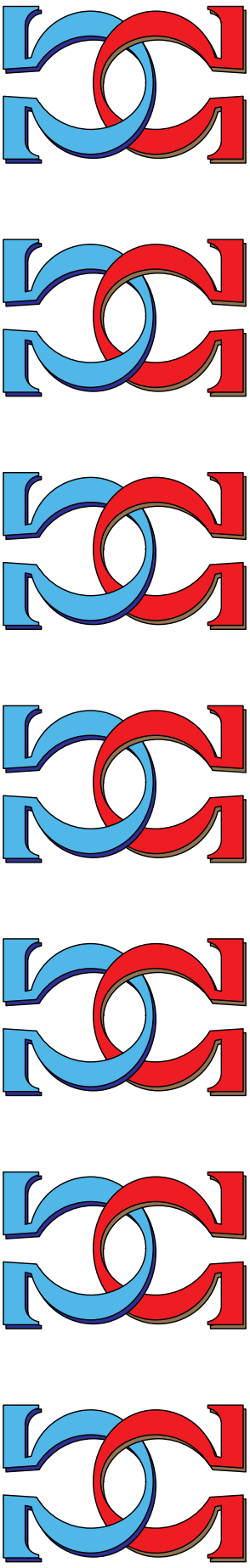

CDMTCS
Research
Report
Series

The Implication Problem of Data Dependencies over SQL Table Definitions: Axiomatic, Algorithmic and Logical Characterizations

Sven Hartmann
Department of Informatics,
Clausthal University of Technology,
Clausthal-Zellerfeld, Germany

Sebastian Link
Department of Computer Science,
University of Auckland,
Auckland, New Zealand

CDMTCS-425
October 2012

Centre for Discrete Mathematics and
Theoretical Computer Science

The implication problem of data dependencies over SQL table definitions: axiomatic, algorithmic and logical characterizations

Sven Hartmann
Department of Informatics
Clausthal University of Technology
Clausthal-Zellerfeld, Germany
sven.hartmann@tu-clausthal.de

Sebastian Link
Department of Computer Science
The University of Auckland, Private Bag 92019
Auckland, New Zealand
s.link@auckland.ac.nz

October 10, 2012

Abstract

We investigate the implication problem for classes of data dependencies over SQL table definitions. Under Zaniolo's "no information" interpretation of null markers we establish an axiomatization and algorithms to decide the implication problem for the combined class of functional and multivalued dependencies in the presence of NOT NULL constraints. The resulting theory subsumes three previously orthogonal frameworks. We further show that the implication problem of this class is equivalent to that in a propositional fragment of Cadoli and Schaerf's family of para-consistent $\mathcal{S}$-3 logics. In particular, $\mathcal{S}$ is the set of variables that correspond to attributes declared NOT NULL. We also show how our equivalences for multivalued dependencies can be extended to Delobel's class of full first-order hierarchical decompositions, and the equivalences for functional dependencies can be extended to arbitrary Boolean dependencies. These dualities allow us to transfer several findings from the propositional fragments to the corresponding classes of data dependencies, and vice versa. We show that our results also apply to Codd's null interpretation "value unknown at present", but not to Imielinski's or-relations utilizing Levene and Loizou's weak possible world semantics. Our findings establish NOT NULL constraints as an effective mechanism to balance not only the

certainty in database relations but also the expressiveness with the efficiency of entailment relations. They also control the degree by which the implication of data dependencies over total relations is soundly approximated in SQL table definitions.

Keywords: Axiomatization, Boolean dependency, Boolean logic, Functional dependency, Incomplete Information, Implication, Logic of Paradox, Multivalued dependency, $\mathcal{S}$-3 logic, SQL

1 Introduction and Motivation

A database system manages a collection of persistent information in a shared, reliable, effective and efficient way. Most commercial database systems are still founded on *the relational model of data* [1]. Data administrators utilize various classes $\mathcal{C}$ of first-order formulae, called *data dependencies*, to restrict the relations in the database to those considered meaningful to the application at hand. A central problem in logic, mathematics and computer science is the *implication problem* of such classes $\mathcal{C}$ [2]. In terms of data dependencies the problem is to decide whether for an arbitrarily given set $\Sigma \cup \{\varphi\}$ of data dependencies in $\mathcal{C}$, Σ implies φ, i.e. whether every relation that satisfies all the elements of Σ also satisfies φ. For databases specifically, solutions to the implication problem are essential for their modeling and design [3, 4], and can advance many data processing tasks such as updates [5, 6, 7, 8], queries [9], security [10], maintenance [11], cleaning [12], integration [13] and exchange [14]. According to Delobel and Adiba [15] the class of functional dependencies (FDs) captures around two-thirds, and the class of multivalued dependencies (MVDs) around one-quarter of all uni-relational dependencies (those defined over a single relation schema) that arise in practice. In particular, MVDs are frequently exhibited in database applications [16], e.g. after de-normalization or in views [3]. The next example illustrates how instances of the implication problem arise naturally from table definitions in SQL [17], which has been the industry standard for defining and querying data for several decades.

Example 1 *Consider an SQL table definition* Supplies *with column headers* A(rticle), S(upplier), L(ocation) *and* C(ost). *The table definition collects information about suppliers that deliver articles from a location at a certain cost.*

```
CREATE TABLE Supplies (
    Article CHAR[20],
    Supplier VARCHAR NOT NULL,
    Location VARCHAR NOT NULL,
    Cost CHAR[8]);
```

Suppose the database management system enforces the following constraints: The FD $A \to S$ says that for every article there is at most one supplier, the FD $AL \to C$ says that the cost is determined by the article and the location, and the MVD $S \twoheadrightarrow L$ says that the locations are determined by the supplier independently of the articles and costs. Do the following meaningful constraints need to be enforced explicitly, or are they already enforced implicitly: i) the FD $A \to C$ and ii) the MVD $A \twoheadrightarrow L$? ∎

While research on the implication problem for the combined class of FDs and MVDs has been extensive, currently existing theories only apply to the two extreme cases of SQL table definitions where all attributes are `NOT NULL` or all attributes are `NULL`, respectively. Note that we adopt SQL terminology here. If an attribute is declared `NOT NULL`, then it does not allow occurrences of the null marker, and if an attribute is declared `NULL`, then it does allow occurrences of the null marker. One may prefer to say that an attribute is (not) nullable instead of saying that an attribute is `(NOT) NULL`, respectively. The classical theory of FDs and MVDs (e.g. [18, 19, 20, 21]) only applies to total relations, i.e. where every attribute is `NOT NULL`. Using Zaniolo's "no information" null marker [22], denoted by `ni`, Lien investigated the combined class of FDs and MVDs over partial relations where every attribute is assumed to be `NULL` [23, 24]. Atzeni and Morfuni studied the class of FDs in the presence of a null-free subschema (NFS) that denotes the set of attributes declared `NOT NULL` [25], but they did not consider MVDs. As Example 1 illustrates, SQL table definitions motivate to study the implication problem for classes $\mathcal{C}$ of data dependencies in the presence of an (arbitrary) NFS: given a relation schema R, a set $\Sigma \cup \{\varphi\}$ of elements in $\mathcal{C}$ and an NFS R_s over R, decide whether Σ implies φ in the presence of R_s, i.e. whether every relation over R that satisfies Σ and R_s also satisfies φ. Indeed, the following example illustrates that the ability to specify arbitrary null-free subschemata has a significant impact on the implication problem of the combined class of FDs and MVDs.

Example 2 *Let $R = ASLC$, $R_s = SL$, $\Sigma = \{A \to S, AL \to C, S \twoheadrightarrow L\}$ as in Example 1. It turns out that Σ implies both the FD $A \to C$ and the MVD $A \twoheadrightarrow L$ in the presence of R_s. However, if $R_s = ASC$, then the relation*

Article	Supplier	Location	Cost
Kiwi	*G6Kiwi*	***ni***	*1.50*
Kiwi	*G6Kiwi*	***ni***	*2.50*

satisfies Σ and R_s, but violates $A \to C$. Indeed, a relation satisfies an FD $X \to Y$ if every pair of its tuples that has matching non-null values on every attribute in X has also matching values on every attribute in Y. In particular, the relation satisfies $AL \to C$ since there are no tuples with non-null values on L. Moreover, a relation satisfies an MVD $X \to Y$ if for every pair of its tuples that has matching non-null values on every attribute in X there is some tuple in the relation that agrees with one of the two tuples on every attribute in XY and agrees with the other tuple on every attribute in $R - XY$. For $R_s = ALC$ the relation

Article	Supplier	Location	Cost
Kiwi	***ni***	*Maunganui*	*1.50*
Kiwi	***ni***	*Taranaki*	*2.50*

satisfies Σ and R_s, but violates $A \to C$ and $A \twoheadrightarrow L$. ∎

In summary, currently existing solutions to the implication problem do not apply to important classes of data dependencies or cover only extreme cases of SQL table definitions that do not do justice to the power of SQL's `NOT NULL` constraint.

Organization. A summary of related work in Section 2 provides further motivation for our study. The contributions of this article are discussed in Section 3. We give the basic definitions required for our treatment of data dependencies over incomplete relations in Section 4. In Section 5 we establish an axiomatization of FDs and MVDs in the presence of an arbitrary NFS, and develop an almost linear time algorithm to decide the corresponding implication problem. In Section 6 we establish the equivalences between the implication of FDs and MVDs in the presence of an NFS and the implication of a propositional fragment in Boolean and $\mathcal{S}$-3 logics. In Section 7 we analyze whether our results also apply to other approaches towards handling incomplete information, including Levene and Loizou's weak possible world semantics of data dependencies for Codd's null marker interpretation "value unknown at present" and for Imielinksi's or-relations. In Section 8 we discuss three data processing applications of our results. We conclude in Section 9 and discuss some possible directions of future work in Section 10.

2 Related Work

Data dependencies have been studied thoroughly in the relational model of data, cf. [3, 26, 27]. Applications comprise almost the full range of database topics including normalization [28, 20, 6, 7, 8], requirements engineering and schema validation [29], data mining [30], database security [31], view maintenance [11] and query optimization [32]. They have received considerable attention in other data models [33, 25, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44]. New application areas involve data cleaning [12], data transformations [45], consistent query answering [46], data exchange [47, 48] and data integration [13].

FDs capture around two-thirds and MVDs around one-quarter of all uni-relational dependencies that arise in applications [15, 16]. Join, embedded, equality- and tuple-generating dependencies are more expressive, but are beyond our scope here [49, 50, 51, 52]. Join dependencies are not Hilbert-style axiomatizable [53], acyclic join dependencies are captured by sets of MVDs [54], and the correspondences to propositional logic fragments do not extend beyond FDs and MVDs [21]. The use of other equality- and tuple-generating dependencies [26] have their major motivation in data exchange [14].

For total relations, Armstrong [55] established the first axiomatization for FDs. Beeri, Fagin, and Howard extended this axiomatization to the combined class of FDs and MVDs [19]. In general, an axiomatization can be applied to enumerate all implicit knowledge from the knowledge given explicitly. In databases specifically, axiomatizations equip administrators and designers with means to validate the correct specification of explicit knowledge, to design and fine-tune databases or to optimize queries. In fact, an axiomatization ensures that all opportunities of utilizing implicit knowledge can be exploited effectively. Moreover, an analysis of the completeness argument can usually provide invaluable hints for finding algorithms that efficiently decide the implication problem. For FDs over total relations, implication can be decided in time linear in the input [56, 57], for MVDs the best known algorithm runs in almost linear time [18, 58, 59, 60, 61, 62, 63].

Such decision algorithms complement the enumeration algorithm by a further reasoning capability that can make efficient, but only partial decisions about implicit knowledge since the input requires a candidate for an implied dependency. Equivalences between the implication of FDs and the implication of Horn clauses in Boolean propositional logic were established by Fagin [64]. These equivalences were extended to the combined class of FDs and MVDs, and its corresponding fragment of Boolean propositional logic by Sagiv, Delobel, Parker and Fagin [21]. This also resulted in the introduction of Boolean dependencies whose implication problem is equivalent to that of Boolean propositional logic [21]. Since the implication problem of Boolean propositional formulae is coNP-complete to decide, the fragment that corresponds to FDs and MVDs is of great practical significance [59, 63].

One of the most important extensions of Codd's basic relational model [1] is incomplete information. This is mainly due to the high demand for the correct handling of such information in real-world applications. Approaches to deal with incomplete information comprise incomplete relations [65, 66, 67], or-relations [68, 69, 70, 71], fuzzy relations [72] and rough sets [73]. In the literature many kinds of null markers have been proposed; for example, "missing" or "value unknown at present" [65, 74, 75], "non-existence" [76], "inapplicable" [75], "no information" [22] and "open" [77]. In particular, Zaniolo's "no information" interpretation allows users of the database to model both non-existent and unknown information in a simple way. Lien [23] axiomatized FDs and MVDs in partial relations under this interpretation, but assumed that all attributes are `NULL`. Atzeni and Morfuni established axiomatizations and linear time algorithms for deciding the implication of FDs combined with existence constraints including null-free subschemata [25]. Using Codd's interpretation "value unknown at present", Levene and Loizou introduced and axiomatized the combined class of weak and strong FDs with respect to a possible world semantics [41]. The axiomatization of strong FDs is given by the Armstrong axioms, while weak FDs have the same axiomatization as the FDs of Lien [23], Atzeni and Morfuni [25]. Here, we establish a unifying framework for the implication problem of FDs and MVDs as motivated by the use of a single null marker and an arbitrary null-free subschema in SQL table definitions.

In previous work [78] the majority of the results in the present article were announced in a ten page long paper. The present article is a result of comprehensive revisions and extensions. In particular, it contains the proofs of all results and a detailed treatment of the subject that allow the reader to gain deep insight into the findings. Particular emphasis has been placed on examples that illustrate the concepts developed and the characterizations established. The motivation for studying the implication problem is extensive, a detailed analysis and comparison to previous work is included, and the impact of the results and techniques on other popular approaches to null markers in the literature are explained. Finally, several areas are indicated to which the results can be applied.

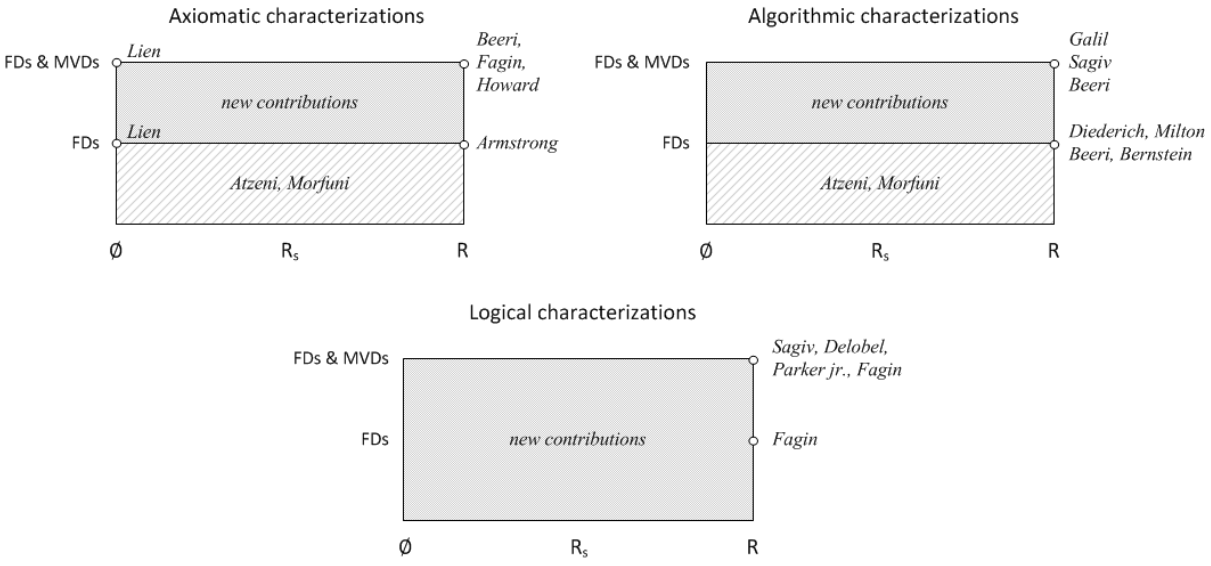

Figure 1: Summary of previous work and new contributions

3 Contributions

In this article we study the implication problem of important classes of data dependencies in the presence of an arbitrary null-free subschema. This will address a gap between database theory and practice. Our study will identify SQL's `NOT NULL` constraint as an *effective* mechanism to not only control the degree of certainty in database relations but also the expressiveness with the efficiency of entailment relations. That is, the more attributes are declared `NOT NULL` the more data dependencies will be implied (and the more semantic knowledge is expressed) and the less efficient it becomes to decide implication. In particular, `NOT NULL` constraints can regulate the degree by which the implication of data dependencies over total relations is soundly approximated in SQL table definitions. Figure 1 contains a summary of previous work and our new contributions. We will now summarize the contributions of the article in detail.

3.1 A unifying theory

As a first contribution of this article we establish a finite axiomatization $\mathfrak{D}$ for the combined class of FDs and MVDs in the presence of an arbitrary NFS. In order to unify the currently existing, but orthogonal theories of FDs and MVDs [25, 19, 23] we first adapt Zaniolo's "no information" interpretation of null markers [22]. Our completeness argument is new, already in the special case of total relations, since we utilize a two-tuple relation. Hence, we obtain the equivalence of the implication problem to the one in the world of two-tuple relations. For the special case where the given NFS is empty, our completeness proof answers the open question whether the axiomatization established for MVDs is also complete for non-standard MVDs [23], i.e. where the attribute set on the left-hand side is empty.

As a further contribution we show that Galil and Sagiv's almost linear time algorithms for computing the dependency basis and deciding the associated implication problem, specifically developed for the case of total relations [59, 63], also apply to the presence of an arbitrary NFS. Our axiomatization $\mathfrak{D}$ shows that for any given relation schema R and any given set $\Sigma \cup \{\varphi\}$ of FDs and MVDs and any given NFS R_s over R, the problem of deciding whether Σ implies φ in the presence of R_s is equivalent to the problem of deciding whether $\Sigma[XR_s]$ implies φ over total R-relations. Herein, X denotes the set of attributes on the left-hand side of φ and $\Sigma[XR_s]$ denotes those elements of Σ whose left-hand side is a subset of XR_s. This allows us to establish an $\mathcal{O}(|\Sigma|+\min\{k_{\Sigma[XR_s]}, \log p_{\Sigma[XR_s]}\}\times|\Sigma[XR_s]|)$ time algorithm to compute the dependency basis $DepB_{\Sigma,R_s}(X)$, and an

$$\mathcal{O}(|\Sigma| + \min\{k_{\Sigma[XR_s]}, \log \bar{p}_{\Sigma[XR_s]}\} \times |\Sigma[XR_s]|)$$

time algorithm for deciding whether the dependency φ with left-hand side attribute set X is implied by Σ in the presence of the NFS R_s. Herein, k_Σ denotes the number of MVDs in Σ, p_Σ denotes the number of sets in the dependency basis $DepB_{\Sigma,R_s}(X)$ of X with respect to Σ and R_s, and $\bar{p}_\Sigma$ denotes the number of sets in $DepB_{\Sigma,R_s}(X)$ that have non-empty intersection with the right-hand side of φ. Note that Galil's algorithms run in linear time when Σ contains only FDs but no MVDs. The upper bounds illustrate the impact of the NFS R_s on the time-complexity of computing the dependency basis and deciding the implication problem.

3.2 Equivalences to propositional fragments of Boolean and para-consistent logics

As a second major contribution of this article we establish equivalences between the implication problem of the combined class of FDs and MVDs in the presence of an NFS R_s and the implication problem of a propositional fragment $\mathcal{F}$ in Cadoli and Schaerf's family of para-consistent $\mathcal{S}$-3 logics [79]. Herein, $\mathcal{S}$ is the set of propositional variables that corresponds to the NFS R_s. We first exemplify how the equivalences to Boolean implication in $\mathcal{F}$, established by Sagiv, Delobel, Parker and Fagin for the special case of total relations [21], apply indirectly also to an arbitrary NFS.

Example 3 *Let $R = ASLC$, $R_s = ALC$, $\Sigma = \{A \to S, AL \to C, S \twoheadrightarrow L\}$, $\varphi_1 = A \twoheadrightarrow L$ and $\varphi_2 = A \to C$. The problems whether Σ implies φ_1 and φ_2 in the presence of R_s are equivalent to the problems whether $\Sigma[ALC] = \{A \to S, AL \to C\}$ implies φ_1 and φ_2 over total R-relations, respectively. The two-tuple relation r*

Article	Supplier	Location	Cost
Kiwi	*G6Kiwi*	*Gisborne*	*1.50*
Kiwi	*G6Kiwi*	*Wellington*	*2.50*

shows that $\Sigma[ALC]$ implies neither φ_1 nor φ_2 over total relations. Let A', S', L' and C' denote the propositional variables that correspond to the attributes of Supplies*, respectively. Let ω_r denote the special truth assignment that assigns true, denoted by $\mathbb{T}$, to a*

variable V' if and only if the two tuples of r agree on the corresponding attribute V. We can see that ω_r is a Boolean model of the Horn clauses $\neg A' \vee S'$ and $\neg A' \vee \neg L' \vee C'$ that result from the FDs in $\Sigma[ALC]$, but not a Boolean model of neither the formula $\neg A' \vee L' \vee (S' \wedge C')$ that results from the MVD φ_1 nor the Horn clause $\neg A' \vee C'$ that results from the FD φ_2. In particular, an MVD $X \twoheadrightarrow Y$ over R results in the Boolean formula $\vee_{A \in X}(\neg A') \vee (\wedge_{B \in Y-X} B') \vee (\wedge_{C \in R-XY} C')$ [21]. This example illustrates the strong correspondence between counter-example relations for the implication of FDs and MVDs over total relations and counter-example truth assignments for the Boolean implication of the propositional fragment $\mathcal{F}$ [21]. ∎

The equivalence to Boolean implication in the fragment $\mathcal{F}$ is indirect since the input (Σ, φ, R_s) to the implication problem in the presence of an NFS is first converted into the instance $(\Sigma[XR_s], \varphi)$ of the implication problem over total relations. The following example illustrates a direct characterization of FD and MVD implication in the presence of an NFS R_s in terms of $\mathcal{S}$-3 implication for the propositional fragment $\mathcal{F}$. Consequently, we gain the additional insight that the NFS R_s corresponds to the set $\mathcal{S}$ of propositional variables that must be interpreted classically, i.e. for all $V' \in \mathcal{S}$ we have that either V' or $\neg V'$ is true, and not both.

Example 4 *Let $R = ASLC$, $R_s = ALC$, $\Sigma = \{A \to S, AL \to C, S \twoheadrightarrow L\}$, $\varphi_1 = A \twoheadrightarrow L$ and $\varphi_2 = A \to C$ as in Example 3. The relation*

Article	Supplier	Location	Cost
Kiwi	***ni***	*Gisborne*	*1.50*
Kiwi	***ni***	*Wellington*	*2.50*

*is a two-tuple counter-example for the implication of the MVD φ_1 and the FD φ_2 by Σ in the presence of R_s. For partial two-tuple relations we define the special $\mathcal{S}$-3 interpretation that assigns $\mathbb{T}$ to V' and false, denoted by $\mathbb{F}$, to $\neg V'$ if the two tuples agree on V and are different from **ni**, assigns $\mathbb{T}$ to V' and $\neg V'$ if the two tuples are both equal to **ni** on V, and assigns $\mathbb{F}$ to V' and $\mathbb{T}$ to $\neg V'$ if the two tuples disagree on V. Specifically for this example, this $\mathcal{S}$-3 interpretation is a model of the Horn formulae $\neg A' \vee S'$ and $\neg A' \vee \neg L' \vee C'$ and the formula $\neg S' \vee L' \vee (A' \wedge C')$, but not a model of neither the formula $\neg A' \vee L' \vee (S' \wedge C')$ nor the Horn formula $\neg A' \vee C'$. Note that the variable S' does not belong to the set $\mathcal{S}$, i.e. the assignment of $\mathbb{T}$ to both S' and $\neg S'$ conforms to the notion of an $\mathcal{S}$-3 interpretation.* ∎

The special case where $\mathcal{S}$ corresponds to the entire relation schema R covers Sagiv, Delobel, Parker and Fagin's equivalence between the implication of FDs and MVDs and the *BL* implication of the fragment $\mathcal{F}$ in Boolean logic *BL* [64, 21]. For the special case where $\mathcal{S} = \emptyset$ the implication of Lien's class of FDs and MVDs corresponds to *LP* implication of $\mathcal{F}$ in Graham Priest's well-known *Logic of Paradox LP* [80]. The implication of Atzeni and Morfuni's class of FDs in the presence of an NFS R_s [25] corresponds to $\mathcal{S}$-3 implication of propositional Horn clauses.

When proving the equivalence to $\mathcal{S}$-3 implication we do not simply extend the proof arguments applied to the special case of total relations [21]. In this special case, Sagiv,

Delobel, Parker and Fagin showed the strong result that every relation that satisfies a given set Σ of FDs and MVDs and violates a given FD or MVD φ has a two-tuple *subrelation* that satisfies Σ and violates φ. While we show that the result holds in the context of "no information" nulls, we demonstrate that it is not required to establish the equivalence, even in the special case of total relations. Instead, we simply utilize the two-tuple relation from the completeness proof for the axiomatization $\mathfrak{D}$.

3.3 Further impact and applications

As the third contribution we show that our results carry over to Codd's null marker `unk` (value unknown at present) but not to Imielinski's or-relations under Levene and Loizou's weak possible world semantics [41]. We also illustrate three major areas to which our reasoning abilities can be applied: updates, queries and access control.

3.4 Further equivalences

As the final major contribution of this article we illustrate the wide applicability of our proof techniques to establish equivalences for further classes of data dependencies. These are reported in the electronic appendix. From the case of total relations it is known that the equivalences to *BL* implication do not extend to more general classes of dependencies such as join or embedded dependencies [21]. Delobel introduced the class of full first-order hierarchical decompositions (FOHDs) [81] as an important subclass of join dependencies. As a generalization of the class of MVDs, we introduce the class of FOHDs to the context of the "no information" nulls. We establish an equivalence between the implication problem of the combined class of FDs and FOHDs in the presence of an NFS and that of a propositional fragment in both Boolean and $\mathcal{S}$-3 logics. As an application of the special $\mathcal{S}$-3 interpretations derived from two-tuple relations, cf. Example 4, we introduce and analyze the class of Boolean dependencies (BDs) in the presence of an NFS. This class subsumes the class of BDs from total relations [21] and Atzeni and Morfuni's class of FDs in the presence of an NFS. In particular, we obtain the equivalence between the implication problem for BDs in the presence of an NFS R_s and that of propositional formulae in $\mathcal{S}$-3 logics.

As an application of our new equivalence we obtain directly results on the time-complexity of the associated implication problem. In fact, detailed findings with respect to Vardi's notions of expression, data and combined complexity transfer straight from the Logic of Paradox [82] to the class of BDs in the absence of an NFS. In the general case where R_s is arbitrary, we obtain immediately a uniform complexity of $\mathcal{O}(|\Sigma| \times |\varphi| \times 2^{|R_s|})$ time for the implication problem whether a set Σ of BDs in Negation Normal Form implies a BD φ in Conjunctive Normal Form in the presence of R_s [79]. The situation is illustrated in Figure 2 for the case where $R = ABC$. Declaring additional attributes as `NOT NULL` increases the certainty and consistency of any of the future database relations, and increases the expressiveness of the entailment relations for the classes of data dependencies studied (in the sense that additional data dependencies are captured implicitly). On the other hand, this results in a decrease in the efficiency of deciding these entailment relations. Hence, by specifying attributes as `NOT NULL` the data administrator has

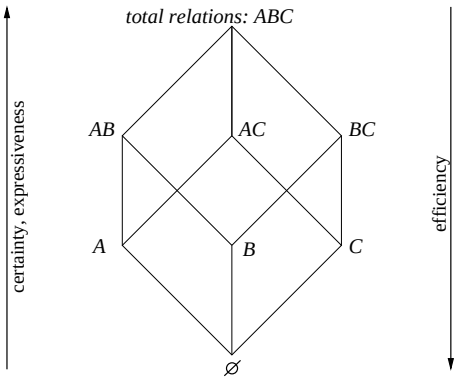

Figure 2: Sound approximations of implication over total relations by null-free subschemata

a powerful mechanism to approximate soundly the implication of data dependencies over total relations.

Our equivalences can also be applied in the opposite direction. In fact, our algorithms for deciding FD and MVD implication in the presence of an arbitrary NFS R_s establish directly new upper bounds for the time-complexity of deciding $\mathcal{S}$-3 implication in the fragment $\mathcal{F}$. Moreover, the axiomatization for the implication of FDs and MVDs in the presence of an NFS applies to $\mathcal{S}$-3 implication of $\mathcal{F}$, too.

As a further application of our techniques we show how reasoning about Boolean dependencies in the presence of an NFS can be simulated by reasoning about Boolean dependencies in the absence of an NFS. In this sense, the special case of an empty NFS is powerful. We also extend the logical characterization of the notion of a dependency basis and attribute set closure from the special case of total relations [21] to the presence of an arbitrary NFS.

4 Preliminaries

We summarize the basic notions required for our treatment of data dependencies over incomplete relations in the following sections.

4.1 Total and partial relations

Let $\mathfrak{A} = \{A_1, A_2, \ldots\}$ be a (countably) infinite set of distinct symbols, called attributes (representing column names of tables). A *relation schema* is a finite non-empty subset R of $\mathfrak{A}$. Each attribute A of a relation schema R is associated with an infinite domain $dom(A)$ which represents the possible values that can occur in column A. Note that the validity of our results only depends on having at least two element values in each domain. This is a consequence of our proof techniques. In order to encompass incomplete information the domain of each attribute contains the null marker, denoted by $\texttt{ni} \in dom(A)$. The intention of `ni` is to mean "no information". This is the most primitive interpretation,

and it can model non-existing as well as unknown information [25, 22]. We stress that the null marker is not a domain value. In fact, it is a purely syntactic convenience that we include the null marker in the domain of each attribute as a distinguished element.

For attribute sets X and Y we may write XY for their set union $X \cup Y$. If $X = \{A_1, \ldots, A_m\}$, then we may write $A_1 \cdots A_m$ for X. In particular, we may write simply A to represent the singleton $\{A\}$. A *tuple* over R (R-tuple or simply tuple, if R is understood) is a function $t : R \to \bigcup_{A \in R} dom(A)$ with $t(A) \in dom(A)$ for all $A \in R$. The null marker occurrence $t(A) = \texttt{ni}$ associated with an attribute A in a tuple t means that "no information" is available about the value $t(A)$ of t on attribute A. For $X \subseteq R$ let $t[X]$ denote the restriction of the tuple t over R to X. A (partial) *relation* r over R is a finite set of tuples over R. Let t_1 and t_2 be two tuples over R. It is said that t_1 *subsumes* t_2 if for every attribute $A \in R$, $t_1(A) = t_2(A)$ or $t_2(A) = \texttt{ni}$ holds. In consistency with previous work [25, 23, 22], the following restriction will be imposed, unless stated otherwise: No relation shall contain two tuples t_1 and t_2 such that t_1 subsumes t_2. With no null markers present this means that no duplicate tuples occur. The validity of our results is independent of this restriction. The restriction is necessary for partial relations to have a lossless decomposition, if (and only if) they exhibit an FD (MVD).

For a tuple t over R and a set $X \subseteq R$, t is said to be X-total, if for all $A \in X$, $t[A] \neq \texttt{ni}$. Similar, a relation r over R is said to be X-total, if every tuple t of r is X-total. A relation r over R is said to be a *total relation*, if it is R-total.

We recall the definition of projection and join operations on partial relations [25, 23]. Let r be some relation over R. Let X be some subset of R. The *projection* $r[X]$ of r on X is the set of tuples t for which (i) there is some $t_1 \in r$ such that $t = t_1[X]$ and (ii) there is no $t_2 \in r$ such that $t_2[X]$ subsumes t and $t_2[X] \neq t$. For $Y \subseteq X$, the *Y-total projection* $r_Y[X]$ of r on X is $r_Y[X] = \{t \in r[X] \mid t \text{ is } Y\text{-total}\}$. Given an X-total relation r over R and an X-total relation s over S such that $X = R \cap S$ the *natural join* $r \bowtie s$ of r and s is the relation over $R \cup S$ which contains those tuples t such that there are some $t_1 \in r$ and $t_2 \in s$ with $t_1 = t[R]$ and $t_2 = t[S]$ [25, 23].

4.2 FDs, MVDs and Null-Free Subschemata

Functional dependencies are important for the relational [56, 83, 1] and other data models [84, 36, 40, 41, 42, 43, 85, 86, 44]. According to Lien [23], a *functional dependency with nulls* (FD) over R is a statement $X \to Y$ where $X, Y \subseteq R$. The FD $X \to Y$ over R is satisfied by a relation r over R, denoted by $\models_r X \to Y$, if and only if for all $t_1, t_2 \in r$ the following holds: if t_1 and t_2 are X-total and $t_1[X] = t_2[X]$, then $t_1[Y] = t_2[Y]$. Recall that $\texttt{ni} \in dom(A)$ for every attribute A. For total relations the FD definition reduces to the standard definition of a functional dependency [3, 27], and so is a sound generalization. It is also consistent with the "no information" interpretation [25, 23]. In fact, tuples with nulls in attributes in X cannot cause a violation of the FD $X \to Y$: the nulls mean that "no information" is available about those attributes. Two X-total tuples t_1, t_2 where $t_1[X] = t_2[X]$ and t_2 is A-total while t_1 is not, violate any FD $X \to Y$ with $A \in Y$: t_1 indicates that "no information" is available about the value for A associated with $t_1[X]$, while t_2 indicates that the value for A associated with $t_2[X] = t_1[X]$ does exist. Hence,

it violates the natural requirement of an FD that if the values for X are the same for two tuples, both tuples must contain the same information for the attributes in Y [25]. Note that if r satisfies the FD $X \to Y$ over R, then $r_X[R] = r_X[XY] \bowtie r_X[X(R-Y)]$.

According to Lien [23], a *multivalued dependency with nulls* (MVD) over R is a statement $X \twoheadrightarrow Y$ where $X, Y \subseteq R$. The MVD $X \twoheadrightarrow Y$ over R is satisfied by a relation r over R, denoted by $\models_r X \twoheadrightarrow Y$, if and only if for all $t_1, t_2 \in r$ the following holds: if t_1 and t_2 are X-total and $t_1[X] = t_2[X]$, then there is some $t \in r$ such that $t[XY] = t_1[XY]$ and $t[X(R-Y)] = t_2[X(R-Y)]$. Informally, the relation r satisfies $X \twoheadrightarrow Y$ when every X-total value determines the set of values on Y independently of the set of values on $R-Y$. This definition of an MVD is a sound generalization of the standard definition of a multivalued dependency over total relations [20, 27]. In particular, it has been shown that MVDs provide a necessary and sufficient condition for the X-total subrelation of a relation to be decomposable into two of its projections without loss of information (in the sense that the X-total subrelation is the natural join of the two projections) [23]. That is, $\models_r X \twoheadrightarrow Y$ if and only if $r_X[R] = r_X[XY] \bowtie r_X[X(R-Y)]$ [23].

Following Atzeni and Morfuni [25], a *null-free subschema* (NFS) over the relation schema R is a an expression R_s where $R_s \subseteq R$. The NFS R_s over R is satisfied by a relation r over R, denoted by $\models_r R_s$, if and only if r is R_s-total. SQL allows the specification of attributes as `NOT NULL`, cf. Example 1. NFSs occur in everyday database practice: the set of attributes declared `NOT NULL` forms the single NFS over the underlying relation schema.

For a set Σ of constraints over some relation schema R we say that a relation r over R *satisfies* Σ, denoted by $\models_r \Sigma$, if r satisfies every $\sigma \in \Sigma$. If for some $\sigma \in \Sigma$ the relation r does not satisfy σ we say that r *violates* σ (and violates Σ) and write $\not\models_r \sigma$ ($\not\models_r \Sigma$). We will consider classes $\mathcal{C}$ of constraints over a single relation schema, e.g. the combined class of FDs and MVDs in the presence of an NFS.

4.3 Implication and inference

In schema design data dependencies are normally specified as semantic constraints over the relations intended to be instances of the schema. During the design process or the lifetime of a database one usually needs to determine further dependencies which are implied by the given ones. Let R be a relation schema, let $R_s \subseteq R$ denote an NFS over R, and let $\Sigma \cup \{\varphi\}$ be a set of data dependencies over R in the class $\mathcal{C}$. We say that Σ (finitely) *implies* φ in the presence of R_s, denoted by $\Sigma \models^{(f)}_{R_s} \varphi$, if every (finite) relation r over R that satisfies Σ and R_s also satisfies φ. For the classes $\mathcal{C}$ of dependencies we consider here we have $\Sigma \models^{f}_{R_s} \varphi$ if and only if $\Sigma \models_{R_s} \varphi$. For this reason, we will not distinguish between implication and finite implication. Instead of proving this result here, we will show an even stronger result in the next section, cf. Corollary 1. If Σ does not imply φ in the presence of R_s we may also write $\Sigma \not\models_{R_s} \varphi$.

The *implication problem for $\mathcal{C}$ in the presence of a null-free subschema* is to decide, given any relation schema R, any NFS R_s over R, and any set $\Sigma \cup \{\varphi\}$ of data dependencies in $\mathcal{C}$ over R, whether $\Sigma \models_{R_s} \varphi$. For the classes $\mathcal{C}$ of dependencies we consider here, the sets $\Sigma \cup \{\varphi\}$ over a relation schema R are always finite. Moreover, if $R_s = \emptyset$ we also write $\Sigma \models \varphi$ instead of $\Sigma \models_\emptyset \varphi$. This covers the case where every attribute is `NULL`. The

case where every attribute is `NOT NULL` is covered when $R_s = R$.

We say that Σ *implies* φ *in the world of two-tuple relations in the presence of an NFS* R_s, denoted by $\Sigma \models_{2,R_s} \varphi$, if every two-tuple relation r over R that satisfies Σ and the NFS R_s also satisfies φ. The *two-tuple implication problem for* $\mathcal{C}$ *in the presence of a null-free subschema* is to decide, given any relation schema R, any NFS R_s over R and any set $\Sigma \cup \{\varphi\}$ of dependencies in $\mathcal{C}$ over R, whether $\Sigma \models_{2,R_s} \varphi$ holds. Again, we may simply write $\Sigma \models_2 \varphi$, if $R_s = \emptyset$.

For a set Σ of data dependencies in $\mathcal{C}$ and an NFS R_s over a relation schema R, let $\Sigma^*_{R_s} = \{\varphi \in \mathcal{C} \mid \Sigma \models_{R_s} \varphi\}$ be its *semantic closure*. In order to determine the semantic closure, one can utilize a syntactic approach by applying inference rules, e.g. those in Table 1. These inference rules have the form

$$\frac{\text{premise}}{\text{conclusion}} \text{ condition},$$

and inference rules without a premise are called axioms. An inference rule is called *sound* for the implication of dependencies in the presence of an NFS, if for all relation schemata R, for all NFSs R_s and for all sets Σ of dependencies over R that form the premise and satisfy the condition of the rule, Σ implies the dependency in the conclusion of the rule in the presence of R_s. For a finite set $\Sigma \cup \{\varphi\}$ of dependencies and a set $\mathfrak{R}$ of inference rules let $\Sigma \vdash_{\mathfrak{R}} \varphi$ denote the *inference* of φ from Σ by $\mathfrak{R}$. That is, there is some sequence $\gamma = [\sigma_1, \ldots, \sigma_n]$ of dependencies such that $\sigma_n = \varphi$ and every σ_i is an element of Σ or is the conclusion that results from an application of an inference rule in $\mathfrak{R}$ to some premises in $\{\sigma_1, \ldots, \sigma_{i-1}\}$. For a finite set Σ of dependencies in $\mathcal{C}$, let $\Sigma^+_{\mathfrak{R}} = \{\varphi \mid \Sigma \vdash_{\mathfrak{R}} \varphi\}$ be its *syntactic closure* under inferences by $\mathfrak{R}$. A set $\mathfrak{R}$ of inference rules is said to be *sound* (*complete*) for the implication of dependencies in $\mathcal{C}$ in the presence of an NFS if for every relation schema R, for every NFS R_s over R and for every set Σ of dependencies in $\mathcal{C}$ over R we have $\Sigma^+_{\mathfrak{R}} \subseteq \Sigma^*_{R_s}$ ($\Sigma^*_{R_s} \subseteq \Sigma^+_{\mathfrak{R}}$). The (finite) set $\mathfrak{R}$ is said to be a (finite) *axiomatization* for the implication of dependencies in $\mathcal{C}$ in the presence of an NFS if $\mathfrak{R}$ is both sound and complete for the implication of dependencies in $\mathcal{C}$ in the presence of an NFS.

5 Dedicated tools for reasoning

In this section we establish the set $\mathfrak{D}$ from Table 1 as the first finite axiomatization for the implication of FDs and MVDs in the presence of an NFS. This subsumes Beeri, Fagin, and Howard's axiomatization over total relations [19], Atzeni and Morfuni's axiomatization of FDs in the presence of an NFS [25], and Lien's axiomatization of FDs and MVDs in the absence of an NFS [23]. Using the axiomatization $\mathfrak{D}$ we show how an arbitrary instance of the implication problem for FDs and MVDs in the presence of an NFS can be reduced to an instance of the implication problem for FDs and MVDs over total relations. This allows us to decide the associated implication problem in the presence of an NFS in almost linear time. Our bounds show the impact of the null-free subschema R_s on the time complexity of deciding the implication problem and computing the dependency basis.

Table 1: Axiomatization $\mathfrak{D}$ for FDs and MVDs in the presence of an NFS R_s

$$\frac{}{XY \to Y}$$
(reflexivity, $\mathcal{R}_{\mathrm{F}}$)

$$\frac{X \to YZ}{X \to Y}$$
(decomposition, $\mathcal{D}_{\mathrm{F}}$)

$$\frac{X \to Y \qquad X \to Z}{X \to YZ}$$
(FD union, $\mathcal{U}_{\mathrm{F}}$)

$$\frac{X \twoheadrightarrow Y}{X \twoheadrightarrow R - Y}$$
(R-complementation, $\mathcal{C}^R_{\mathrm{M}}$)

$$\frac{X \twoheadrightarrow Y \qquad X \twoheadrightarrow Z}{X \twoheadrightarrow YZ}$$
(MVD union, $\mathcal{U}_{\mathrm{M}}$)

$$\frac{X \twoheadrightarrow W \qquad Y \twoheadrightarrow Z}{X \twoheadrightarrow Z - W}\ Y \subseteq X(W \cap R_s)$$
(null pseudo-transitivity, $\mathcal{T}_{\mathrm{M}}$)

$$\frac{X \to Y}{X \twoheadrightarrow Y}$$
(implication, $\mathcal{I}_{\mathrm{FM}}$)

$$\frac{X \twoheadrightarrow W \qquad Y \to Z}{X \to Z - W}\ Y \subseteq X(W \cap R_s)$$
(null mixed pseudo-transitivity, $\mathcal{T}_{\mathrm{FM}}$)

5.1 Soundness

We show first that the set $\mathfrak{D}$ is sound for the implication of FDs and MVDs in the presence of an NFS. This follows from the soundness of the FD inference rules of reflexivity $\mathcal{R}_{\mathrm{F}}$, decomposition $\mathcal{D}_{\mathrm{F}}$, and FD union $\mathcal{U}_{\mathrm{F}}$, cf. [25, 23], R-complementation $\mathcal{C}^R_{\mathrm{M}}$, MVD union $\mathcal{U}_{\mathrm{M}}$ and implication $\mathcal{I}_{\mathrm{FM}}$, cf. [23], and the *null pseudo-transitivity rule* $\mathcal{T}_{\mathrm{M}}$ and the *null mixed pseudo-transitivity rule* $\mathcal{T}_{\mathrm{FM}}$ for the implication of FDs and MVDs in the presence of an NFS, which is shown in the following lemma.

Lemma 1 *The* null pseudo-transitivity rule $\mathcal{T}_M$ *and the* null mixed pseudo-transitivity rule $\mathcal{T}_{FM}$ *are both sound for the implication of FDs and MVDs in the presence of an NFS.*

Proof We show the soundness of the *null pseudo-transitivity rule* $\mathcal{T}_{\mathrm{M}}$ first. Suppose that r satisfies the MVDs $X \twoheadrightarrow W$ and $Y \twoheadrightarrow Z$, and the NFS R_s. Furthermore, let $Y \subseteq X(W \cap R_s)$. Let $t_1, t_2 \in r$ be X-total and such that $t_1[X] = t_2[X]$. Since $\models_r X \twoheadrightarrow W$ there is some $t' \in r$ such that $t'[XW] = t_2[XW]$ and $t'[X(R-W)] = t_1[X(R-W)]$. Since $Y \subseteq X(W \cap R_s) = XW \cap XR_s$ it follows that $t'[Y] = t_2[Y]$ and that t' and t_2 are Y-total. Since $\models_r Y \twoheadrightarrow Z$ there is some $t \in r$ such that $t[YZ] = t'[YZ]$ and $t[Y(R-Z)] = t_2[Y(R-Z)]$. It is easy to see that $t[X(Z-W)] = t_1[X(Z-W)]$ and $t[XW(R-Z)] = t_2[XW(R-Z)]$ hold. That is, r satisfies $X \twoheadrightarrow Z - W$.

Next we show the soundness of the *null mixed pseudo-transitivity rule* $\mathcal{T}_{\mathrm{FM}}$. Suppose that r satisfies the MVD $X \twoheadrightarrow W$, the FD $Y \to Z$, and NFS R_s. Furthermore, let $Y \subseteq X(W \cap R_s)$. Let $t_1, t_2 \in r$ be X-total and such that $t_1[X] = t_2[X]$. We need

to show that $t_1[Z-W] = t_2[Z-W]$. Since $\models_r X \twoheadrightarrow W$ there is some $t \in r$ such that $t[XW] = t_1[XW]$ and $t[X(R-W)] = t_2[X(R-W)]$. Since $Y \subseteq X(W \cap R_s) = XW \cap XR_s$ it follows that $t[Y] = t_1[Y]$ and that t and t_1 are Y-total. Since $\models_r Y \to Z$ it follows that $t[Z] = t_1[Z]$. Let $A \in X(Z-W)$. Then $t_1(A) = t(A) = t_2(A)$. In particular, $t_1[Z-W] = t_2[Z-W]$. That is, r satisfies $X \to Z - W$. ∎

Remark 1 *Neither of the following rules:*

$$\frac{X \twoheadrightarrow W \qquad Y \twoheadrightarrow Z}{X \twoheadrightarrow Z - Y}\, Y \subseteq X(W \cap R_s) \qquad \frac{X \twoheadrightarrow W \qquad Y \to Z}{X \to Z - Y}\, Y \subseteq X(W \cap R_s).$$

is sound. For an illustration that the second rule is not sound let $R = ABCDE$ and $R_s = ABCE$, and let Σ consist of the MVD $A \twoheadrightarrow BC$ and the FD $AB \to CD$. For $b \neq b'$ and $c \neq c'$ the following relation

A	B	C	D	E
a	b	c	***ni***	e
a	b'	c'	***ni***	e

satisfies Σ and R_s but violates $A \to CD$. ∎

Remark 2 *A fundamental rule of inference is Atzeni and Morfuni's* null transitivity rule *[25]*

$$\frac{X \to Y \qquad Y \to Z}{X \to Z}\, Y \subseteq XR_s.$$

The inference

$$\dfrac{\dfrac{\dfrac{X \to Y}{\mathcal{I}_{FM}:\ X \twoheadrightarrow Y} \qquad Y \to Z}{\mathcal{T}_{FM}:\quad X \twoheadrightarrow Z - Y}\, Y \subseteq XR_s \qquad \dfrac{X \to Z}{\mathcal{D}_F:\ X \to Y \cap Z}}{\mathcal{U}_F:\qquad X \to Z}$$

shows how the null transitivity rule can be inferred from $\mathfrak{D}$. ∎

5.2 Completeness

For a relation schema R, an NFS R_s, a set Σ of FDs and MVDs, and an attribute subset X over R let $Dep_{\Sigma,R_s}(X) := \{Y \mid \Sigma \vdash_{\mathfrak{D}} X \twoheadrightarrow Y\}$ denote the set of all attribute subsets Y such that $X \twoheadrightarrow Y$ can be inferred from Σ by $\mathfrak{D}$. The soundness of the union and R-complementation rules imply that

$$(Dep_{\Sigma,R_s}(X), \subseteq, \cup, \cap, (\cdot)^{\mathcal{C}}, \emptyset, R)$$

forms a finite Boolean algebra where $(\cdot)^{\mathcal{C}}$ maps an attribute set Y to its complement $R - Y$. Recall that an element $a \in P$ of a poset $(P, \sqsubseteq, 0)$ with least element 0 is called an *atom* of $(P, \sqsubseteq, 0)$ precisely when $a \neq 0$ and every element $b \in P$ with $b \sqsubseteq a$ satisfies

$b = 0$ or $b = a$. Further, $(P, \sqsubseteq, 0)$ is said to be *atomic* if for every element $b \in P - \{0\}$ there is an atom $a \in P$ with $a \sqsubseteq b$. In particular, every finite Boolean algebra is atomic. Let $DepB_{\Sigma,R_s}(X)$ denote the set of all atoms of $(Dep_{\Sigma,R_s}(X), \subseteq, \emptyset)$. Following Beeri [87] we call $DepB_{\Sigma,R_s}(X)$ the *dependency basis* of X with respect to Σ and R_s.

Moreover, let $X^+_{\Sigma,R_s} = \{A \mid \Sigma \vdash_{\mathfrak{D}} X \to A\}$ denote the *closure* of X with respect to Σ and R_s [56]. The significance of these notions is embodied in the following theorem. The proof shows, in particular, which of the inference rules in $\mathfrak{D}$ are required to establish this result.

Theorem 1 *Let Σ be a set of FDs and MVDs, and R_s an NFS over the relation schema R. Then we have:*

1. *$\Sigma \vdash_{\mathfrak{D}} X \twoheadrightarrow Y$ if and only if $Y = \bigcup \mathcal{Y}$ for some $\mathcal{Y} \subseteq DepB_{\Sigma,R_s}(X)$,*
2. *$\Sigma \vdash_{\mathfrak{D}} X \to Y$ if and only if $Y \subseteq X^+_{\Sigma,R_s}$, and*
3. *if $\Sigma \vdash_{\mathfrak{D}} X \to A$, then $\{A\} \in DepB_{\Sigma,R_s}(X)$.*

Proof *(1)* Let $Y \in Dep_{\Sigma,R_s}(X)$. Since every element b of a Boolean algebra is the union over those atoms a with $a \subseteq b$ it follows that $Y = \bigcup \mathcal{Y}$ for $\mathcal{Y} = \{Z \in DepB_{\Sigma,R_s}(X) \mid Z \subseteq Y\}$.

Vice versa, let $Y = \bigcup \mathcal{Y}$ for some $\mathcal{Y} \subseteq DepB_{\Sigma,R_s}(X)$. Since $X \twoheadrightarrow Z \in \Sigma^+_{\mathfrak{D}}$ holds for every $Z \in \mathcal{Y}$ successive applications of the MVD union rule $\mathcal{U}_{\mathrm{M}}$ result in $X \twoheadrightarrow Y \in \Sigma^+_{\mathfrak{D}}$.
(2) If $X \to Y \in \Sigma^+_{\mathfrak{D}}$, then also $X \to A \in \Sigma^+_{\mathfrak{D}}$ for all $A \in Y$ by means of the FD decomposition rule $\mathcal{D}_{\mathrm{F}}$. Consequently, $Y \subseteq X^+_{\Sigma,R_s}$. Vice versa, if $X \to A \in \Sigma^+_{\mathfrak{D}}$ for all $A \in Y$, then $X \to Y \in \Sigma^+_{\mathfrak{D}}$ by means of the FD union rule $\mathcal{U}_{\mathrm{F}}$.
(3) If $X \to A \in \Sigma^+_{\mathfrak{D}}$, then $X \twoheadrightarrow A \in \Sigma^+_{\mathfrak{D}}$ by means of the implication rule $\mathcal{I}_{\mathrm{FM}}$. According to *(1)* the set $\{A\}$ must be an element of $DepB_{\Sigma,R_s}(X)$. ∎

We will now establish the completeness of $\mathfrak{D}$. Note that our proof is not just a mere extension of the arguments used for the special case where $R_s = R$ [19] or where $R_s = \emptyset$ [23]. In particular, note that in contrast to the previous work on these special cases [19, 23] we only require a two-tuple counter-example relation in our proof. Furthermore, Lien's proof of completeness for the special case $R_s = \emptyset$ did not apply to non-standard MVDs $\emptyset \twoheadrightarrow Y$ and it was an open question whether the rules were also complete for the class of all MVDs [23]. It follows from our proof that this is indeed the case.

Theorem 2 *$\mathfrak{D}$ is a finite axiomatization for the implication of FDs and MVDs in the presence of null-free subschemata.*

Proof Let R be some relation schema, R_s some NFS and Σ a set of FDs and MVDs over R.

Soundness. We need to show that $\Sigma^+_{\mathfrak{D}} \subseteq \Sigma^*_{R_s}$ holds. Let $\varphi \in \Sigma^+_{\mathfrak{D}}$. The soundness of the rules in $\mathfrak{D}$ has been established in previous work and in Lemma 1. A simple induction over the inference length of φ from Σ by $\mathfrak{D}$ shows that $\varphi \in \Sigma^*_{R_s}$.

Table 2: The relation r_φ in the completeness proof

	$X(X^+_\Sigma \cap R_s)$	$(X^+_\Sigma - X) - R_s$	$W_1 \cap R_s$	$W_1 - R_s$	$\cdots$	W_i	$\cdots$	$W_k \cap R_s$	$W_k - R_s$
t_1	$0 \cdots 0$	$\texttt{ni} \cdots \texttt{ni}$	$0 \cdots 0$	$\texttt{ni} \cdots \texttt{ni}$		$0 \cdots 0$		$0 \cdots 0$	$\texttt{ni} \cdots \texttt{ni}$
t_2	$0 \cdots 0$	$\texttt{ni} \cdots \texttt{ni}$	$0 \cdots 0$	$\texttt{ni} \cdots \texttt{ni}$		$1 \cdots 1$		$0 \cdots 0$	$\texttt{ni} \cdots \texttt{ni}$

Completeness. We need to show that $\Sigma^*_{R_s} \subseteq \Sigma^+_\mathfrak{D}$ holds. Suppose first there is some MVD φ, say $X \twoheadrightarrow Y$, such that $\varphi \notin \Sigma^+_\mathfrak{D}$. We will construct a two-tuple relation r_φ that violates $X \twoheadrightarrow Y$ but satisfies Σ and the NFS R_s.

Let $DepB_{\Sigma,R_s}(X)$ be the disjoint union of $\{\{A\} \mid A \in X^+_{\Sigma,R_s}\}$ and $\{W_1, \ldots, W_k\}$, in particular $\{X^+_{\Sigma,R_s}, W_1, \ldots, W_k\}$ forms a partition of R. Since $\varphi \notin \Sigma^+_\mathfrak{D}$ we conclude by Theorem 1 that the attribute set Y is not the union of some elements of $DepB_{\Sigma,R_s}(X)$. Consequently, there is some $i \in \{1, \ldots, k\}$ such that $Y \cap W_i \neq \emptyset$ and $W_i - Y \neq \emptyset$ hold. Let $r_\varphi := \{t_1, t_2\}$ be the relation in Table 2. That is, for all $A \in R$ we have: $t_1(A) \neq t_2(A)$ if and only if $A \in W_i$. Moreover, for all $A \in R$ we have: $t_1(A) \neq \texttt{ni}$ ($t_2(A) \neq \texttt{ni}$) if and only if $A \in XR_sW_i$. Note that r_φ satisfies the following property: if $Z = \bigcup_{B \in \mathcal{B}} B$ for some $\mathcal{B} \subseteq DepB_{\Sigma,R_s}(X)$, then $t_1[Z] = t_2[Z]$ (if $W_i \notin \mathcal{B}$) or $t_1[R - Z] = t_2[R - Z]$ (if $W_i \in \mathcal{B}$). Also note that $t_1[X^+_{\Sigma,R_s}] = t_2[X^+_{\Sigma,R_s}]$.

It follows from the construction that r_φ violates φ and r_φ satisfies the NFS R_s. In order to show that $\varphi \notin \Sigma^*_{R_s}$ it remains to prove that r_φ satisfies Σ.

Let $U \twoheadrightarrow V \in \Sigma$. Suppose that $t_1[U] = t_2[U]$ and t_1, t_2 are U-total. Let

$$W := \bigcup \{W_j \in DepB_{\Sigma,R_s}(X) \mid W_j \cap U \neq \emptyset\}.$$

From $t_1[U] = t_2[U]$ and the construction of r_φ we conclude that $t_1[W] = t_2[W]$. Since W is the union of elements from $DepB_{\Sigma,R_s}(X)$ we conclude by Theorem 1 that $X \twoheadrightarrow W \in \Sigma^+_\mathfrak{D}$. Note that X^+_{Σ,R_s} is also the union of elements from $DepB_{\Sigma,R_s}(X)$, i.e., $X \twoheadrightarrow X^+_{\Sigma,R_s} \in \Sigma^+_\mathfrak{D}$, and by an application of the *MVD union rule* $\mathcal{U}_\mathrm{M}$, $X \twoheadrightarrow X^+_{\Sigma,R_s}W \in \Sigma^+_\mathfrak{D}$, too.

Since $t_1[U] = t_2[U]$ and t_1, t_2 are U-total, the construction of r_φ implies that

$$U \subseteq X((X^+_\Sigma W) \cap R_s).$$

We now apply the *null pseudo-transitivity rule* $\mathcal{T}_\mathrm{M}$ to $X \twoheadrightarrow X^+_{\Sigma,R_s}W \in \Sigma^+_\mathfrak{D}$, $U \twoheadrightarrow V \in \Sigma^+_\mathfrak{D}$ and $U \subseteq X((X^+_{\Sigma,R_s}W) \cap R_s)$ to infer $X \twoheadrightarrow V - X^+_{\Sigma,R_s}W \in \Sigma^+_\mathfrak{D}$. From the definition of X^+_{Σ,R_s} it follows that $X \to X^+_{\Sigma,R_s} \in \Sigma^+_\mathfrak{D}$ by applications of the *FD union rule* $\mathcal{U}_\mathrm{F}$. From $X \to X^+_{\Sigma,R_s} \in \Sigma^+_\mathfrak{D}$ we conclude $X \to ((V - W) \cap X^+_{\Sigma,R_s}) \in \Sigma^+_\mathfrak{D}$ by means of the *decomposition rule* $\mathcal{D}_\mathrm{F}$, and $X \twoheadrightarrow ((V - W) \cap X^+_{\Sigma,R_s}) \in \Sigma^+_\mathfrak{D}$ by an application of the *implication rule* $\mathcal{I}_\mathrm{FM}$. Moreover, an application of the *MVD union rule* $\mathcal{U}_\mathrm{M}$ to $X \twoheadrightarrow V - X^+_{\Sigma,R_s}W \in \Sigma^+_\mathfrak{D}$ and $X \twoheadrightarrow ((V - W) \cap X^+_{\Sigma,R_s}) \in \Sigma^+_\mathfrak{D}$ results in $X \twoheadrightarrow V - W \in \Sigma^+_\mathfrak{D}$. Therefore, $V - W$ is the union of elements from $DepB_{\Sigma,R_s}(X)$. Consequently, $t_1[V - W] = t_2[V - W]$ or $t_1[W(R - V)] = t_2[W(R - V)]$. In summary, we have $t_1[X^+_{\Sigma,R_s}W(V - W)] = t_2[X^+_{\Sigma,R_s}W(V - W)]$ or $t_1[X^+_{\Sigma,R_s}W(R - V)] = t_2[X^+_{\Sigma,R_s}W(R - V)]$. The first case implies $t_1[UV] = t_2[UV]$ and the second case implies $t_1[U(R - V)] = t_2[U(R - V)]$, respectively. In any case we know that r_φ satisfies $U \twoheadrightarrow V$.

Let $U \to V \in \Sigma$. Suppose that $t_1[U] = t_2[U]$ and t_1, t_2 are U-total. As before let

$$W := \bigcup \{W_j \in DepB_{\Sigma,R_s}(X) \mid W_j \cap U \neq \emptyset\}.$$

From $t_1[U] = t_2[U]$ and the construction of r_φ we conclude that $t_1[W] = t_2[W]$. As before we conclude that $X \twoheadrightarrow X^+_{\Sigma,R_s}W \in \Sigma^+_{\mathfrak{D}}$. As before, it follows from the construction of r_φ that

$$U \subseteq X((X^+_{\Sigma,R_s}W) \cap R_s).$$

We now apply the *null mixed pseudo-transitivity rule* $\mathcal{T}_{\rm FM}$ to $X \twoheadrightarrow X^+_{\Sigma,R_s}W \in \Sigma^+_{\mathfrak{D}}$, $U \to V \in \Sigma^+_{\mathfrak{D}}$ and $U \subseteq X((X^+_{\Sigma,R_s}W) \cap R_s)$ to infer $X \to V - X^+_{\Sigma,R_s}W \in \Sigma^+_{\mathfrak{D}}$. As before, we conclude that $X \to ((V-W) \cap X^+_{\Sigma,R_s}) \in \Sigma^+_{\mathfrak{D}}$, and therefore $X \to V - W \in \Sigma^+_{\mathfrak{D}}$ by means of the *FD union rule* $\mathcal{U}_{\rm F}$. Therefore, $V - W \subseteq X^+_{\Sigma,R_s}$. Consequently, $t_1[V-W] = t_2[V-W]$ and since $t_1[W] = t_2[W]$ holds as well, we conclude $t_1[V] = t_2[V]$. Therefore, r_φ satisfies $U \to V$.

Finally, suppose there is some FD φ, say $X \to Y$, such that $\varphi \notin \Sigma^+_{\mathfrak{D}}$. Due to the *FD union rule* $\mathcal{U}_{\rm F}$ there is some $A \in Y$ such that $X \to A \notin \Sigma^+_{\mathfrak{D}}$. It follows that $A \notin X^+_{\Sigma,R_s}$. Without loss of generality let $A \in W_i$. Let r_φ be the two-tuple relation from before. It follows that r_φ violates $X \to Y$ since $t_1[X] = t_2[X]$ and t_1, t_2 are X-total, and $t_1(A) \neq t_2(A)$. We know that r_φ satisfies Σ and the NFS R_s. Consequently, $\varphi \notin \Sigma^*_{R_s}$.

We have shown the completeness of $\mathfrak{D}$ for the implication of FDs and MVDs in the presence of an NFS. ∎

The two-tuple counter-example relation that we utilize in the proof of Theorem 2 allows us to derive the following corollary.

Corollary 1 *Let $\Sigma \cup \{\varphi\}$ denote a set of FDs and MVDs, and let R_s denote an NFS over the relation schema R. Then $\Sigma \models_{R_s} \varphi$ if and only if $\Sigma \models_{2,R_s} \varphi$.*

Proof If $\Sigma \models_{2,R_s} \varphi$ does not hold, then $\Sigma \models_{R_s} \varphi$ does not hold. If $\Sigma \models_{R_s} \varphi$ does not hold, then $\Sigma \vdash_{\mathfrak{D}} \varphi$ does not hold by the soundness of $\mathfrak{D}$. Consequently, we can utilize the same two-tuple relation r_φ as in the proof of Theorem 2 to derive that $\Sigma \models_{2,R_s} \varphi$ does not hold. ∎

In the electronic appendix we show *how* our axiomatization $\mathfrak{D}$ subsumes three axiomatizations [25, 19, 23] for the special cases where i) $R_s = R$ [19], ii) $R_s = \emptyset$ [23], and iii) the set Σ consists of FDs only [25]. That is, we show how the rules in these axiomatizations can be derived from our axiomatization $\mathfrak{D}$ when restricted to the corresponding special case, respectively. Moreover, we show in the electronic appendix an even stronger result than that reported in Corollary 1. That is, if r is an arbitrary relation that satisfies Σ and R_s but violates φ, then there is some two-tuple *subrelation* $r' \subseteq r$ that satisfies Σ and R_s but violates φ.

Remark 3 *The system $\mathfrak{D}' := (\mathfrak{D} - \{\mathcal{T}_{FM}\}) \cup \{\bar{\mathcal{T}}_{FM}\}$, where $\bar{\mathcal{T}}_{FM}$ denotes*

$$\frac{X \twoheadrightarrow Y \qquad Y \to Z}{X \to Z - Y}\, Y \subseteq XR_s,$$

is incomplete. Let $R = ABC$, $R_s = A$, $\Sigma = \{\emptyset \twoheadrightarrow AB, A \to BC\}$ and $\varphi = \emptyset \to C$. Due to the soundness of $\mathcal{T}_{FM}$ it follows that φ is implied by Σ in the presence of R_s. However, φ cannot be inferred from Σ by the system $\mathfrak{D}'$. In particular, an inference of φ from $\emptyset \twoheadrightarrow AB$ and from $AB \to BC$ by an application of $\bar{\mathcal{T}}_{FM}$ would require the attribute B to be an element of R_s. Similar observations show that the system $\mathfrak{D}'' := (\mathfrak{D} - \{\mathcal{T}_M\}) \cup \{\bar{\mathcal{T}}_M\}$, where $\bar{\mathcal{T}}_M$ denotes the rule

$$\frac{X \twoheadrightarrow Y \qquad Y \twoheadrightarrow Z}{X \twoheadrightarrow Z - Y} \, Y \subseteq XR_s,$$

is also incomplete. ∎

5.3 Algorithms

In this subsection we will establish algorithms for i) deciding the implication problem $\Sigma \models_{R_s} \varphi$ for sets $\Sigma \cup \{\varphi\}$ of FDs and MVDs in the presence of an arbitrary NFS R_s, and ii) computing the dependency basis $DepB_{\Sigma,R_s}(X)$ of an attribute set X with respect to Σ and R_s. We will derive a tight worst-case upper time bound that highlights the impact of the NFS R_s. The results follow from a reduction of the implication problem to its counter-part over total relations. The reduction itself is a consequence of our axiomatization.

5.3.1 The special case of total relations

For total relations, Beeri [18] presented an algorithm for computing $DepB_{\Sigma,R}(X)$ that runs in time $\mathcal{O}(|\Sigma|^4)$. It is based on Beeri's rules:

$$\frac{X \twoheadrightarrow W \qquad Y \twoheadrightarrow Z}{X \twoheadrightarrow W \cap Z} \, W \cap Y = \emptyset \qquad \frac{X \twoheadrightarrow W \qquad Y \twoheadrightarrow Z}{X \twoheadrightarrow W - Z} \, W \cap Y = \emptyset \quad .$$

The idea of Beeri's algorithm for computing the dependency basis $DepB_{\Sigma,R}(X)$ is to start with a partition $\mathcal{B} := \{R - X, A \mid A \in X\}$ of R which is then refined incrementally by applying Beeri's rules to sets $W \in \mathcal{B}$ and dependencies $Y \twoheadrightarrow Z$ or $Y \to Z \in \Sigma$ that meet the conditions in Beeri's rules and satisfy $W \cap Z \neq \emptyset$ and $W - Z \neq \emptyset$. In each step, W is split into $W \cap Z$ and $W - Z$. Note that $\Sigma \vdash_{\mathcal{D}} X \twoheadrightarrow W$ for all $W \in \mathcal{B}$ is a loop invariant. The algorithm stops when no further refinement is possible. The final partition $\mathcal{B}$ is then the dependency basis $DepB_{\Sigma,R}(X)$ of X with respect to Σ and R.

More sophisticated implementations of this idea by Hagihara et al. [60] resulted in an $\mathcal{O}(\min\{(f_\Sigma + k_\Sigma)^2 \times d_\Sigma, |\Sigma|^2\})$ time algorithm, and later by Galil [59] resulted in an $\mathcal{O}(|\Sigma| + \min\{k_\Sigma, \log p_\Sigma\} \times |\Sigma|)$ time algorithm. Herein, f_Σ, k_Σ and d_Σ are the numbers of FDs, of MVDs, and of distinct attributes, respectively, in Σ, while p_Σ denotes the number of sets in $DepB_{\Sigma,R}(X)$. Note that Galil's algorithm runs in linear time when Σ contains only FDs but no MVDs.

Let Σ be a set of FDs and MVDs. If $DepB_{\Sigma,R}(X)$ is known, the implication problem $\Sigma \models_R \varphi$ for a given FD or MVD φ with left-hand side X can be decided in linear time.

In particular, $\Sigma \models_R X \to A$ holds when $\{A\} \in DepB_{\Sigma,R}(X)$ and Σ contains a non-trivial FD $Y \to Z$ with $A \in Z$. Galil established an

$$\mathcal{O}(|\Sigma| + \min\{k_\Sigma, \log \bar{p}_\Sigma\} \times |\Sigma|)$$

time algorithm for deciding $\Sigma \models_R \varphi$ [59]. Here, $\bar{p}_\Sigma$ is the number of sets in $DepB_{\Sigma,R}(X)$ that have non-empty intersection with the right-hand side of φ.

5.3.2 The general case

Let $\Sigma[U]$ contain only those dependencies from Σ whose left-hand side is a subset of the attribute set U. For an FD or MVD φ let $lhs(\varphi)$ denote the set of attributes that occur on the left-hand side of φ. Let $\mathfrak{S}_1$ denote an axiomatization for the implication of FDs and MVDs over total relations [27].

Lemma 2 *Let $\Sigma \cup \{\varphi\}$ be a set of FDs and MVDs, and R_s an NFS over the relation schema R. Then the following are equivalent:*

1. $\Sigma \vdash_{\mathfrak{D}} \varphi$,
2. $\Sigma[lhs(\varphi)R_s] \vdash_{\mathfrak{D}} \varphi$, *and*
3. $\Sigma[lhs(\varphi)R_s] \vdash_{\mathfrak{S}_1} \varphi$.

Proof Since $\Sigma[lhs(\varphi)R_s]$ is a subset of Σ, it follows that (2) implies (1). A simple induction over the length of an inference of φ from Σ by $\mathfrak{D}$ shows that φ can already be inferred from $\Sigma[lhs(\varphi)R_s]$ by $\mathfrak{D}$. Hence, (1) also implies (2). The equivalence of (2) and (3) follows from the fact that both $\mathfrak{S}_1$ and $\mathfrak{D}$ are axiomatizations for the implication of FDs and MVDs over total relations. ∎

Corollary 2 *Let Σ be a set of FDs and MVDs, R_s an NFS and X an attribute set over the relation schema R. Then*

1. $X^+_{\Sigma,R_s} = X^+_{\Sigma[XR_s],R_s} = X^+_{\Sigma[XR_s],R}$, *and*
2. $DepB_{\Sigma,R_s}(X) = DepB_{\Sigma[XR_s],R_s}(X) = DepB_{\Sigma[XR_s],R}(X)$. ∎

We conclude that Galil's algorithm, developed specifically for the case of total relations, applies even to the general case of an arbitrary NFS R_s. Indeed, we only need to consider dependencies $Y \twoheadrightarrow Z$ or $Y \to Z \in \Sigma[XR_s]$.

Corollary 3 *Let Σ be a set of FDs and MVDs, R_s an NFS and X an attribute set over the relation schema R. Then Galil's algorithm [59] computes the dependency basis $DepB_{\Sigma,R_s}(X)$ of X with respect to Σ and R_s in time $\mathcal{O}(|\Sigma| + \min\{k_{\Sigma[XR_s]}, \log p_{\Sigma[XR_s]}\} \times |\Sigma[XR_s]|)$.* ∎

Let Σ be a set of FDs and MVDs, and R_s an NFS over R. If $DepB_{\Sigma,R_s}(X)$ is known, the implication problem $\Sigma \models_{R_s} \varphi$ for a given FD or MVD φ with left-hand side X can be decided in linear time. In particular, $\Sigma \models_{R_s} X \to A$ holds when $\{A\} \in DepB_{\Sigma,R_s}(X)$ and $\Sigma[XR_s]$ contains a non-trivial FD $Y \to Z$ with $A \in Z$.

Corollary 4 *Using Galil's algorithm [59], the implication problem* $\Sigma \models_{R_s} \varphi$ *of sets* $\Sigma \cup \{\varphi\}$ *of FDs and MVDs and NFS* R_s *over a relation schema* R *can be decided in time* $\mathcal{O}(|\Sigma| + \min\{k_{\Sigma[lhs(\varphi)R_s]}, \log \bar{p}_{\Sigma[lhs(\varphi)R_s]}\} \times |\Sigma[lhs(\varphi)R_s]|)$.

Proof The problem $\Sigma \models_{R_s} \varphi$ is equivalent to $\Sigma \vdash_{\mathfrak{D}} \varphi$ according to Theorem 2. Lemma 2 shows us that this problem is the same as deciding whether $\Sigma[lhs(\varphi)R_s] \vdash_{\mathfrak{S}_1} \varphi$, and according to the soundness and completeness of $\mathfrak{S}_1$ this is equivalent to deciding whether $\Sigma[lhs(\varphi)R_s] \models_R \varphi$. The last problem can be decided by Galil's algorithm in $\mathcal{O}(|\Sigma| + \min\{k_{\Sigma[lhs(\varphi)R_s]}, \log \bar{p}_{\Sigma[lhs(\varphi)R_s]}\} \times |\Sigma[lhs(\varphi)R_s]|)$ time. ∎

5.3.3 Characterization of implication

The proof of Corollary 4 implies the following characterization of the implication problem in terms of reducing the given set of data dependencies based on the candidate dependency φ and the NFS R_s.

Corollary 5 *Let* $\Sigma \cup \{\varphi\}$ *be a set of FDs and MVDs, and* R_s *an NFS over the relation schema* R. *Then* $\Sigma \models_{R_s} \varphi$ *if and only if* $\Sigma[lhs(\varphi)R_s] \models_R \varphi$. ∎

5.3.4 A correction to Atzeni and Morfuni's algorithm

Based on Theorem 1 an FD $X \to Y$ is implied by an FD set Σ in the presence of an NFS R_s over R if and only if $Y \subseteq X^+_{\Sigma,R_s}$. Atzeni and Morfuni proposed an algorithm NFSClosure(X,Σ,R_s,R) for computing the closure X^+_{Σ,R_s} of an attribute set X with respect to an FD set Σ and an NFS R_s over a relation schema R [25, Algorithm 3, page 14]. We report a flaw of this algorithm and a simple correction of it. As an example of the flaw, consider the relation schema Employment=$\{$*Emp*,*Dept*,*Mgr*$\}$ with the NFS $\{$*Dept*$\}$ and let Σ consist of the two FDs *Emp* $\to$ *Dept* and *Dept* $\to$ *Mgr*. On input ($\{$*Emp*$\}$,Σ,$\{$*Dept*$\}$,Employment), the original algorithm [25] returns the set $\{$*Emp*, *Dept*$\}$. However, the correct result is the set $\{$*Emp*,*Dept*,*Mgr*$\}$. This can be verified easily by an application of the *null transitivity rule*, see Remark 2, to *Emp* $\to$ *Dept*, and *Dept* $\to$ *Mgr* and *Dept* $\in$ Employment$_s$ to infer the FD *Emp* $\to$ *Mgr*. We will now present an algorithm that corrects the flaw.

Algorithm 1 works very similar to Beeri and Bernstein's algorithm for computing the closure in the special case of total relations [56]. The only difference is that, in the presence of an NFS R_s, it suffices to consider FDs $V \to W \in \Sigma$ where $V \subseteq XR_s$ holds. The minor flaw of Atzeni and Morfuni's algorithm [25] results from the incorrectly stricter restriction that $V \subseteq R_s$.

Example 5 *Consider the relation schema* Employment *with the null-free subschema* Employment$_s$ = $\{$*Emp*, *Mgr*$\}$, *and let* Σ *consist of the two FDs Emp* $\to$ *Dept and Dept* $\to$ *Mgr. On input* ($\{$*Emp*$\}$,Σ,Employment$_s$,Employment) *Algorithm 1 returns the set* $\{$*Emp*,*Dept*$\}$. *Consequently, the FD Emp* $\to$ *Mgr is not implied by* Σ *in the presence of the NFS* Employment$_s$ *since Mgr is not an element of* $\{$*Emp*$\}^*_{\Sigma,\text{Employment}_s}$ = $\{$*Emp*, *Dept*$\}$. ∎

ALGORITHM 1: NFSClosure(X,Σ,R_s,R)

Input: attribute subset X, FD set Σ, NFS R_s all over relation schema R.
Output: attribute closure X^+_{Σ,R_s} of X with respect to Σ and R_s.

$CLOSURE = X$;
repeat
  $OLDCLOSURE = CLOSURE$;
  for *each* $V \to W \in \Sigma$ **do**
    if $V \subseteq CLOSURE \cap XR_s$ **then**
      $CLOSURE = CLOSURE \cup W$;
    end
  end
until $OLDCLOSURE = CLOSURE$;
return $CLOSURE$;

Note that our results show that Galil's algorithm can also decide any instance $\Sigma \models_{R_s} \varphi$ of the implication problem for FDs in the presence of an NFS R_s, cf. Corollary 4. However, the flaw of the original NFSClosure algorithm [25] has not been pointed out nor corrected previously.

5.3.5 Sagiv's algorithms

Alternative algorithms for computing the dependency basis and deciding implication over total relations were given by Sagiv [63], cf. Section 6. Though Sagiv's approach does not provide better time bounds, it is of interest as it directly exploits the equivalence between the implication of FDs and MVDs over total relations and the logical implication in a fragment of propositional logic. Galil [59] predicts that using this equivalence one may possibly come up with a linear time algorithm to decide implication. This provides strong motivation for investigating the implication of FDs and MVDs in the presence of an NFS from a logical point of view.

6 Equivalences to Boolean and Para-consistent Implication

For the special case where the NFS covers every attribute of the underlying relation schema, Sagiv, Delobel, Parker and Fagin showed an equivalence between the implication of the combined class of FDs and MVDs and the Boolean implication of a propositional fragment $\mathcal{F}$ [64, 21]. In this section, we will establish an equivalence between the implication problem of the combined class of FDs and MVDs in the presence of an arbitrary NFS and the implication problem of the fragment $\mathcal{F}$ in a family $LP_{\mathcal{S}}$ of para-consistent logics. $LP_{\mathcal{S}}$ implication is equivalent to $\mathcal{S}$-3 implication in Cadoli and Schaerf's well-known approximation logic [79]. In fact, the NFS corresponds to the set $\mathcal{S}$ of propositional variables V' which cannot be *paradoxical* (i.e. V' cannot be *true* and *false* at the same time) in $LP_{\mathcal{S}}$ interpretations, or equivalently, where either V' or $\neg V'$ is

Table 3: Truth functions in $LP_{\mathcal{S}}$

$\neg$	
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{P}$	$\mathbb{P}$
$\mathbb{F}$	$\mathbb{T}$

$\wedge$	$\mathbb{T}$	$\mathbb{P}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{P}$	$\mathbb{F}$
$\mathbb{P}$	$\mathbb{P}$	$\mathbb{P}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

$\vee$	$\mathbb{T}$	$\mathbb{P}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{P}$	$\mathbb{T}$	$\mathbb{P}$	$\mathbb{P}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{P}$	$\mathbb{F}$

$\rightarrow$	$\mathbb{T}$	$\mathbb{P}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{P}$	$\mathbb{F}$
$\mathbb{P}$	$\mathbb{T}$	$\mathbb{P}$	$\mathbb{P}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$

true but not both in $\mathcal{S}$-3 interpretations. For Lien's class of FDs and MVDs we obtain an equivalence to LP implication in Graham Priest's well-known Logic of Paradox [80]. For Atzeni and Morfuni's class of FDs in the presence of an NFS we obtain an equivalence to $\mathcal{S}$-3 implication for propositional Horn clauses. In the electronic appendix, we will show how our techniques can be applied to characterize the implication of full first-order hierarchical decompositions, and the implication of Boolean dependencies in the presence of an NFS, respectively.

6.1 The Family $LP_{\mathcal{S}}$ of para-consistent Logics

The proof-theoretic aim of para-consistent logics is to reason about systems that may be inconsistent. Formalisms such as theory change deal with inconsistencies in knowledge bases by avoiding them, and by removing them once they are located. Para-consistent logics reason non-explosively in the presence of inconsistencies. In classical logic, a theory is consistent if and only if it has a model. The trademark of para-consistent logics is that inconsistent theories can have models.

In our family of para-consistent logics a sentence is either true (and not false), denoted by $\mathbb{T}$, or false (and not true), denoted by $\mathbb{F}$, or paradoxical (both true and false), denoted by $\mathbb{P}$. This yields a family of three-valued logics based on Kleene's truth tables (Table 3), but the third truth value indicates that a sentence is paradoxical, as opposed to being undefined or undetermined in strong Kleene logic [88]. Codd [65] suggested the same tables for the null interpretation of "value unknown at present" to extend the relational algebra by means of a three-valued logic and the null substitution principal, cf. Section 7.

Let $\mathcal{L}^*$ denote the *propositional language* over a finite set $\mathcal{L}$ of propositional variables, generated from the unary connective $\neg$ (negation), and the binary connectives $\wedge$ (conjunction) and $\vee$ (disjunction). $\mathcal{L}^*$ is the smallest set that satisfies:

1. $\mathcal{L} \subseteq \mathcal{L}^*$,
2. if $\varphi' \in \mathcal{L}^*$, then $(\neg\varphi') \in \mathcal{L}^*$, and
3. if $\varphi'_1, \varphi'_2 \in \mathcal{L}^*$, then $(\varphi'_1 \vee \varphi'_2) \in \mathcal{L}^*$ and $(\varphi'_1 \wedge \varphi'_2) \in \mathcal{L}^*$.

For convenience, we also use the binary connective $\rightarrow$ (implication) defined by $\varphi'_1 \rightarrow \varphi'_2 := (\neg\varphi'_1) \vee \varphi'_2$. The corresponding truth table is shown in Table 3. We assume that negation binds stronger than conjunction and disjunction, and that conjunction and disjunction bind stronger than implication. We omit parentheses if it does not cause ambiguity.

We may call the formulae of $\mathcal{L}^*$ also $\mathcal{L}$-formulae. We denote variables with upper-case accented Latin letters, e.g. A', B', C', or subscripted as A'_1, A'_2, A'_3. Note that we use accented versions to denote the correspondence between propositional variables and attributes, e.g. the propositional variable A' corresponds to the attribute A. Elements of $\mathcal{L}^*$ are denoted by lower-case Greek letters such as φ', σ', ψ', or their subscripted version, and subsets of $\mathcal{L}^*$ are denoted by the upper-case Greek letter Σ'.

An *LP interpretation* ω' of $\mathcal{L}$ is a total function from $\mathcal{L}$ to the set of truth values $\{\mathbb{F}, \mathbb{P}, \mathbb{T}\}$. For $\mathcal{S} \subseteq \mathcal{L}$, an *LP interpretation* ω' of $\mathcal{L}$ is an *$LP_\mathcal{S}$ interpretation*, if for all $A' \in \mathcal{S}$ we have $\omega'(A') \in \{\mathbb{F}, \mathbb{T}\}$. A variable in $\mathcal{S}$ is said to be *paradox-free*.

The semantics of an $\mathcal{L}$-formula φ' in an *LP* interpretation ω' of $\mathcal{L}$ is defined in the usual compositional way given by the truth tables in Table 3. That is, we can extend ω' to a total function $\Omega' : \mathcal{L}^* \to \{\mathbb{F}, \mathbb{P}, \mathbb{T}\}$ as follows:

1. $\Omega'(A') := \omega'(A')$ for all $A' \in \mathcal{L}$,
2. $\Omega'(\neg\varphi') := \neg\Omega'(\varphi')$,
3. $\Omega'(\varphi' \wedge \psi') := \Omega'(\varphi') \wedge \Omega'(\psi')$, and
4. $\Omega'(\varphi' \vee \psi') := \Omega'(\varphi') \vee \Omega'(\psi')$.

As usual, the left-hand sides of the definitions contain the symbols that denote the connectives that generate $\mathcal{L}^*$ from $\mathcal{L}$, whereas the right-hand sides of the definitions contain the symbols that denote the semantic truth functions defined in Table 3.

When working with more than two truth values, one has to define the set of designated values. For our family of $LP_\mathcal{S}$ logics we have $\{\mathbb{P}, \mathbb{T}\}$ as the set of designated truth values since a paradoxical formula is true (and false).

An *LP* interpretation ω' is a *model* of a set Σ' of $\mathcal{L}$-formulae, denoted by $\models_{\omega'} \Sigma'$, if and only if for all $\sigma' \in \Sigma'$ we have $\Omega'(\sigma') \in \{\mathbb{P}, \mathbb{T}\}$. For $\mathcal{S} \subseteq \mathcal{L}$, we say that Σ' *$LP_\mathcal{S}$ implies* an $\mathcal{L}$-formula φ', denoted by $\Sigma' \models_{LP_\mathcal{S}} \varphi'$, if and only if every $LP_\mathcal{S}$ interpretation that is a model of Σ' is also a model of φ'. For the special case where $\mathcal{S} = \emptyset$ we may simply speak of an *LP* interpretation or *LP* model, respectively.

Let $\Sigma' = \{A' \to B', B' \to C'\}$ and $\varphi' = A' \to C'$. The *LP* interpretation ω' that maps A' to $\mathbb{T}$, B' to $\mathbb{P}$ and C' to $\mathbb{F}$ shows that Σ' does not *LP* imply φ', i.e., $\Sigma' \not\models_{LP} \varphi'$. However, it is not difficult to see that Σ' $LP_{\{B'\}}$ implies φ'.

LP is distinguished from classical logic by the invalidity of the *Modus Ponens*, e.g. from A' and $A' \to B'$ one may not conclude B': assign $\mathbb{F}$ to B' and $\mathbb{P}$ to A'.

6.2 Equivalences to $LP_\mathcal{S}$ implication

In a first step, we define the fragment of $\mathcal{L}$-formulae that corresponds to FDs and MVDs in the presence of an NFS R_s over a relation schema R. Let $\phi : R \to \mathcal{L}$ denote a bijection between R and the set $\mathcal{L} = \{A' \mid A \in R\}$ of propositional variables that corresponds to R. For an NFS R_s over R let $\mathcal{S} = \phi(R_s)$ be the set of propositional variables in $\mathcal{L}$ that corresponds to R_s. Hence, the paradox-free variables of $\mathcal{L}$ are the images of those attributes of R declared `NOT NULL`.

We now extend ϕ to a mapping Φ from the set of FDs and MVDs over R to the set $\mathcal{L}^*$. For an attribute set $X = \{A_1, \ldots, A_n\}$ we write $\bigwedge_{A \in X} A'$ as a shortcut for $A'_1 \wedge \cdots \wedge A'_n$. For an FD $X \to B$ over R, let

$$\Phi(X \to B) := \left(\bigwedge_{A \in X} A' \right) \to B'.$$

For the sake of presentation, but without loss of generality, assume that FDs have only a single attribute on their right-hand side. For an MVD $X \twoheadrightarrow Y$ over R, let

$$\Phi(X \twoheadrightarrow Y) := \left(\bigwedge_{A \in X} A' \right) \to \left(\left(\bigwedge_{B \in Y - X} B' \right) \vee \left(\bigwedge_{C \in R - XY} C' \right) \right).$$

As usual, disjunctions over zero disjuncts are interpreted as $\mathbb{F}$ and conjunctions over zero conjuncts are interpreted as $\mathbb{T}$. In what follows, we may simply denote $\Phi(\varphi) = \varphi'$ and $\Phi(\Sigma) = \{\sigma' \mid \sigma \in \Sigma\} = \Sigma'$.

Example 6 *Let $R = ASLC$ denote the relation schema* Supplies*, $R_s = ALC$ and let Σ contain the FDs $\sigma_1 = A \to S$ and $\sigma_2 = AL \to C$, and the MVD $\sigma_3 = S \twoheadrightarrow L$. Let $\varphi_1 = A \twoheadrightarrow L$ and $\varphi_2 = A \to C$. Then $\mathcal{L} = \{A', S', L', C'\}$, $\mathcal{S} = \{A', L', C'\}$, $\Sigma' = \{\sigma'_1, \sigma'_2, \sigma'_3\}$ with $\sigma'_1 = A' \to S'$, $\sigma'_2 = A' \wedge L' \to C'$, and $\sigma'_3 = S' \to L' \vee (A' \wedge C')$, as well as $\varphi'_1 = A' \to L' \vee (S' \wedge C')$ and $\varphi'_2 = A' \to C'$.* ∎

Our aim is to show that for every relation schema R, for every FD and MVD set $\Sigma \cup \{\varphi\}$ and for every NFS R_s over R, there is some R_s-total relation r that satisfies Σ and violates φ if and only if there is an $LP_{\mathcal{S}}$ model ω'_r of Σ' that is not an $LP_{\mathcal{S}}$ model of φ'. For arbitrary finite relations r it is not obvious how to define the $LP_{\mathcal{S}}$ interpretation ω'_r.

Corollary 1 tells us that for deciding the implication problem $\Sigma \models_{R_s} \varphi$ it suffices to examine two-tuple relations (instead of arbitrary finite relations). For two-tuple relations $\{t_1, t_2\}$, however, we can define a corresponding LP interpretation $\omega'_{\{t_1,t_2\}}$. For this purpose, we introduce an extension of the notion of *agree sets* of distinct tuples to the presence of null markers [89, 29]. For two tuples t_1, t_2 over relation schema R we define

$$\begin{array}{lcl} ag^s(t_1, t_2) & = & \{A \in R \mid t_1(A) = t_2(A) \text{ and } t_1(A) \neq \texttt{ni} \neq t_2(A)\}, \\ ag^w(t_1, t_2) & = & \{A \in R \mid t_1(A) = \texttt{ni} = t_2(A)\}, \\ ag(t_1, t_2) & = & ag^s(t_1, t_2) \cup ag^w(t_1, t_2)\ . \end{array}$$

If $A \in ag^s(t_1, t_2)$ we say that t_1 and t_2 agree *strongly* on A. If $A \in ag^w(t_1, t_2)$ we say that t_1 and t_2 agree *weakly* on A, and if $A \notin ag(t_1, t_2)$ we say that t_1 and t_2 *disagree* on A.

We now define the *special LP interpretation*: for two tuples t_1, t_2 over the relation schema R let $\omega'_{\{t_1,t_2\}}$ denote the following LP interpretation of $\mathcal{L}$:

$$\omega'_{\{t_1,t_2\}}(A') = \begin{cases} \mathbb{T} & \text{, if } A \in ag^s(t_1, t_2) \\ \mathbb{P} & \text{, if } A \in ag^w(t_1, t_2) \\ \mathbb{F} & \text{, if } A \notin ag(t_1, t_2) \end{cases}.$$

Note that the special LP interpretation is a generalization of Fagin's special truth assignment: for all $\mathcal{S} \subseteq \mathcal{L}$ and for all variables $A' \in \mathcal{S}$ we have $\omega'_{\{t_1,t_2\}}(A') = \mathbb{T}$ if and only if $t_1(A) = t_2(A)$ [64]. The following lemma shows that the special LP interpretation is an $LP_{\mathcal{S}}$ interpretation whenever $\{t_1, t_2\}$ is R_s-total.

Lemma 3 *Let R be some relation schema, r a two-tuple relation over R and R_s an NFS over R. Let $\mathcal{L}$ be the set of propositional variables that corresponds to R, and $\mathcal{S}$ the set of propositional variables that corresponds to R_s. If r satisfies R_s, then ω'_r is an $LP_{\mathcal{S}}$ interpretation of $\mathcal{L}$, i.e., for all $A' \in \mathcal{S}$ it is true that $\omega'_r(A') \neq \mathbb{P}$.*

Proof If r satisfies R_s, then the two tuples of r are R_s-total. According to the definition of the special LP interpretation ω'_r it cannot be the case that $\omega'_r(A') = \mathbb{P}$ for any $A' \in \mathcal{S}$. ∎

The converse of Lemma 3 is not valid. In fact, let $R = AB$ and $R_s = A$, and let $r = \{(a, b), (\mathtt{ni}, b')\}$. We have $\omega'_r(A') = \mathbb{F} = \omega'_r(B')$, but r violates $R_s = A$. However, note that we can replace the null marker occurrence $\mathtt{ni}$ in r by a non-null marker different from a. The resulting relation r' satisfies $R_s = A$ and $\omega'_{r'} = \omega'_r$. This strategy is always applicable.

Lemma 4 *Let R be some relation schema, and R_s an NFS over R. Let $\mathcal{L}$ be the set of propositional variables that corresponds to R, and $\mathcal{S}$ the set of propositional variables that corresponds to R_s. Let ω' be an $LP_{\mathcal{S}}$ interpretation of $\mathcal{L}$, i.e. for all $A' \in \mathcal{S}$ it is true that $\omega'(A') \neq \mathbb{P}$. Then there is a two-tuple relation r over R such that r satisfies R_s and $\omega'_r = \omega'$.*

Proof Define a tuple t over R as follows: for $A \in R$ let $t_1(A) := a \in dom(A) - \{\mathtt{ni}\}$, if $\omega'(A') \neq \mathbb{P}$, and $t_1(A) := \mathtt{ni}$ otherwise. Define another tuple t_2 over R as follows: i) let $t_2(A) := t_1(A)$, if $\omega'(A') \neq \mathbb{F}$, and ii) let $t_2(A) := a' \in dom(A) - \{\mathtt{ni}, t_1(A)\}$ otherwise. Let $r := \{t_1, t_2\}$. From this definition it follows that r is R_s-total. Moreover, for all $A' \in \mathcal{L}$ we have $\omega'_r(A') = \omega'(A')$. ∎

The following lemma justifies the definitions of the corresponding fragment of $\mathcal{L}$-formulae and the special LP interpretation of $\mathcal{L}$.

Lemma 5 *Let r be a two-tuple relation over relation schema R, and let φ denote an FD or MVD over R. Then r satisfies φ if and only if ω'_r is an LP model of φ'.*

Proof Let $r = \{t_1, t_2\}$.

Sufficiency. Assume that ω'_r is an LP model of φ'. We show that r satisfies φ.

First, let φ denote the FD $X \to B$ where $X = \{A_1, \ldots, A_n\}$. If $t_1[X] = t_2[X]$ and t_1, t_2 are X-total, then $\omega'_r(A'_i) = \mathbb{T}$ for all $i = 1, \ldots, n$. Since ω'_r is an LP model of φ' it follows that $\omega'_r(B) \neq \mathbb{F}$. Hence, $B \in ag(t_1, t_2)$. That is, r satisfies φ.

Now let φ denote the MVD $X \twoheadrightarrow Y$ where $X = \{A_1, \ldots, A_n\}$, $Y - X = \{B_1, \ldots, B_m\}$ and $R - XY = \{C_1, \ldots, C_l\}$. If $t_1[X] = t_2[X]$ and t_1, t_2 are X-total, then $\omega'_r(A'_i) = \mathbb{T}$ for all $i = 1, \ldots, n$. Since ω'_r is an LP model of φ' it follows that not both $B'_1 \wedge \cdots \wedge B'_m$

and $C'_1 \wedge \cdots \wedge C'_l$ are $\mathbb{F}$ under ω'_r. That is, if $\omega'_r(B'_j) = \mathbb{F}$ for some $1 \leq j \leq m$, then for all $k = 1, \ldots, l$ we have $\omega'_r(C'_k) \neq \mathbb{F}$. Hence, for all $k = 1, \ldots, l$ we have $C_k \in ag(t_1, t_2)$. Therefore, r satisfies $X \twoheadrightarrow Y$.

Necessity. Assume that r satisfies φ. We show that ω'_r is an LP model of φ'.

First, let φ' denote the formula $\neg A'_1 \vee \cdots \vee \neg A'_n \vee B$. Suppose that $\omega'_r(A'_i) = \mathbb{T}$ for all $i = 1, \ldots, n$ (otherwise ω'_r is an LP model of φ'). It follows that $t_1[X] = t_2[X]$ and t_1, t_2 are X-total. Since r satisfies φ it follows that $B \in ag(t_1, t_2)$. Consequently, $\omega'_r(B) \neq \mathbb{F}$. Therefore, ω'_r is an LP model of φ'.

Now let φ' denote the formula

$$\neg A'_1 \vee \cdots \vee \neg A'_n \vee (B'_1 \wedge \cdots \wedge B'_m) \vee (C'_1 \wedge \cdots \wedge C'_l).$$

Suppose that $\omega'_r(A'_i) = \mathbb{T}$ for all $i = 1, \ldots, n$ (otherwise ω'_r is an LP model of φ'). It follows that $t_1[X] = t_2[X]$ and t_1, t_2 are X-total. Since r satisfies φ it follows that for all $j = 1, \ldots, m$ we have $B_j \in ag(t_1, t_2)$ or for all $k = 1, \ldots, l$ we have $C_k \in ag(t_1, t_2)$. Hence, not both $B'_1 \wedge \cdots \wedge B'_m$ and $C'_1 \wedge \cdots \wedge C'_m$ evaluate to $\mathbb{F}$ under ω'_r. Consequently, ω'_r is an LP model of φ'. ∎

In fact, Corollary 1 and Lemmata 3, 4 and 5 allow us to establish the anticipated equivalence between the implication of FDs and MVDs in the presence of an NFS R_s and the $LP_{\mathcal{S}}$ implication of their corresponding fragment of $\mathcal{L}$-formulae.

Theorem 3 *Let $\Sigma \cup \{\varphi\}$ be a set of FDs and MVDs, and let R_s be an NFS over some relation schema R. Let $\mathcal{L}$ be the set of propositional variables that corresponds to R, $\mathcal{S}$ the set of propositional variables that corresponds to R_s, and let $\Sigma' \cup \{\varphi'\}$ denote the set of $\mathcal{L}$-formulae that corresponds to $\Sigma \cup \{\varphi\}$. Then $\Sigma \models_{R_s} \varphi$ if and only if $\Sigma' \models_{LP_{\mathcal{S}}} \varphi'$.*

Proof According to Corollary 1 it suffices to show that $\Sigma \models_{2,R_s} \varphi$ if and only if $\Sigma' \models_{LP_{\mathcal{S}}} \varphi'$.

We show first that if $\Sigma' \models_{LP_{\mathcal{S}}} \varphi'$ holds, then $\Sigma \models_{2,R_s} \varphi$ holds, too. For this purpose, suppose that $\Sigma \models_{2,R_s} \varphi$ does not hold. Consequently, there is some two-tuple relation r over R that satisfies Σ and R_s but violates φ. Following Lemma 3, ω'_r is an $LP_{\mathcal{S}}$ interpretation. According to Lemma 5, ω'_r is an $LP_{\mathcal{S}}$ model of Σ' but not an $LP_{\mathcal{S}}$ model of φ'. Consequently, $\Sigma' \models_{LP_{\mathcal{S}}} \varphi'$ does also not hold.

It now remains to show that if $\Sigma \models_{2,R_s} \varphi$ holds, then $\Sigma' \models_{LP_{\mathcal{S}}} \varphi'$ holds, too. For this purpose, suppose that $\Sigma' \models_{LP_{\mathcal{S}}} \varphi'$ does not hold. Consequently, there is some $LP_{\mathcal{S}}$ interpretation ω' of $\mathcal{L}$ that is a model of Σ' but not a model of φ'. According to Lemma 4 there is some two-tuple relation r that satisfies R_s and $\omega'_r = \omega'$. In particular, since ω' violates φ' the construction in the proof of Lemma 4 ensures that r is subsumption-free. Since $\omega'_r = \omega'$, Lemma 5 guarantees that r satisfies Σ but violates φ. We conclude that $\Sigma \models_{2,R_s} \varphi$ does also not hold. ∎

Example 7 *Let $R = ASLC$ denote the relation schema* Supplies, *$R_s = ALC$ and let Σ contain the FDs $A \to S$ and $AL \to C$, and the MVD $S \twoheadrightarrow L$. The following relation r*

Article	Supplier	Location	Cost
Kiwi	***ni***	*Maunganui*	*1.50*
Kiwi	***ni***	*Taranaki*	*2.50*

shows that Σ implies neither the MVD $\varphi_1 = A \twoheadrightarrow L$ nor the FD $\varphi_2 = A \rightarrow C$ in the presence of R_s. For ω'_r we obtain $\omega'_r(A') = \mathbb{T}$, $\omega'_r(S') = \mathbb{P}$, $\omega'_r(L') = \mathbb{F}$ and $\omega'_r(C') = \mathbb{F}$. Indeed, ω'_r is an $LP_{\{A',L',C'\}}$ model of Σ' but not an $LP_{\{A',L',C'\}}$ model of neither φ'_1 nor φ'_2. ∎

Remark 4 *Note that the proof of Theorem 3 is not a simple extension of the proof for the special case where $R_s = R$ [21]. In particular, we utilize the two-tuple relation from the proof of Theorem 2, whereas Sagiv, Delobel, Parker and Fagin show that every relation that satisfies Σ and violates φ contains a two-tuple subrelation that satisfies Σ and violates φ. While this is a stronger result, our proof shows that this result is not necessary to establish the equivalence to the implication in logical fragments, in particular not for the special case where $R_s = R$. However, we do extend the stronger result to the general case of arbitrary null-free subschemata in the electronic appendix.* ∎

6.3 Equivalences to *LP* and $\mathcal{S}$-3 implication

6.3.1 Logic of Paradox

Priest [80] introduced the Logic of Paradox *LP* as "a new way of handling the logical paradoxes". Priest argues that "the most satisfactory account of the paradoxes is to view as what they appear, prima facie, to be true contradictions, i.e., sentences such that both they and their negations are true". Our family of $LP_{\mathcal{S}}$ logics subsumes the logic *LP* as the special case where $\mathcal{S}$ is the empty set of variables.

6.3.2 $\mathcal{S}$-3 Logics

Schaerf and Cadoli [79] introduced $\mathcal{S}$-3 logics as "a semantically well-founded logical framework for sound approximate reasoning, which is justifiable from the intuitive point of view, and to provide fast algorithms for dealing with it even when using expressive languages". For a finite set $\mathcal{L}$ of propositional variables let $\mathcal{L}^\ell$ denote the set of all literals over $\mathcal{L}$, i.e., $\mathcal{L}^\ell = \mathcal{L} \cup \{\neg A' \mid A' \in \mathcal{L}\} \subseteq \mathcal{L}^*$. Let $\mathcal{S} \subseteq \mathcal{L}$. An $\mathcal{S}$-3 interpretation of $\mathcal{L}$ is a total function $\hat{\omega} : \mathcal{L}^\ell \rightarrow \{\mathbb{F}, \mathbb{T}\}$ that maps every variable $A' \in \mathcal{S}$ and its negation $\neg A'$ into opposite values ($\hat{\omega}(A') = \mathbb{T}$ if and only if $\hat{\omega}(\neg A') = \mathbb{F}$), and that does not map both a variable $A' \in \mathcal{L} - \mathcal{S}$ and its negation $\neg A'$ into $\mathbb{F}$ (we must not have $\hat{\omega}(A') = \mathbb{F} = \hat{\omega}(\neg A')$ for any $A' \in \mathcal{L} - \mathcal{S}$). Accordingly, for each variable $A' \in \mathcal{L}$ and each $\mathcal{S}$-3 interpretation $\hat{\omega}$ of $\mathcal{L}$ there are the following possibilities:

- $\hat{\omega}(A') = \mathbb{T}$ and $\hat{\omega}(\neg A') = \mathbb{F}$,
- $\hat{\omega}(A') = \mathbb{F}$ and $\hat{\omega}(\neg A') = \mathbb{T}$,
- $\hat{\omega}(A') = \mathbb{T}$ and $\hat{\omega}(\neg A') = \mathbb{T}$ (only if $A' \in \mathcal{L} - S$).

$\mathcal{S}$-3 interpretations generalize both standard 2-valued interpretations and the 3 interpretations of Levesque [90]. That is, a 2-valued interpretation is an $\mathcal{S}$-3 interpretation where $\mathcal{S} = \mathcal{L}$, while a 3 interpretation is an $\mathcal{S}$-3 interpretation with $\mathcal{S} = \emptyset$. Hence, in $\mathcal{S}$-3 interpretations the situation where A' and $\neg A'$ are both true corresponds to the situation in $LP_{\mathcal{S}}$ interpretations where A' is $\mathbb{P}$.

An $\mathcal{S}$-3 interpretation $\hat{\omega} : \mathcal{L}^{\ell} \to \{\mathbb{F}, \mathbb{T}\}$ of $\mathcal{L}$ can be lifted to a total function $\hat{\Omega} : \mathcal{L}^{*} \to \{\mathbb{F}, \mathbb{T}\}$. This lifting has been defined as follows [79]. An arbitrary formula φ' in $\mathcal{L}^{*}$ is firstly converted (in linear time in the size of the formula) into its corresponding formula φ'_N in *Negation Normal Form* (NNF) using the following rewriting rules: $\neg(\varphi' \wedge \psi') \mapsto (\neg\varphi' \vee \neg\psi')$, $\neg(\varphi' \vee \psi') \mapsto (\neg\varphi' \wedge \neg\psi')$, and $\neg(\neg\varphi') \mapsto \varphi'$. Therefore, negation in a formula in NNF occurs only at the literal level. The rules for assigning truth values to NNF formulae are as follows:

- $\hat{\Omega}(\varphi') = \hat{\omega}(\varphi')$, if $\varphi' \in \mathcal{L}^{\ell}$,
- $\hat{\Omega}(\varphi' \vee \psi') = \mathbb{T}$ if and only if $\hat{\Omega}(\varphi') = \mathbb{T}$ or $\hat{\Omega}(\psi') = \mathbb{T}$,
- $\hat{\Omega}(\varphi' \wedge \psi') = \mathbb{T}$ if and only if $\hat{\Omega}(\varphi') = \mathbb{T}$ and $\hat{\Omega}(\psi') = \mathbb{T}$.

An $\mathcal{S}$-3 interpretation $\hat{\omega}$ is a *model* of a set Σ' of $\mathcal{L}$-formulae if and only if $\hat{\Omega}(\sigma'_N) = \mathbb{T}$ holds for every $\sigma' \in \Sigma'$. We say that Σ' *$\mathcal{S}$-3 implies* an $\mathcal{L}$-formula φ', denoted by $\Sigma' \models^{3}_{S} \varphi'$, if and only if every $\mathcal{S}$-3 interpretation that is a model of Σ' is also a model of φ'. $\mathcal{S}$-3 interpretations are related closely to $LP_{\mathcal{S}}$ interpretations.

Proposition 1 *Let $\mathcal{S} \subseteq \mathcal{L}$ and $\omega' : \mathcal{L} \to \{\mathbb{F}, \mathbb{P}, \mathbb{T}\}$ be an $LP_{\mathcal{S}}$ interpretation of $\mathcal{L}$, i.e. for all $A' \in \mathcal{S}$, $\omega'(A') \neq \mathbb{P}$. Then we can associate in a bijective way an $\mathcal{S}$-3 interpretation $\hat{\omega}' : \mathcal{L}^{\ell} \to \{\mathbb{F}, \mathbb{T}\}$, i.e. for all $A' \in \mathcal{L}$ we never have $\hat{\omega}'(A') = \mathbb{F} = \hat{\omega}'(\neg A')$, and for all $A' \in \mathcal{S}$, $\hat{\omega}'(A') \neq \hat{\omega}'(\neg A')$, where $\hat{\omega}'$ is:*

- *$\hat{\omega}'(A') = \mathbb{T}$ and $\hat{\omega}'(\neg A') = \mathbb{F}$ if and only if $\omega'(A') = \mathbb{T}$,*
- *$\hat{\omega}'(A') = \mathbb{F}$ and $\hat{\omega}'(\neg A') = \mathbb{T}$ if and only if $\omega'(A') = \mathbb{F}$,*
- *$\hat{\omega}'(A') = \mathbb{T}$ and $\hat{\omega}'(\neg A') = \mathbb{T}$ if and only if $\omega'(A') = \mathbb{P}$.*

For all formulae $\varphi' \in \mathcal{L}^{*}$ we have $\hat{\Omega}'(\varphi'_N) = \mathbb{T}$ if and only if $\Omega'(\varphi') \in \{\mathbb{P}, \mathbb{T}\}$. ∎

Proposition 1 and Theorem 3 establish the equivalence between the implication of FDs and MVDs in the presence of an NFS R_s over a relation schema R, and the $\mathcal{S}$-3 implication of the corresponding fragment of $\mathcal{L}$-formulae.

Corollary 6 *Let $\Sigma \cup \{\varphi\}$ be a set of FDs and MVDs over the relation schema R, and let R_s denote an NFS over R. Let $\mathcal{L}$ denote the set of propositional variables that corresponds to R, $\mathcal{S}$ the set of variables that corresponds to R_s, and $\Sigma' \cup \{\varphi'\}$ the set of $\mathcal{L}$-formulae that corresponds to $\Sigma \cup \{\varphi\}$. Then $\Sigma \models_{R_s} \varphi$ if and only if $\Sigma' \models^{3}_{\mathcal{S}} \varphi'$.* ∎

We may also define the *special*-3-*interpretation*: for two tuples t_1, t_2 over the relation schema R let $\hat{\omega}'_{\{t_1,t_2\}}$ denote the following 3-interpretation of $\mathcal{L}$: for all $A \in R$, if $t_1(A) = t_2(A)$ and $t_1(A) \neq \texttt{ni} \neq t_2(A)$, then $\hat{\omega}'_{\{t_1,t_2\}}(A') = \mathbb{T}$ and $\hat{\omega}'_{\{t_1,t_2\}}(\neg A') = \mathbb{F}$, if $t_1(A) = \texttt{ni} = t_2(A)$, then $\hat{\omega}'_{\{t_1,t_2\}}(A') = \mathbb{T}$ and $\hat{\omega}'_{\{t_1,t_2\}}(\neg A') = \mathbb{T}$, and if $t_1(A) \neq t_2(A)$, then $\hat{\omega}'_{\{t_1,t_2\}}(A') = \mathbb{F}$ and $\hat{\omega}'_{\{t_1,t_2\}}(\neg A') = \mathbb{T}$. In particular, if $\{t_1, t_2\}$ is R_s-total, then $\hat{\omega}'_{\{t_1,t_2\}}$ is an $\mathcal{S}$-3 interpretation. The following example continues Example 7.

Example 8 *Let $R = ASLC$ denote the relation schema* Supplies, *$R_s = ALC$ and let Σ contain the FDs $A \to S$ and $AL \to C$, and the MVD $S \twoheadrightarrow L$. The following relation r*

Article	Supplier	Location	Cost
Kiwi	***ni***	*Maunganui*	*1.50*
Kiwi	***ni***	*Taranaki*	*2.50*

shows that Σ implies neither the MVD $\varphi_1 = A \twoheadrightarrow L$ nor the FD $\varphi_2 = A \to C$ in the presence of R_s. For $\hat{\omega}'_r$ we obtain $\hat{\omega}'_r(V) = \mathbb{T}$, if $V \in \{A', S', \neg S', \neg L', \neg C'\}$, and $\hat{\omega}'_r(V) = \mathbb{F}$, if $V \in \{\neg A', L', C'\}$. Indeed, $\hat{\omega}'_r$ is an $\{A', L', C'\}$-3 model of Σ' but neither an $\{A', L', C'\}$-3 model of φ'_1 nor φ'_2. ∎

Remark 5 *Cadoli and Schaerf [79] have shown that $\mathcal{S}$-3 entailment is equivalent to $\mathcal{S}$-3 unsatisfiability. Furthermore, they have reduced tests for $\mathcal{S}$-3 satisfiability to tests for Boolean satisfiability. Therefore, classic algorithms for satisfiability like Davis and Putnam's [91] or Robinson's [92] can be applied to $\mathcal{S}$-3 entailment, and, by our results, to decide implication of FDs and MVDs in the presence of an NFS. Vice versa, our algorithms for deciding implication of FDs and MVDs in the presence of an NFS can be applied to decide $\mathcal{S}$-3 entailment and $\mathcal{S}$-3 satisfiability of the corresponding fragments of propositional formulae. In particular, Dowling and Gallier's unit propagation algorithm [93], developed to decide entailment of Horn clauses, can be applied to decide the implication of FDs in the presence of an NFS. Vice versa, Algorithm 1 for the computation of attribute set closures can be applied to decide the $\mathcal{S}$-3 entailment of Horn clauses.* ∎

Finally, we exemplify that our framework allows very general reasoning about key and uniqueness constraints.

Example 9 *Let* Supplier $= ASLC$, Supplier$_{\mathcal{S}} = SLC$ *and $\Sigma = \{A \to S, AL \to C, AC \to L, S \twoheadrightarrow L\}$. It follows that $A \to SLC$ is implied by Σ in the presence of* Supplier$_{\mathcal{S}}$. *Hence, A is a uniqueness constraint, i.e., every non-null marker in the A-column is unique in the A-column (there can still be distinct rows which are both null on A). If we also declare A to be* ***NOT NULL*** *and Σ is enforced by the database management system, then A is even a candidate key. That is, for every table over* Supplier *every row in that table has a total and unique value in the A-column.* ∎

6.3.3 Sagiv, Delobel, Parker and Fagin's class of FDs and MVDs over total relations

The results for this class of data dependencies [21] are subsumed by our theory for the special case where $R_s = R$. This is an important special case since there is a direct equivalence to Boolean implication. In particular, Fagin's special truth assignment [64] applies to all the variables in $\mathcal{S}$, i.e., the variables that correspond to attributes of R declared `NOT NULL`.

6.3.4 Lien's class of FDs and MVDs

Theorem 3 subsumes equivalences for Lien's class of FDs and MVDs where no NFS R_s is assumed to be given [23]. In fact, we obtain an equivalence to the *LP* implication of the propositional fragment in Priest's Logic of Paradox [80], which itself corresponds to the special case $\mathcal{S} = \emptyset$ in Cadoli and Schaerf's family of $\mathcal{S}$-3 logics [79]. The arguments that result in Theorem 3 also show that if the given set of data dependencies consists of FDs only, then we obtain an equivalence to the Horn fragment of these logics.

6.3.5 Atzeni and Morfuni's class of FDs in the presence of an NFS

Finally, Theorem 3 subsumes equivalences for Atzeni and Morfuni's class of FDs in the presence of an arbitrary NFS R_s [25]. In this case, we obtain an equivalence to $\mathcal{S}$-3 implication of propositional Horn formulae.

6.4 Equivalences to Boolean implication

For the special case where the set $\mathcal{S}$ is the full underlying set $\mathcal{L}$ of propositional variables all variables are interpreted classically. If we want to emphasize the fact that we speak about the implication of classical Boolean propositional logic we use $\models_{BL}$ to denote the entailment relation $\models^3_{\mathcal{L}}$. Recall that by Corollary 5 we have that $\Sigma \models_{R_s} \varphi$ if and only if $\Sigma[lhs(\varphi)R_s] \models_R \varphi$. However, $\Sigma[lhs(\varphi)R_s] \models_R \varphi$ holds if and only if $(\Sigma[lhs(\varphi)R_s])' \models_{BL} \varphi'$ holds [21].

Corollary 7 *Let $\Sigma \cup \{\varphi\}$ be a set of FDs and MVDs over the relation schema R, and let R_s denote an NFS over R. Let $\mathcal{L}$ denote the set of propositional variables that corresponds to R, and $\Sigma' \cup \{\varphi'\}$ the set of $\mathcal{L}$-formulae that corresponds to $\Sigma \cup \{\varphi\}$. Then $\Sigma \models_{R_s} \varphi$ if and only if $(\Sigma[lhs(\varphi)R_s])' \models_{BL} \varphi'$.* ∎

Example 10 *Let $R = ASLC$ denote the relation schema* Supplies*, $R_s = ALC$ and let Σ contain the FDs $A \to S$ and $AL \to C$, and the MVD $S \twoheadrightarrow L$. Let φ_1 denote the MVD $A \twoheadrightarrow L$, and let φ_2 denote the FD $A \to C$. The problems whether Σ implies φ_1 and φ_2 in the presence of R_s are equivalent to the problems whether $\Sigma[ALC] = \{A \to S, AL \to C\}$ implies φ_1 and φ_2 over total relations, respectively. The following relation r*

Article	Supplier	Location	Cost
Kiwi	*G6Kiwi*	*Gisborne*	*1.50*
Kiwi	*G6Kiwi*	*Wellington*	*2.50*

shows that $\Sigma[ALC]$ implies neither φ_1 nor φ_2. For ω'_r we obtain $\omega'_r(V) = \mathbb{T}$, if $V \in \{A', S'\}$, and $\omega'_r(V) = \mathbb{F}$, if $V \in \{L', C'\}$. Indeed, ω'_r is a Boolean model of $\Sigma[ALC]'$ but not a Boolean model of neither φ'_1 nor φ'_2. ∎

6.5 Summary of equivalences

Finally, we will give a summary of the characterizations we have established for the implication problem of functional and multivalued dependencies in the presence of a null-free subschema.

Theorem 4 *Let $\Sigma \cup \{\varphi\}$ be a set of FDs and MVDs over the relation schema R, and let R_s denote an NFS over R. Let $\mathcal{L}$ denote the set of propositional variables that corresponds to R, $\mathcal{S}$ the set of variables that corresponds to R_s, and $\Sigma' \cup \{\varphi'\}$ the set of $\mathcal{L}$-formulae that corresponds to $\Sigma \cup \{\varphi\}$. Then the following are equivalent:*

1. $\Sigma \models_{R_s} \varphi$
2. $\Sigma \models_{2,R_s} \varphi$ *(Corollary 1)*
3. $\Sigma \vdash_{\mathfrak{D}} \varphi$ *(Theorem 2)*
4. $\Sigma[lhs(\varphi)R_s] \models_R \varphi$ *(Corollary 5)*
5. $\Sigma[lhs(\varphi)R_s] \models_{2,R} \varphi$ *[21]*
6. $(\Sigma[lhs(\varphi)R_s])' \models_{BL} \varphi'$ *[21]*
7. $\Sigma' \models_{LP_{\mathcal{S}}} \varphi'$ *(Theorem 3)*
8. $\Sigma' \models^3_{\mathcal{S}} \varphi'$ *(Corollary 6).* ∎

7 Impact on other approaches

As pointed out in Section 2 there are several other approaches to handle incomplete information. In this section, we demonstrate how slight adjustments to the notions in the "no information" context result in the applicability of our results to Codd's null marker interpretation "value unknown at present" [65] under Levene and Loizou's weak possible world semantics [41]. We further demonstrate that the results do not apply in this form to Imielinksi's or-relations under Levene and Loizou's weak possible world semantics [68].

7.1 Value unknown at present

Codd's original proposal [65] to handle incomplete information suggested the addition to the database domains of a null marker `unk`, whose meaning is "value unknown at present". Following Codd's proposal, incomplete information is represented in SQL by using `unk` as

a distinguished null marker [17]. We will discuss in this section how the results from the previous sections carry over to this approach towards handling incomplete information.

Levene and Loizou introduced and axiomatized strong and weak FDs (WFDs) with respect to a possible world semantics [41]. We will start by summarizing their approach towards defining WFDs. For this purpose, we assume that the domains of all attributes contain the distinguished element `unk` (and no longer the distinguished element `ni`). With this change in mind, we re-apply the definitions of an X-total tuple and relation as before. The set of all *possible worlds* relative to a relation r over R, denoted by $Poss(r)$, is defined by

$$\begin{aligned} Poss(r) \;:=\; & \{s \mid s \text{ is a relation over } R \text{ and there is a total and onto mapping} \\ & f : r \to s \text{ where } \forall t \in r, t \text{ is subsumed by } f(t) \text{ and } f(t) \text{ is } R\text{-total}\}. \end{aligned}$$

This definition of possible worlds embodies the *closed world assumption* (CWA) [66, 94], since $Poss(r)$ allows only R-total tuples from the relation r to be present in $Poss(r)$.

A *weak functional dependency* (WFD) over a relation schema R is a statement of the form $\Diamond(X \to Y)$, where $XY \subseteq R$. A relation r over R is said to *satisfy* the WFD $\Diamond(X \to Y)$ over R, if there is some $s \in Poss(r)$ such that for all $t_1, t_2 \in s$, if $t_1[X] = t_2[X]$, then $t_1[Y] = t_2[Y]$. We note that the definition of satisfaction of a WFD in a relation reduces to the standard definition of the satisfaction of an FD when the relation is R-total (in this case there is exactly one $s \in Poss(r)$ and $\forall s \in Poss(r)$ is equivalent to $\exists s \in Poss(r)$). We observe that $\Diamond$ can be viewed as representing the modal operator *possibly* of a normal system of propositional modal logic [95]. Finally we remark that the weak approach to satisfaction of an FD by an incomplete relation allows a higher degree of uncertainty to be represented in the database than the strong approach (where an FD must be satisfied in all possible worlds) [41]. The disadvantage of the weak over the strong approach is that strongly satisfied FDs are easier to maintain [41]. Hence, both approaches complement one another.

It is known that WFDs in the absence of an NFS enjoy the same axiomatization as "no information" FDs (NFDs) [25, 23]. However, WFDs are different from NFDs. First of all, WFDs are defined with respect to Codd's null marker `unk`. Under this interpretation we know that a value exists, whereas under the "no information" interpretation it may also be the case that no value exists at all. Moreover, WFDs and NFDs also behave differently. For example, the relation r over $R = ASL$ with the two tuples (Kiwi,G6Kiwi,Wellington) and (Kiwi,`unk`,Gisborne) satisfies the WFD $\Diamond(A \to S)$. However, the NFD $A \to S$ is violated by $r = \{(\text{Kiwi}, \text{G6Kiwi}, \text{Wellington}), (\text{Kiwi}, \texttt{ni}, \text{Gisborne})\}$. That is, we have two distinct tuples which have an information on the attribute A and the information is the same, but the first tuple has some information for S while the second tuple has "no information" for S.

In the context of NFDs we defined the weak agree set of two tuples as $ag^w(t_1, t_2) = \{A \in R \mid t_1(A) = \texttt{ni} = t_2(A)\}$. For WFDs we re-define this to be $ag^w(t_1, t_2) := \{A \in R \mid t_1(A) = \texttt{unk} \text{ or } t_2(A) = \texttt{unk}\}$. Intuitively, this makes perfect sense in this context: two tuples weakly agree on an attribute if there is a possible world on which they agree on A. The definition of a strong agree set $ag^s(t_1, t_2) := \{A \in R \mid t_1(A) = t_2(A) \text{ and } t_1(A) \neq \texttt{unk} \neq t_2(A)\}$ requires no adjustment apart from the notation of the null marker, and

$ag(t_1, t_2) := ag^s(t_1, t_2) \cup ag^w(t_1, t_2)$ as before. The next proposition, which gives a syntactic characterization of satisfaction of a WFD, follows from the definition of satisfaction.

Proposition 2 *Let $XY \subseteq R$ and r be a relation over R. Then r satisfies $\Diamond(X \to Y)$ if and only if for all $t_1, t_2 \in r$, if $X \subseteq ag^s(t_1, t_2)$, then $Y \subseteq ag(t_1, t_2)$.* ∎

A *weak multivalued dependency* (WMVDs) over R is a statement $\Diamond(X \twoheadrightarrow Y)$, where $XY \subseteq R$. A relation r over R is said to *satisfy* the WMVD $\Diamond(X \twoheadrightarrow Y)$ over R, if there is some $s \in Poss(r)$ such that for all $t_1, t_2 \in s$ the following holds: if $t_1[X] = t_2[X]$, then there is some $t \in s$ such that $t[XY] = t_1[XY]$ and $t[X(R - Y)] = t_2[X(R - Y)]$.

WMVDs behave quite differently from multivalued dependencies (NMVDs) in the "no information" context. For example, the following relation r over $R = ASLC$:

Article	*Supplier*	*Location*	*Cost*
Gold Kiwi	G6Kiwi	Wellington	1.50
Green Kiwi	G6Kiwi	Gisborne	2.50
Green Kiwi	unk	Wellington	2.50
Gold Kiwi	unk	Gisborne	1.50

satisfies the WMVD $\Diamond(S \twoheadrightarrow L)$. However, the "no information" MVD (NMVD) $S \twoheadrightarrow L$ is violated by

Article	*Supplier*	*Location*	*Cost*
Gold Kiwi	G6Kiwi	Wellington	1.50
Green Kiwi	G6Kiwi	Gisborne	2.50
Green Kiwi	ni	Wellington	2.50
Gold Kiwi	ni	Gisborne	1.50

.

For WMVDs, we can obtain the following syntactic characterization for their satisfaction by an incomplete relation.

Proposition 3 *Let $XY \subseteq R$ and r be a relation over R. Then r satisfies $\Diamond(X \twoheadrightarrow Y)$ if and only if for all $t_1, t_2 \in r$ with $X \subseteq ag^s(t_1, t_2)$ there is some $t \in r$ such that $X \subseteq ag^s(t, t_1)$, $Y \subseteq ag(t, t_1)$ and $R - Y \subseteq ag(t, t_2)$.* ∎

Let $\mathfrak{D}'$ denote the set of inference rules obtained from replacing the FDs and MVDs in $\mathfrak{D}$ by WFDs and WMVDs, respectively. Using Propositions 2 and 3 it is not difficult to show that the inference rules of $\mathfrak{D}'$ are sound for the implication of WFDs and WMVDs in the presence of an NFS. Following the same line of arguments as in Section 5 it can be shown that the system $\mathfrak{D}'$ forms a finite axiomatization for the combined class of WFDs and WMVDs in the presence of an NFS. In particular, the two-tuple relation r_φ

	$X(X^+_\Sigma \cap R_s)$	$(X^+_\Sigma - X) - R_s$	$W_1 \cap R_s$	$W_1 - R_s$	$\cdots$	W_i	$\cdots$	$W_k \cap R_s$	$W_k - R_s$
t_1	$0 \cdots 0$	unk $\cdots$ unk	$0 \cdots 0$	unk $\cdots$ unk		$0 \cdots 0$		$0 \cdots 0$	unk $\cdots$ unk
t_2	$0 \cdots 0$	$0 \cdots 0$	$0 \cdots 0$	$0 \cdots 0$		$1 \cdots 1$		$0 \cdots 0$	$0 \cdots 0$

shows that a WFD or WMVD φ is not implied by a set Σ of WFDs and WMVDs whenever φ cannot be inferred from Σ by $\mathfrak{D}'$, cf. the proof of Theorem 2. A number of further results follow, including the counter-part to Corollary 1. With the new definition of weak agree sets, Theorem 3 and Corollary 6 carry over to sets of WFDs and WMVDs in the presence of an NFS R_s and the families $LP_{\mathcal{S}}$ and $\mathcal{S}$-3, respectively, of para-consistent logics. The special LP interpretation remains the same, in particular $\omega'_{\{t_1,t_2\}}(A') := \mathbb{P}$ whenever t_1 and t_2 weakly agree on A, i.e., when they agree (in a possible world of $\{t_1, t_2\}$) on A and they disagree (in a possible world of $\{t_1, t_2\}$) on A. We summarize these results as follows.

Proposition 4 *Let $\Sigma \cup \{\varphi\}$ be a set of WFDs and WMVDs over the relation schema R, and let R_s denote an NFS over R. Let $\mathcal{L}$ denote the set of propositional variables that corresponds to R, $\mathcal{S}$ the set of variables that corresponds to R_s, and $\Sigma' \cup \{\varphi'\}$ the set of $\mathcal{L}$-formulae that corresponds to $\Sigma \cup \{\varphi\}$. Then the following are equivalent: (1) $\Sigma \models_{R_s} \varphi$, (2) $\Sigma \models_{2,R_s} \varphi$, (3) $\Sigma \vdash_{\mathfrak{D}'} \varphi$, (4) $\Sigma[lhs(\varphi)R_s] \models_R \varphi$, (5) $\Sigma[lhs(\varphi)R_s] \models_{2,R} \varphi$, (6) $(\Sigma[lhs(\varphi)R_s])' \models_{BL} \varphi'$, (7) $\Sigma' \models_{LP_{\mathcal{S}}} \varphi'$, (8) $\Sigma' \models^3_{\mathcal{S}} \varphi'$.* ∎

Hence, our results also establish SQL's `NOT NULL` constraint as an effective mechanism to control the expressiveness and efficiency of consequence relations under Levene and Loizou's weak possible world semantics for Codd's null marker `unk`.

7.2 Or-relations

In the case of the null marker `unk` an incomplete relation can have an infinite set of possible worlds, assuming that attribute domains are countably infinite. Consequently, each value in the domain is possible, or in other words, `unk` represents the disjunction of all the possible domain values. This simple approach has the disadvantage of being too vague in the case that we have some partial information. For example, let (Kiwi,`unk`) denote a tuple over the schema $R = AL$. The occurrence of `unk` in this tuple implies that we do not know from which location the article Kiwi is delivered. However, we may know that the article will be delivered from either Wellington or Gisborne, and this information could be represented as the finite set {Wellington, Gisborne} in the tuple (Kiwi,{Wellington,Gisborne}). Assume that this tuple represents all the information we have about the article Kiwi. Then this new tuple represents an increase of information: we can answer the query "Is the article *Kiwi* delivered from Auckland?" with a *no*. On the other hand, using the null marker *unk* we would have to answer the same query with *maybe*.

We will now define the framework for or-relations [68]. A non-empty finite set $\{v_1, \ldots, v_m\}$ of values, one of which is the true value, drawn from a given attribute domain $dom(A)$ is called an *or*-set over A. If $m = 1$, then the singleton or-set represents a known value, and otherwise it represents a set of possible values where it is unknown which value in the or-set is the true value. An *or*-tuple over $R = \{A_1, \ldots, A_n\}$ is a function $t : R \to \bigcup_{A \in R} \mathcal{P}_0(dom(A))$ such that for all $A \in R$, $t(A) \in \mathcal{P}_0(dom(A))$ holds, where $\mathcal{P}_0(dom(A))$ denotes the set of all or-sets over A. For some subset $X \subseteq R$, we say that t is *X-total*, if $t(A)$ is a singleton or-set for all $A \in X$. An *or*-relation over R is a

finite set of or-tuples. For some subset $X \subseteq R$, an or-relation r is said to be X-total, if every $t \in r$ is X-total. We say that an or-tuple t_2 is *subsumed* by an or-tuple t_1, if for all $A \in R$, $t_1(A) \subseteq t_2(A)$ holds. That is, having less values in an or-set represents having more information. The set of all *possible worlds* relative to an or-relation r over R, denoted by $Poss(r)$, is defined by

$$\begin{aligned} Poss(r) \;:=\; & \{s \mid s \text{ is an or-relation over } R \text{ and there is a total and onto mapping} \\ & f : r \to s \text{ where } \forall t \in r, t \text{ is subsumed by } f(t) \text{ and } f(t) \text{ is } R\text{-total}\}. \end{aligned}$$

An *or-free* subschema R_s of R with $R_s \subseteq R$ is satisfied by an or-relation r, if r is R_s-total. A *weak functional dependency* (WFD) over R is a statement $\Diamond(X \to Y)$ where $XY \subseteq R$. An or-relation r over R *satisfies* the WFD $\Diamond(X \to Y)$ over R, if there is some $s \in Poss(r)$ such that for all $t_1, t_2 \in s$ the following holds: if $t_1[X] = t_2[X]$, then $t_1[Y] = t_2[Y]$. The behavior of WFDs in the context of or-relations is quite different from that of NFDs in the "no information" context and WFDs in the "value unknown at present" context. For example, the following or-relation r

Article	*Location*	*Cost*
Gold Kiwi	Wellington	1.50
{Gold Kiwi, Green Kiwi}	Wellington	2.50
{Gold Kiwi, Green Kiwi}	Gisborne	2.50

satisfies the WFDs $\Diamond(A \to L)$ and $\Diamond(A \to C)$, but it violates the WFD $\Diamond(A \to LC)$. Consequently, the union rule $\mathcal{U}_{\mathrm{F}}$ is not sound for the implication of weak functional dependencies over or-relations.

A *weak multivalued dependency* (WMVDs) over R is a statement $\Diamond(X \twoheadrightarrow Y)$, where $XY \subseteq R$. An or-relation r over R is said to *satisfy* the WMVD $\Diamond(X \twoheadrightarrow Y)$ over R, if there is some $s \in Poss(r)$ such that for all $t_1, t_2 \in s$ the following holds: if $t_1[X] = t_2[X]$, then there is some $t \in s$ such that $t[XY] = t_1[XY]$ and $t[X(R-Y)] = t_2[X(R-Y)]$.

The behavior of WMVDs in the context of or-relations is different from that of NMVDs in the "no information" context and WMVDs in the "value unknown at present" context. For example, the following or-relation r

Article	*Supplier*	*Location*
Kiwi	G6Kiwi	{Wellington, Gisborne}
Kiwi	Kiwifruitz	{Maunganui, Auckland}
Kiwi	G6Kiwi	{Maunganui, Gisborne}
Kiwi	Kiwifruitz	{Wellington, Auckland}

satisfies the NFS $R_s = AS$ and the WMVDs $\Diamond(A \twoheadrightarrow S)$ and $\Diamond(S \twoheadrightarrow L)$, but it violates the WMVD $\Diamond(A \twoheadrightarrow L)$. Consequently, the null pseudo-transitivity rule $\mathcal{T}_{\mathrm{M}}$ is not sound for the implication of weak multivalued dependencies over or-relations. Note that the same relation also satisfies the WFD $\Diamond(S \to L)$, i.e., it shows that the null mixed pseudo-transitivity rule $\mathcal{T}_{\mathrm{FM}}$ is not sound for the implication of weak functional and multivalued dependencies over or-relations.

A further significant difference of or-relations to incomplete relations is the following fact. For the or-relation r

Article	*Location*
Kiwi	{Wellington,Gisborne}
Kiwi	{Gisborne,Auckland}
Kiwi	{Wellington,Auckland}

over $R = AL$ there is no possible world that satisfies the WFD $\Diamond(A \to L)$, but for every two-tuple subrelation of r there is a possible world that satisfies $\Diamond(A \to L)$. Hence, over or-relations it is not true that for every set $\Sigma \cup \{\varphi\}$ of WFDs and WMVDs with a relation r that satisfies Σ and violates φ there is a two-tuple subrelation of r that satisfies Σ and violates φ. The example above shows that this result already fails in the case where Σ is empty and φ is a WFD.

The examples above illustrate that the same classes of data dependencies behave quite differently over relations that allow arbitrary disjunctions of domain values than over relations where only finite disjunctions of domain values are allowed to occur. This also warrants future research on the implication problem of classes of data dependencies over or-relations.

8 Applications

In this section we illustrate the potential impact of our results on three major data processing tasks: updates, queries and access control.

8.1 Efficient processing of updates

As a first major application we illustrate how the choice of a null-free subschema impacts on the properties of decompositions that are derived from an extension of standard normalization algorithms [3, 96].

We say that a relation schema R is in *4NF* with respect to a set Σ of FDs and MVDs in the presence of an NFS R_s, cf. [20], if for every MVD $X \twoheadrightarrow Y \in \Sigma_{\mathfrak{D}}^+$ where $X \neq XY \neq R$ it follows that $X \to R \in \Sigma_{\mathfrak{D}}^+$, too. As in the special case where $R_s = R$ the syntactic condition for being in 4NF can be semantically justified by the absence of suitable notions of data redundancy and update anomalies [8]. The exact definitions and the proofs of these results, however, are beyond the scope of this article. We conclude that it is a desirable goal for a database designer to obtain a database schema in which every relation schema is in 4NF with respect to the given FDs and MVDs in the presence of an NFS. In fact, this condition guarantees that updates on every future database instance can be efficiently processed.

We will analyze how the choice of a null-free subschema R_s impacts on the properties of being lossless and dependency-preserving for database decompositions. We define the set $\{(R_1, R_s^1), \ldots, (R_n, R_s^n)\}$ with $R_s^i \subseteq R_i$ for all $i = 1, \ldots, n$ to be a *lossless join decomposition* of R with respect to Σ and R_s, if $R = \bigcup_{i=1}^n R_i$ and every relation r over

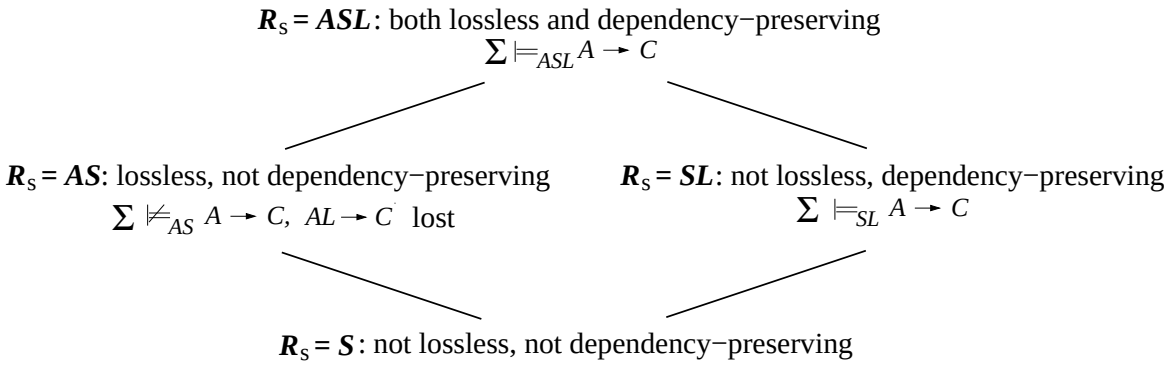

Figure 3: Properties of decompositions with respect to the choice of a null-free subschema

R that satisfies Σ and R_s also satisfies the following conditions: $r = r[R_1] \bowtie \cdots \bowtie r[R_n]$ and for $i = 1, \ldots, n$, $r[R_i]$ is R_s^i-total. We define the set $\{(R_1, R_s^1), \ldots, (R_n, R_s^n)\}$ with $R_s^i \subseteq R_i$ for all $i = 1, \ldots, n$ to be a *lossless 4NF decomposition* of R with respect to Σ and R_s, if $\{(R_1, R_s^1), \ldots, (R_n, R_s^n)\}$ is a lossless join decomposition of R with respect to Σ and R_s, and for all $i = 1, \ldots, n$, R_i is in 4NF with respect to

$$\Sigma_{\mathfrak{D}}^+[R_i, R_s^i] = \{X \to Y \in \Sigma_{\mathfrak{D}}^+ \mid XY \subseteq R_i\} \cup \{X \twoheadrightarrow Y \cap R_i \in \Sigma_{\mathfrak{D}}^+ \mid X \subseteq R_i\}$$

and R_s^i. Moreover, the set $\{(R_1, R_s^1), \ldots, (R_n, R_s^n)\}$ with $R_s^i \subseteq R_i$ for all $i = 1, \ldots, n$ is a *dependency-preserving decomposition* of R with respect to Σ and R_s, if for every set $\{r_1, \ldots, r_n\}$ of relations such that for all $i = 1, \ldots, n$, r_i is an R_s^i-total relation over R_i that satisfies $\Sigma_{\mathfrak{D}}^+[R_i, R_s^i]$, there is some R_s-total relation r over R that satisfies Σ and for which $r_i = r[R_i]$ for all $i = 1, \ldots, n$.

Consider now our running example where $R = ASLC$ and $\Sigma = \{A \to S, AL \to C, S \twoheadrightarrow L\}$. Following a 4NF-decomposition strategy one may decompose R based on the MVD $S \twoheadrightarrow L$ into $R_1 = SL$ and ASC, and then decompose ASC based on the FD $A \to S$ into $R_2 = AS$ and $R_3 = AC$.

Consider four different choices for the null-free subschema R_s and its natural propagation $R_s^i := R_i \cap R_s$ to the elements R_i of our decomposition. First, let $R_s = S$. Then $\{(R_1, S), (R_2, S), (R_3, \emptyset)\}$ is neither lossless nor dependency-preserving with respect to Σ and R_s. For $R_s = SL$, $\{(R_1, SL), (R_2, S), (R_3, \emptyset)\}$ is not lossless, but dependency-preserving. In fact, $\Sigma \models_{R_s} A \to C$. For $R_s = AS$, the set $\{(R_1, S), (R_2, AS), (R_3, A)\}$ is lossless, but not dependency-preserving. Recall that if a relation r satisfies an FD or MVD with left-hand side X and right-hand side Y, then $r_X[R] = r_X[XY] \bowtie r_X[X(R - Y)]$. Since we decomposed based on $S \twoheadrightarrow L$ and $A \to C$, including A and S into R_s ensures losslessness. Finally, for $R_s = ALS$ the set $\{(R_1, SL), (R_2, AS), (R_3, A)\}$ is a lossless and dependency-preserving 4NF decomposition of R with respect to Σ and R_s. The situation is illustrated in Figure 3. Hence, our results empower database designers to determine effectively the properties of decompositions.

8.2 Efficient processing of queries

Besides updates, the efficient processing of database queries is one of the most significant tasks of a database management system. We will now illustrate by example how the ability to decide efficiently the implication problem for sets of FDs and MVDs in the presence of an NFS can result in effective optimizations of database queries. Basically, new opportunities for query optimization arise whenever certain data dependencies are proven to be implied by a given set of data dependencies in the presence of a given NFS. Recall that the NFS has a significant impact on the decision whether a given data dependency is implied or not.

Consider again our running example where $R = ASLC$, $R_s = SL$, and $\Sigma = \{A \to S, AL \to C, S \twoheadrightarrow L\}$. Consider first the query that retrieves all combinations of locations and costs associated with the same article. A naive implementation of this query would be

$$\begin{array}{ll} \texttt{SELECT} & R.L,\ R'.C \\ \texttt{FROM} & R,\ R\ \texttt{AS}\ R' \\ \texttt{WHERE} & R.A = R'.A \end{array} .$$

However, since $\Sigma \models_{R_s} A \to C$ the cost of the article is the same for every location the article is delivered from. Consequently, the query can be rewritten into

$$\texttt{SELECT}\ R.L,\ R.C\ \texttt{FROM}\ R$$

which requires no join at all.

Another opportunity for optimizing queries is the identification of superfluous `DISTINCT` clauses. The gains in efficiency can be significant since duplicate elimination often requires an expensive sort of the query result [97]. For a detailed discussion how the detection of superfluous `DISTINCT` clauses can be used by database management systems we refer the interested reader to [98]. Essentially, our tools for reasoning about data dependencies over SQL table definitions enable database management systems to decide efficiently whether the attributes selected for a query output permit occurrences of duplicates. As an illustration consider Example 9 again, where Supplier $= ASLC$, Supplier$_{\mathcal{S}} = SLC$ and $\Sigma = \{A \to S, AL \to C, AC \to L, S \twoheadrightarrow L\}$. Consider now the query where we retrieve all distinct articles from the Article-column of Supplier that are not null. A naive implementation of this query would be

$$\begin{array}{ll} \texttt{SELECT DISTINCT} & \textsc{Supplier}.\mathrm{A} \\ \texttt{FROM} & \textsc{Supplier} \\ \texttt{WHERE} & \textsc{Supplier}.\mathrm{A}\ \texttt{NOT NULL} \end{array} .$$

However, since Σ implies the FD $A \to SLC$ in the presence of Supplier$_{\mathcal{S}}$, the `DISTINCT` clause is superfluous. Hence, the results we develop in this article can help to identify automatically superfluous `DISTINCT` clauses.

8.3 Inference control

Inference control is a security mechanism developed to ensure confidentiality in databases [31, 99, 100]. The objective is to avoid inferences of secrets by users based on their query

history and their knowledge about the database. Consider again our running example where $R = ASLC$, $R_s = SL$, and $\Sigma = \{A \to S, AL \to C, S \twoheadrightarrow L\}$. Suppose the fact that there is a supplier that delivers *Kiwis* from the location *Wellington* for the cost of *2NZD* is a business secret that some users of that database are not supposed to learn. That is, the fact that the sentence Ψ

$$\exists X_S R(\text{Kiwi}, X_S, \text{Wellington}, \text{2NZD})$$

is true in the database must not be revealed to unauthorized users. Nevertheless, a user may issue the queries:

- $\Phi_1 = (\exists X_S)(\exists X_L)R(\text{Kiwi}, X_S, X_L, \text{2NZD})$ and
- $\Phi_2 = (\exists X_S)(\exists X_C)R(\text{Kiwi}, X_S, \text{Wellington}, X_C)$

and learn that both queries are true in the current instance, since neither Φ_1 nor Φ_2 individually reveal Ψ. Unfortunately, this form of access control does not guarantee confidentiality since an attacker can exploit the fact that $\Sigma \models_{SL} A \twoheadrightarrow L$ holds. An application of the MVD $A \twoheadrightarrow L$ to the two tuples Φ_1 and Φ_2 reveals to the attacker that the potential secret Ψ is also an element of the database instance. Hence, clever attackers can utilize their background knowledge to bypass access control policies. Note that attackers are unable to draw the conclusion that Ψ is an element of the database instance if $S \notin R_s$. As a consequence, the tools developed in this article result in an advanced understanding of entailment relations that can assist security officers in preventing inference attacks on future database instances.

9 Conclusion

Previous theories and database practice warrant a thorough study of FDs and MVDs in the presence of an NFS. We established a finite axiomatization and efficient algorithms to decide the associated implication problem. These close the gap between theory and practice, and unify previously orthogonal theories for i) FDs and MVDs over total relations, ii) FDs in the presence of an NFS, and iii) FDs and MVDs in the absence of an NFS. For Lien, Atzeni and Morfuni's class of FDs and MVDs we established correspondences between their implication and the implication of fragments in Priest's Logic of Paradox. More generally, we established equivalences between the implication of FDs and MVDs in the presence of an arbitrary NFS and the implication of fragments of Cadoli and Schaerf's $\mathcal{S}$-3 logics. We also established the equivalence of the implication problem of this class to that of a reduced set of FDs and MVDs over total relations and, therefore, to that of a propositional fragment in Boolean logic by previous results from Sagiv, Delobel, Parker and Fagin. This enables the use of Galil's almost linear time algorithm to decide the implication problem for this class of data dependencies and that of its corresponding $\mathcal{S}$-3 fragment. In the electronic appendix, we extend our equivalences to the combined class of FDs and full first-order hierarchical decompositions, and the class of Boolean dependencies. Our findings apply to Zaniolo's "no information" nulls and to Codd's "value unknown at present", but not to Imielinksi's or-relations under

Levene and Loizou's weak possible world semantics. Our theory establishes SQL's `NOT NULL` constraint as an effective mechanism to balance the expressiveness and efficiency of entailment relations for significant classes of uni-relational dependencies that arise in practice. It also explains how Boolean entailment is soundly approximated by SQL table definitions.

10 Future directions

There are at least three directions to pursue in future work. Firstly, one may analyze other classes of data dependencies. A prime example are inclusion dependencies [101, 102]. Particularly interesting would be to study the impact of null-free subschemata on the interaction of functional and inclusion dependencies. It is also an open problem if the combined classes of strong and weak, functional and multivalued dependencies can be axiomatized [41]. Secondly, one should consider other approaches to incomplete information, including other interpretations of null markers [103, 77, 74, 75, 76], or-relations [68], fuzzy [72], rough sets [73], or world-set decompositions to manage probabilistic information [104]. Thirdly, a main observation for the application areas of Section 8 is that most of the existing theory does not apply to SQL tables. These areas include normalization [3, 27, 8], semantic query optimization [32], consistent query answering [46] and controlled query evaluation [10]. Permitting subsumption in database relations means that the class of functional and multivalued dependencies does no longer subsume the class of uniqueness constraints. It is therefore interesting to study the combined class of uniqueness constraints, functional and multivalued dependencies in the presence of an NFS. Results on the implication problem for the combined class of uniqueness constraints and FDs in the presence of an NFS, and normal forms that characterize the absence of data redundancy in relations that permit subsumption have been reported [105].

We plan to extend current design aids available for total relations [89, 106, 107, 29, 108]. Intuitively, design teams find it more difficult to understand the interaction of FDs and MVDs in the presence of an arbitrary NFS. Hence, Armstrong databases [109] may be of even greater value than for the special case of total relations [110]. It is therefore desirable to extend the results about Armstrong relations from the class of FDs in the presence of an NFS [111] to the combined class of FDs and MVDs in the presence of an NFS.

Other directions include the problems of dependency inference [30], data cleaning [12], and extremal problems [112, 113, 114] in the presence of null markers.

Our equivalences pave the way to develop a preference-based theory of dependencies where the administrator ranks sets of dependencies according to some preferences regarding the urgency of their enforcement or their relevance for query optimization, for example. As a starting point one may apply the para-consistent entailment relations of logical frameworks [115].

Bayesian networks provide a semantic modeling tool which facilitates the acquisition of probabilistic knowledge [116]. Here, Bayesian multivalued dependencies allow us to decompose a joint probability distribution into two of its marginalizations without the loss of information. Consequently, the probability of an event can be obtained, in principle, by

appropriate marginalizations of the joint probability distribution. It would be interesting to study the impact of our results on the relationships between dependencies over total relations and Bayesian dependencies over total probability distributions [116, 117, 118].

Multivalued dependencies have largely been unexplored for XML, except for [119, 120]. This is surprising as the body of research on functional dependencies over XML data is substantial, and multivalued dependencies aim to explore the lossless decompositions of documents in which they are exhibited.

Finally, one may study data exchange problems in the presence of inconsistent sets of source, target or source-to-target dependencies [14]. While inconsistencies may easily arise in practice, it is not clear how to deal with them in general, and what a reasonable solution to a data exchange problem constitutes in particular.

11 Acknowledgements

This research is supported by the Marsden fund council from Government funding, administered by the Royal Society of New Zealand. Sven Hartmann is supported by a research grant of the Alfried Krupp von Bohlen and Halbach foundation, administered by the German Scholars organisation.

References

[1] Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM **13**(6) (1970) 377–387

[2] Börger, E., Grädel, E., Gurevich, Y.: The classical decision problem. Springer, Heidelberg, Germany (1997)

[3] Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Boston, MA, USA (1995)

[4] Thalheim, B.: Entity-Relationship modeling. Springer, Heidelberg, Germany (2000)

[5] Arenas, M., Libkin, L.: An information-theoretic approach to normal forms for relational and XML data. J. ACM **52**(2) (2005) 246–283

[6] Köhler, H., Link, S.: Armstrong axioms and Boyce-Codd-Heath normal form under bag semantics. Inf. Process. Lett. **110**(16) (2010) 717–724

[7] Kolahi, S., Libkin, L.: An information-theoretic analysis of worst-case redundancy in database design. ACM Trans. Database Syst. **35**(1) (2010) Article 5

[8] Vincent, M.: Semantic foundations of 4NF in relational database design. Acta Inf. **36**(3) (1999) 173–213

[9] Deutsch, A., Popa, L., Tannen, V.: Query reformulation with constraints. SIGMOD Record **35**(1) (2006) 65–73

[10] Biskup, J.: Security in computing systems. Springer, Heidelberg, Germany (2009)

[11] Klug, A., Price, R.: Determining view dependencies using tableaux. ACM Trans. Database Syst. **7**(3) (1982) 361–380

[12] Fan, W., Geerts, F., Jia, X., Kementsietsidis, A.: Conditional functional dependencies for capturing data inconsistencies. ACM Trans. Database Syst. **33**(2) (2008) Article 6

[13] Cali, A., Calvanese, D., De Giacomo, G., Lenzerini, M.: Data integration under integrity constraints. Inf. Syst. **29**(2) (2004) 147–163

[14] Fagin, R., Kolaitis, P., Miller, R., Popa, L.: Data exchange: semantics and query answering. Theor. Comput. Sci. **336**(1) (2005) 89–124

[15] Delobel, C., Adiba, M.: Relational database systems. Elsevier North Holland, New York, NY, USA (1985)

[16] Wu, M.: The practical need for fourth normal form. In: Proceedings of the Twenty-third ACM SIGCSE Technical Symposium on Computer Science Education, Kansas City, USA, ACM (1992) 19–23

[17] Date, C., Darwen, H.: A guide to the SQL standard. Addison-Wesley Professional, Reading, MA, USA (1997)

[18] Beeri, C.: On the membership problem for functional and multivalued dependencies in relational databases. ACM Trans. Database Syst. **5**(3) (1980) 241–259

[19] Beeri, C., Fagin, R., Howard, J.H.: A complete axiomatization for functional and multivalued dependencies in database relations. In: Proceedings of the SIGMOD International Conference on Management of Data, Toronto, Canada, ACM (1977) 47–61

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

[21] Sagiv, Y., Delobel, C., Parker Jr., D.S., Fagin, R.: An equivalence between relational database dependencies and a fragment of propositional logic. J. ACM **28**(3) (1981) 435–453

[22] Zaniolo, C.: Database relations with null values. J. Comput. System Sci. **28**(1) (1984) 142–166

[23] Lien, E.: On the equivalence of database models. J. ACM **29**(2) (1982) 333–362

[24] Link, S.: On the implication of multivalued dependencies in partial database relations. Int. J. Found. Comput. Sci. **19**(3) (2008) 691–715

[25] Atzeni, P., Morfuni, N.: Functional dependencies and constraints on null values in database relations. Information and Control **70**(1) (1986) 1–31

[26] Fagin, R., Vardi, M.: The theory of data dependencies - a survey. In: Mathematics of Information Processing. Volume 34 of Proceedings of Symposia in Applied Mathematics., Louisville, USA, American Mathematical Society (1986) 19–72

[27] Paredaens, J., De Bra, P., Gyssens, M., Van Gucht, D.: The Structure of the Relational Database Model. Springer, Heidelberg, Germany (1989)

[28] Biskup, J., Dayal, U., Bernstein, P.: Synthesizing independent database schemas. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), Boston, USA, ACM (May 30 - June 1 1979) 143–151

[29] Mannila, H., Räihä, K.J.: Design by example: An application of Armstrong relations. J. Comput. System Sci. **33**(2) (1986) 126–141

[30] Mannila, H., Räihä, K.J.: Algorithms for inferring functional dependencies from relations. Data Knowl. Eng. **12**(1) (1994) 83–99

[31] Biskup, J., Embley, D., Lochner, J.: Reducing inference control to access control for normalized database schemas. Inf. Proc. Letters **106**(1) (2008) 8–12

[32] Deutsch, A., Ludäscher, B., Nash, A.: Rewriting queries using views with access patterns under integrity constraints. Theor. Comput. Sci. **371**(3) (2007) 200–226

[33] Arenas, M., Fan, W., Libkin, L.: On the complexity of verifying consistency of XML specifications. SIAM J. Comput. **38**(3) (2008) 841–880

[34] Bojanczyk, M., Muscholl, A., Schwentick, T., Segoufin, L.: Two-variable logic on data trees and XML reasoning. J. ACM **56**(3) (2009) Article 13

[35] Gottlob, G., Pichler, R., Wei, F.: Tractable database design and datalog abduction through bounded treewidth. Inf. Syst. **35**(3) (2010) 278–298

[36] Hartmann, S., Link, S.: Characterising nested database dependencies by fragments of propositional logic. Ann. Pure Appl. Logic **152**(1-3) (2008) 84–106

[37] Hartmann, S., Link, S.: Efficient reasoning about a robust XML key fragment. ACM Trans. Database Syst. **34**(2) (2009) Article 10

[38] Hartmann, S., Link, S.: Numerical constraints on XML data. Inf. Comp. **208**(5) (2010) 521–544

[39] Jensen, C., Snodgrass, R., Soo, M.: Extending existing dependency theory to temporal databases. IEEE Trans. Knowl. Data Eng. **8**(4) (1996) 563–582

[40] Kolahi, S.: Dependency-preserving normalization of relational and XML data. J. Comput. System Sci. **73**(4) (2007) 636–647

[41] Levene, M., Loizou, G.: Axiomatisation of functional dependencies in incomplete relations. Theor. Comput. Sci. **206**(1-2) (1998) 283–300

[42] Tari, Z., Stokes, J., Spaccapietra, S.: Object normal forms and dependency constraints for object-oriented schemata. ACM Trans. Database Syst. **22** (1997) 513–569

[43] Toman, D., Weddell, G.: On keys and functional dependencies as first-class citizens in description logics. J. Autom. Reasoning **40**(2-3) (2008) 117–132

[44] Wijsen, J.: Temporal FDs on complex objects. ACM Trans. Database Syst. **24**(1) (1999) 127–176

[45] Davidson, S., Fan, W., Hara, C.: Propagating XML constraints to relations. J. Comput. System Sci. **73**(3) (2007) 316–361

[46] Chomicki, J.: Consistent query answering: Five easy pieces. In: Proceedings of the 11th International Conference on Database Theory (ICDT). Volume 4353 of Lecture Notes in Computer Science., Barcelona, Spain, Springer (January 10-12 2007) 1–17

[47] Fagin, R., Kolaitis, P., Popa, L., Tan, W.: Reverse data exchange: coping with nulls. In: Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), Providence, USA, ACM (June 19 - July 1 2009) 23–32

[48] Arenas, M., Barcelo, P., Libkin, L., Murlak, F.: Relational and XML data exchange. Synthesis Lectures on Data Management. Morgan & Claypool Publishers (2010)

[49] Beeri, C., Vardi, M.: Formal systems for tuple and equality generating dependencies. SIAM J. Comput. **13**(1) (1984) 76–98

[50] Chandra, A., Lewis, H., Makowsky, J.: Embedded implicational dependencies and their inference problem. In: Proceedings of the 13th Annual ACM Symposium on Theory of Computing (STOC), Milwaukee, USA, ACM (May 11-13 1981) 342–354

[51] Fagin, R.: Horn clauses and database dependencies. J. ACM **29**(4) (1982) 952–985

[52] Makowsky, J., Vardi, M.: On the expressive power of data dependencies. Acta Inf. **23**(3) (1986) 231–244

[53] Petrov, S.Y.: Finite axiomatization of languages for representation of system properties: Axiomatization of dependencies. Inf. Sci. **47** (1989) 339–372

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

[55] Armstrong, W.W.: Dependency structures of database relationships. Information Processing **74** (1974) 580–583

[56] Beeri, C., Bernstein, P.: Computational problems related to the design of normal form relational schemas. ACM Trans. Database Syst. **4**(1) (1979) 30–59

[57] Diederich, J., Milton, J.: New methods and fast algorithms for database normalization. ACM Trans. Database Syst. **13**(3) (1988) 339–365

[58] Biskup, J., Link, S.: Appropriate inferences of data dependencies in relational databases. Ann. Math. Artif. Intell. **63**(3-4) (2011) 213–255

[59] Galil, Z.: An almost linear-time algorithm for computing a dependency basis in a relational database. J. ACM **29**(1) (1982) 96–102

[60] Hagihara, K., Ito, M., Taniguchi, K., Kasami, T.: Decision problems for multivalued dependencies in relational databases. SIAM J. Comput. **8**(2) (1979) 247–264

[61] Link, S.: Charting the completeness frontier of inference systems for multivalued dependencies. Acta Inf. **45**(7-8) (2008) 565–591

[62] Link, S.: Characterizing multivalued dependency implication over undetermined universes. *J. Comput. System Sci.*, doi:10.1016/j.jcss.2011.12.012 (2011)

[63] Sagiv, Y.: An algorithm for inferring multivalued dependencies with an application to propositional logic. J. ACM **27**(2) (1980) 250–262

[64] Fagin, R.: Functional dependencies in a relational data base and propositional logic. IBM Journal of Research and Development **21**(6) (1977) 543–544

[65] Codd, E.F.: Extending the database relational model to capture more meaning. ACM Trans. Database Syst. **4**(4) (1979) 397–434

[66] Imielinski, T., Lipski Jr., W.: Incomplete information in relational databases. J. ACM **31**(4) (1984) 761–791

[67] Levene, M., Loizou, G.: Database design for incomplete relations. ACM Trans. Database Syst. **24**(1) (1999) 80–125

[68] Imielinksi, T.: Incomplete information in logical databases. IEEE Data Eng. Bull. **12**(2) (1989) 29–40

[69] Imielinski, T., Van der Meyden, R., Vadaparty, K.: Complexity tailored design: a new design methodology for databases with incomplete information. J. Comput. System Sci. **51**(3) (1995) 405–432

[70] Libkin, L., Wong, L.: Semantic representations and query languages for Or-sets. J. Comput. System Sci. **52**(1) (1996) 125–142

[71] Vadaparty, K., Naqvi, S.: Using constraints for efficient query processing in nondeterministic databases. IEEE Trans. Knowl. Data Eng. **7**(6) (1995) 850–864

[72] Sözat, M., Yazici, A.: A complete axiomatization for fuzzy functional and multivalued dependencies in fuzzy database relations. ACM Fuzzy Sets and Systems **117**(2) (2001) 161–181

[73] Ziarko, W.: The discovery, analysis, and representation of data dependencies in databases. In: Knowledge Discovery in Databases, Cambridge, USA, MIT Press (1991) 195–212

[74] Grahne, G.: Dependency satisfaction in databases with incomplete information. In: Proceedings of the Tenth International Conference on Very Large Databases (VLDB), Singapore, IEEE Computer Society (August 27-31 1984) 37–45

[75] Grant, J.: Null values in a relational data base. Inf. Process. Lett. **6**(5) (1977) 156–157

[76] Makinouchi, A.: A consideration on normal form of not-necessarily-normalized relation in the relational data model. In: Proceedings of the Third International Conference on Very Large Databases (VLDB), Tokyo, Japan, IEEE Computer Society (October 6-8 1977) 447–453

[77] Gottlob, G., Zicari, R.: Closed world databases opened through null values. In: Proceedings of the Fourteenth International Conference on Very Large Databases (VLDB), Los Angeles, USA, IEEE Computer Society (August 29 - September 1 1988) 50–61

[78] Hartmann, S., Link, S.: When data dependencies over SQL tables meet the Logics of Paradox and S-3. In: Proceedings of the 29th ACM SIGMOD-SIGART-SIGACT Symposium on Principles of Database Systems (PoDS), Indianapolis, USA, ACM (2010) 317–326

[79] Schaerf, M., Cadoli, M.: Tractable reasoning via approximation. Artif. Intell. **74** (1995) 249–310

[80] Priest, G.: Logic of paradox. Journal of Philosophical Logic **8** (1979) 219–241

[81] Delobel, C.: Normalization and hierarchical dependencies in the relational data model. ACM Trans. Database Syst. **3**(3) (1978) 201–222

[82] Cadoli, M., Schaerf, M.: On the complexity of entailment in propositional multivalued logics. Ann. Math. Artif. Intell. **18**(1) (1996) 29–50

[83] Bernstein, P.: Synthesizing third normal form relations from functional dependencies. ACM Trans. Database Syst. **1**(4) (1976) 277–298

[84] Arenas, M., Libkin, L.: A normal form for XML documents. ACM Trans. Database Syst. **29**(1) (2004) 195–232

[85] Vincent, M., Liu, J., Liu, C.: Strong functional dependencies and their application to normal forms in XML. ACM Trans. Database Syst. **29**(3) (2004) 445–462

[86] Weddell, G.: Reasoning about functional dependencies generalized for semantic data models. ACM Trans. Database Syst. **17**(1) (1992) 32–64

[87] Beeri, C., Bernstein, P.A., Goodman, N.: A sophisticate's introduction to database normalization theory. In: Proceedings of the Fourth International Conference on Very Large Databases (VLDB), West Berlin, Germany, IEEE Computer Society (September 13-15 1978) 113–124

[88] Kleene, S.: An introduction to metamathematics. North Holland, Amsterdam, The Netherlands (1952)

[89] Beeri, C., Dowd, M., Fagin, R., Statman, R.: On the structure of Armstrong relations for functional dependencies. J. ACM **31**(1) (1984) 30–46

[90] Levesque, H.: A knowledge-level account of abduction. In: Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI), Detroit, USA, Morgan Kaufmann (1989) 1061–1067

[91] Davis, M., Putnam, H.: A computing procedure for quantification theory. J. ACM **7** (1960) 201–215

[92] Robinson, J.: A machine oriented logic based on resolution principle. J. ACM **12** (1965) 397–415

[93] Dowling, W., Gallier, J.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. J. Log. Program. **1**(3) (1984) 267–284

[94] Reiter, R.: On closed world data bases. In: Logic and Data Bases, New York, USA, Plenum Press (1978) 119–140

[95] Cresswell, M., Hughes, G.: A new introduction to modal logic. Routledge, London and New York (1996)

[96] Grahne, G., Räihä, K.J.: Database decomposition into Fourth Normal Form. In: Proceedings of the 9th International Conference on Very Large Databases (VLDB), Florence, Italy, IEEE Computer Society (October 31-November 2 1983) 186–196

[97] Paulley, G., Larson, P.A.: Exploiting uniqueness in query optimization. In: Proceedings of the Tenth International Conference on Data Engineering (ICDE), Houston, USA, IEEE Computer Society (February 14-18 1994) 68–79

[98] Paulley, G.: Exploiting functional dependence in query optimization. Technical Report UW-CS-2000-11, University of Waterloo, Waterloo, Canada (2000)

[99] Biskup, J., Weibert, T.: Keeping secrets in incomplete databases. Int. J. Inf. Sec. **7**(3) (2008) 199–217

[100] Farkas, C., Jajodia, S.: The inference problem: a survey. SIGKDD Explorations **4**(2) (2002) 6–11

[101] Casanova, M., Fagin, R., Papadimitriou, C.: Inclusion dependencies and their interaction with functional dependencies. J. Comput. System Sci. **28**(1) (1984) 29–59

[102] Levene, M., Loizou, G.: Null inclusion dependencies in relational databases. Inf. Comput. **136**(2) (1997) 67–108

[103] Codd, E.F.: Understanding relations. ACM SIGFIDET FDT Bulletin **7**(3-4) (1975) 23–28

[104] Antova, L., Koch, C., Olteanu, D.: $10^{(10^6)}$ worlds and beyond: efficient representation and processing of incomplete information. VLDB J. **18**(5) (2009) 1021–1040

[105] Ferrarotti, F., Hartmann, S., Köhler, H., Link, S., Vincent, M.: The Boyce-Codd-Heath normal form for SQL. In: Proceedings of the Eighteenth International Workshop on Logic, Language, Information and Computation (WoLLIC). Volume 6642 of Lecture Notes in Artificial Intelligence., Philadelphia, U.S.A., Springer (2011) 110–122

[106] De Marchi, F., Lopes, S., Petit, J.M., Toumani, F.: Analysis of existing databases at the logical level: the DBA companion project. SIGMOD Record **32**(1) (2003) 47–52

[107] De Marchi, F., Petit, J.M.: Semantic sampling of existing databases through informative Armstrong databases. Inf. Syst. **32**(3) (2007) 446–457

[108] Silva, A., Melkanoff, M.: A method for helping discover the dependencies of a relation. In: Proceedings of the Workshop on Formal Bases for Data Bases - Advances in Data Base Theory, Toulouse, France, Plemum Press (December 12-14 1979) 115–133

[109] Fagin, R.: Armstrong databases. Technical Report RJ3440(40926), IBM Research Laboratory, San Jose, California, USA (1982)

[110] Langeveldt, W., Link, S.: Empirical evidence for the usefulness of Armstrong relations on the acquisition of meaningful functional dependencies. Inf. Syst. **35**(3) (2010) 352–374

[111] Hartmann, S., Kirchberg, M., Link, S.: Design by example for SQL table definitions with functional dependencies. VLDB J. **21**(1) (2012) 121–144

[112] Demetrovics, J., Katona, G., Miklos, D., Thalheim, B.: On the number of independent functional dependencies. In: Proceedings of the Fourth International Symposium on Foundations of Information and Knowledge Bases (FoIKS). Number 3861 in Lecture Notes in Computer Science, Budapest, Hungary, Springer (February 14-17 2006) 83–91

[113] Engel, K.: Sperner theory. Cambridge Univ. Press, Cambridge, UK (1997)

[114] Hartmann, S., Leck, U., Link, S.: On Codd families of keys over incomplete relations. Comput. J. **54**(7) (2011) 1166–1180

[115] Marquis, P., Porquet, N.: Resource-bounded paraconsistent inference. Ann. Math. Artif. Intell. **39** (2003) 349–384

[116] Wong, S., Butz, C., Wu, D.: On the implication problem for probabilistic conditional independency. Trans. Systems, Man, and Cybernetics, Part A: Systems and Humans **30**(6) (2000) 785–805

[117] Malvestuto, F.: A unique formal system for binary decompositions of database relations, probability distributions, and graphs. Inf. Sci. **59**(1-2) (1992) 21–52

[118] Niepert, M., Van Gucht, D., Gyssens, M.: Logical and algorithmic properties of stable conditional independence. Int. J. Approx. Reasoning **51**(5) (2010) 531–543

[119] Saxton, L., Tang, X.: Tree multivalued dependencies for XML datasets. In: Proceedings of the Fifth International Conference on Advances in Web-Age Information Management (WAIM). Volume 3129 of Lecture Notes in Computer Science., Dalian, China, Springer (July 15-17 2004) 357–367

[120] Vincent, M., Liu, J.: Multivalued dependencies and a 4NF for XML. In: Proceedings of the 15th International Conference on Advanced Information Systems Engineering (CaISE). Volume 2681 of Lecture Notes in Computer Science., Klagenfurt, Austria, Springer (June 16-18 2003) 14–29

[121] Biskup, J.: On the complementation rule for multivalued dependencies. Acta Inf. **10** (1978) 297–305

[122] Zaniolo, C.: Mixed transitivity for functional and multivalued dependencies in database relations. Inf. Process. Lett. **10**(1) (1980) 32–34

[123] Khardon, R., Mannila, H., Roth, D.: Reasoning with examples: propositional formulae and database dependencies. Acta Inf. **36** (1999) 267–286

[124] Sagiv, Y., Delobel, C., Parker Jr., D.S., Fagin, R.: Correction to "An equivalence between relational database dependencies and a fragment of propositional logic". J. ACM **34**(4) (1987) 1016–1018

[125] Vardi, M.: The complexity of relational query languages. In: Proceedings of the Fourteenth ACM Symposium on Theory of Computing (STOC), San Francisco, USA, ACM (1982) 137–146

Table 4: Axiomatization $\mathfrak{S}_1$ for FDs & MVDs in the special case $R_s = R$

$$\frac{}{XY \to Y} \quad \text{(reflexivity, } \mathcal{R}_\text{F}\text{)} \qquad \frac{X \to Y}{XU \to YV}\, V \subseteq U \quad \text{(FD augmentation, } \mathcal{A}_\text{F}\text{)}$$

$$\frac{X \to Y \qquad Y \to Z}{X \to Z} \quad \text{(transitivity, } \mathcal{T}'_\text{F}\text{)}$$

$$\frac{X \twoheadrightarrow Y}{X \twoheadrightarrow R - Y} \quad (R\text{-complementation, } \mathcal{C}^R_\text{M}) \qquad \frac{X \twoheadrightarrow Y}{XU \twoheadrightarrow YV}\, V \subseteq U \quad \text{(MVD augmentation, } \mathcal{A}_\text{M}\text{)}$$

$$\frac{X \twoheadrightarrow Y \qquad Y \twoheadrightarrow Z}{X \twoheadrightarrow Z - Y} \quad \text{(pseudo-transitivity, } \mathcal{T}'_\text{M}\text{)}$$

$$\frac{X \to Y}{X \twoheadrightarrow Y} \quad \text{(implication, } \mathcal{I}_\text{FM}\text{)} \qquad \frac{X \twoheadrightarrow Y \qquad Y \to Z}{X \to Z - Y} \quad \text{(mixed pseudo-transitivity, } \mathcal{T}'_\text{FM}\text{)}$$

12 Appendix

In the appendix we establish i) how our new axiomatization subsumes three previous axiomatizations as special cases, ii) that every counter-example relation for an instance of the implication problem for FDs and MVDs in the presence of an NFS contains a two-tuple subrelation that is a counter-example for the same instance, and iii) equivalences for full first-order hierarchical decompositions and Boolean dependencies. In particular, we exemplify how findings on the implication problem can be transferred between the logical and data dependency frameworks.

13 Subsumption of previous axiomatizations

We demonstrate in this section how the reduction of $\mathfrak{D}$ to the previously studied special cases i) $R_s = R$, ii) $R_s = \emptyset$ and iii) $\Sigma \cup \{\varphi\}$ an FD set, subsumes the axiomatizations established for these cases [25, 19, 23], respectively.

13.1 Total relations

For this special case where $R_s = R$, Beeri, Fagin and Howard established the first axiomatization of FDs and MVDs [19]. The set $\mathfrak{S}_1$ from Table 4 forms an axiomatization for FDs and MVDs in the case $R_s = R$ [3, 27]. Biskup introduced the R-complementation rule $\mathcal{C}^R_\text{M}$

in this particular form [121] and Zaniolo [122] introduced the *mixed pseudo-transitivity rule* $\mathcal{T}'_{\mathrm{FM}}$.

In the special case $R_s = R$, the null mixed pseudo-transitivity rule $\mathcal{T}_{\mathrm{FM}}$ reduces to the rule $\mathcal{T}^1_{\mathrm{FM}}$:

$$\frac{X \twoheadrightarrow W \qquad Y \to Z}{X \to Z - W}\, Y \subseteq XW \ .$$

This rule subsumes Zaniolo's mixed pseudo-transitivity rule $\mathcal{T}'_{\mathrm{FM}}$ for the special case where $W = Y$. In fact, the following inference shows that *in the special case where* $R_s = R$ the mixed pseudo-transitivity rule $\mathcal{T}'_{\mathrm{FM}}$ can replace the null mixed pseudo-transitivity rule $\mathcal{T}^1_{\mathrm{FM}}$ in $\mathfrak{D}$ without losing completeness (for the application of $\mathcal{T}'_{\mathrm{FM}}$ note that $Y \subseteq XW$ and hence also $Z - XWY = Z - XW$):

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{\mathcal{R}_{\mathrm{F}}:\ X \to X}}{\mathcal{I}_{\mathrm{FM}}:\ X \twoheadrightarrow X} \quad X \twoheadrightarrow W}{\mathcal{U}_{\mathrm{M}}:\quad X \twoheadrightarrow XW} \quad \dfrac{\overline{\mathcal{R}_{\mathrm{F}}:\ XW \to Y}^{\,Y \subseteq XW}}{\mathcal{I}_{\mathrm{FM}}:\ XW \twoheadrightarrow Y} \quad Y \to Z}{\mathcal{T}'_{\mathrm{FM}}:\quad XW \to Z - Y}}{\mathcal{T}'_{\mathrm{FM}}:\quad X \to Z - XW} \qquad \dfrac{\overline{\mathcal{R}_{\mathrm{F}}:\ X \to (X - W) \cap Z}}{\mathcal{U}_{\mathrm{F}}:\quad X \to Z - W}}{} \ .$$

Similar observations hold for the null pseudo-transitivity rule $\mathcal{T}_{\mathrm{M}}$ and its counter-part, the pseudo-transitivity rule $\mathcal{T}'_{\mathrm{M}}$ in the special case of total relations. The following inference γ shows how the augmentation rule $\mathcal{A}_{\mathrm{F}}$ can be derived from the system $\mathfrak{D}$ in the special case where $R_s = R$:

$$\dfrac{\dfrac{\overline{\mathcal{R}_{\mathrm{F}}:\ XU \to X \cap Y} \quad \dfrac{\dfrac{\overline{XU \to X}}{\mathcal{I}_{\mathrm{FM}}:\ XU \twoheadrightarrow X} \quad X \to Y}{\mathcal{T}^1_{\mathrm{FM}}:\quad XU \to Y - X}}{\mathcal{U}_{\mathrm{F}}:\quad XU \to Y} \qquad \dfrac{\overline{\mathcal{R}_{\mathrm{F}}:\ XU \to V}^{\,V \subseteq U}}{\mathcal{U}_{\mathrm{F}}:\quad XU \to YV}}{} \ .$$

A similar observation holds for the augmentation rule $\mathcal{A}_{\mathrm{M}}$. Finally, the transitivity rule $\mathcal{T}'_{\mathrm{F}}$ can also be inferred from the system $\mathfrak{D}$ in the special case where $R_s = R$:

$$\dfrac{\dfrac{X \to Y \qquad \dfrac{\dfrac{X \to Y}{\mathcal{I}_{\mathrm{FM}}:\ X \twoheadrightarrow Y} \quad Y \to Z}{\mathcal{T}^1_{\mathrm{FM}}:\quad X \to Z - Y}}{\mathcal{D}_{\mathrm{F}}:\ X \to Y \cap Z}}{\mathcal{U}_{\mathrm{F}}:\quad X \to Z} \ .$$

Hence, the axiomatization $\mathfrak{S}_1$ is already subsumed by the axiomatization $\mathfrak{D}$ in the special case where $R_s = R$.

13.2 Lien's class of FDs and MVDs

This is the special case where $R_s = \emptyset$. Lien established the following set

$$\mathfrak{S}_2 = \{\mathcal{R}_{\mathrm{F}}, \mathcal{A}_{\mathrm{F}}, \mathcal{D}_{\mathrm{F}}, \mathcal{U}_{\mathrm{F}}, \mathcal{C}^R_{\mathrm{M}}, \mathcal{A}_{\mathrm{M}}, \mathcal{U}_{\mathrm{M}}, \mathcal{I}_{\mathrm{FM}}\}$$

as an axiomatization for the class of FDs and MVDs [23] in this case. Consequently, the interaction of FDs and MVDs is relatively trivial when $R_s = \emptyset$. For example, the null mixed pseudo-transitivity rule $\mathcal{T}_{\mathrm{FM}}$ reduces to the rule $\mathcal{T}^2_{\mathrm{FM}}$:

$$\frac{X \twoheadrightarrow W \qquad Y \to Z}{X \to Z - W} \, Y \subseteq X \; .$$

Replacing the application of $\mathcal{T}^1_{\mathrm{FM}}$ in the inference γ above by that of $\mathcal{T}^2_{\mathrm{FM}}$ shows how the augmentation rule $\mathcal{A}_{\mathrm{F}}$ can be derived from the system $\mathfrak{D}$ in the special case where $R_s = \emptyset$. Similar observations hold for the null pseudo-transitivity rule, its counter-part:

$$\frac{X \twoheadrightarrow W \qquad Y \twoheadrightarrow Z}{X \twoheadrightarrow Z - Y} \, Y \subseteq X$$

in the case $R_s = \emptyset$, and the augmentation rule $\mathcal{A}_{\mathrm{M}}$. Hence, the axiomatization $\mathfrak{S}_2$ is already subsumed by the axiomatization $\mathfrak{D}$ in the special case where $R_s = \emptyset$.

13.3 Atzeni and Morfuni's class of FDs in the presence of an NFS

Atzeni and Morfuni established the following set

$$\mathfrak{S}_3 = \{\mathcal{R}_{\mathrm{F}}, \mathcal{A}_{\mathrm{F}}, \mathcal{D}_{\mathrm{F}}, \mathcal{T}_{\mathrm{F}}\}$$

of inference rules as an axiomatization for the class of FDs in the presence of an NFS R_s [25]. Here, $\mathcal{T}_{\mathrm{F}}$ denotes the *null transitivity rule*:

$$\frac{X \to Y \qquad Y \to Z}{X \to Z} \, Y - X \subseteq R_s \quad .$$

This rule, however, can be derived from the system $\mathfrak{D}$ as follows:

$$\dfrac{\dfrac{X \to Y}{\mathcal{D}_{\mathrm{F}} : \; X \to Y \cap Z} \quad \dfrac{\dfrac{X \to Y}{\mathcal{I}_{\mathrm{FM}} : \; X \twoheadrightarrow Y} \quad Y \to Z}{\mathcal{T}_{\mathrm{FM}} : \quad X \to Z - Y} \; Y \subseteq X(Y \cap R_s)}{\mathcal{U}_{\mathrm{F}} : \qquad X \to Z}$$

Note that $Y - X \subseteq R_s$ implies that $Y \subseteq X(Y \cap R_s)$. Hence, the axiomatization $\mathfrak{S}_3$ is already subsumed by the axiomatization $\mathfrak{D}$ in the special case where Σ contains only FDs.

14 A model-theoretical result

For a set $\Sigma \cup \{\varphi\}$ of FDs and MVDs over any relation schema R we show in this section that any relation r that satisfies Σ and violates φ there is a two-tuple subrelation $r' \subseteq r$ that satisfies Σ and violates φ. This extends a result from total [21] to partial relations.

For a two-tuple relation $r = \{t_1, t_2\}$ over relation schema R we say that r *actively satisfies* the MVD $X \twoheadrightarrow Y$ over R if r satisfies $X \twoheadrightarrow Y$, t_1 and t_2 are X-total and $t_1[X] = t_2[X]$.

Lemma 6 *Let $r = \{t_1, t_2\}$ be a two-tuple relation over relation schema R. Let $X \twoheadrightarrow Y$ be an MVD over R such that R is the disjoint union of X, Y and Z. Then r actively satisfies $X \twoheadrightarrow Y$ if and only if i) $X \subseteq ag^s(t_1, t_2)$, and ii) $Y \subseteq ag(t_1, t_2)$ or $Z \subseteq ag(t_1, t_2)$.*

Proof If both i) and ii) hold, then r actively satisfies $X \twoheadrightarrow Y$. Suppose that $r = \{t_1, t_2\}$ actively satisfies $X \twoheadrightarrow Y$. Consequently, i) holds. Hence, $t_1[X] = t_2[X]$ and t_1, t_2 are X-total. Then there must be some $t \in r$ such that $t[XY] = t_1[XY]$ and $t[XZ] = t_2[XZ]$. However, $t = t_1$ or $t = t_2$. Consequently, $t_1[XZ] = t_2[XZ]$ or $t_1[XY] = t_2[XY]$, respectively. That is, $Y \subseteq ag(t_1, t_2)$ or $Z \subseteq ag(t_1, t_2)$. ∎

Lemma 7 *Let $r = \{t_1, t_2\}$ and $r' = \{t'_1, t'_2\}$ both be two-tuple relations over relation schema R such that $ag^s(t_1, t_2) \subseteq ag^s(t'_1, t'_2)$ and $ag(t_1, t_2) \subseteq ag(t'_1, t'_2)$. Then every MVD φ over R that is actively satisfied by r is also actively satisfied by r'.*

Proof Let r actively satisfy the MVD $X \twoheadrightarrow Y$ over R. Then we know that $X \subseteq ag^s(t_1, t_2)$. We conclude that $X \subseteq ag^s(t'_1, t'_2)$ since $ag^s(t_1, t_2) \subseteq ag^s(t'_1, t'_2)$.

From Lemma 6 we also conclude that $Y \subseteq ag(t_1, t_2)$ or $R - XY \subseteq ag(t_1, t_2)$. Since $ag(t_1, t_2) \subseteq ag(t'_1, t'_2)$ we conclude that $Y \subseteq ag(t'_1, t'_2)$ or $R - XY \subseteq ag(t'_1, t'_2)$.

Hence, Lemma 6 tells us that r' also actively satisfies $X \twoheadrightarrow Y$. ∎

Lemma 8 *Let $\Sigma \cup \{\varphi\}$ be a set of FDs and MVDs over the relation schema R, and let r be some relation over R. Assume that r satisfies Σ and violates φ. Then there is a two-tuple subrelation $r' \subseteq r$ such that r' satisfies Σ and violates φ.*

Proof We distinguish between two cases, depending on whether φ is an FD or an MVD.

Case 1. Let φ denote an FD. We can assume without loss of generality that φ denotes the FD $X \to A$ in which the right-hand side contains a single attribute. Since r violates $X \to A$ there are two X-total tuples t_1 and t_2 of r that agree on the attributes in X but disagree on the attribute A. Consider all two-tuple subrelations of r in which $X \to A$ is violated. Of all such two-tuple subrelations of r, let r' be the one which satisfies actively the maximal number of MVDs. That is, if s is another two-tuple subrelation of r which violates φ, and if k is the number of MVDs that are satisfied actively by s, then r' satisfies actively at least k MVDs. We shall now show that r' satisfies Σ.

All FDs in Σ are satisfied by r' since they are satisfied by r, and hence in every subrelation of r including r'. Let $U \twoheadrightarrow V$ be an MVD in Σ that is violated by r'. We shall derive a contradiction. Assume without loss of generality that U, V and W form a partition of the relation schema R. The two tuples of r' are clearly U-total and agree on all attributes of U (otherwise r' would satisfy $U \twoheadrightarrow V$). Let (u, v, w) and (u, v', w') denote the two tuples in r'. Consequently, $v \neq v'$ and $w \neq w'$ (or else r' would satisfy $U \twoheadrightarrow V$). By assumption, r' violates $X \to A$. Thus, the two tuples are X-total and agree on all the attributes in X and disagree on the attribute A. Since they disagree on A, we have either $A \in V$ or $A \in W$. Assume without loss of generality that $A \in V$. Let s be the two-tuple relation containing (u, v, w) and (u, v', w). Since r satisfies $U \twoheadrightarrow V$, and since (u, v, w) and (u, v', w') are in r, the tuple (u, v', w) is necessarily in r. Hence, s is a two-tuple subrelation of r. The two tuples of s are X-total and agree on all attributes in X (since the two tuples in r' do) but disagree on A (because v and v' disagree on A).

Thus, s violates $X \to A$, and, unlike the situation in r', we see that s actively satisfies $U \twoheadrightarrow V$. Furthermore, by Lemma 7 every MVD that is actively satisfied by r' is also actively satisfied by s. So more members of Σ are actively satisfied by s than by r'. Since s is a two-tuple subrelation of r which violates $X \to A$, this is a contradiction of the "maximality" in the definition of r'. This completes the proof of case 1.

Case 2. Let φ denote the MVD $X \twoheadrightarrow Y$. Assume without loss of generality that X, Y and Z form a partition of the relation schema R. We say that a pair of tuples (x, y, z) and (x, y', z') *witness the failure* of $X \twoheadrightarrow Y$ in a given relation if they appear in that relation, if they are X-total and if one of (x, y', z) or (x, y, z') does not appear in that relation. Thus an MVD fails in a relation if and only if the relation has a pair of tuples that witness the failure. In particular, since r violates the MVD $X \twoheadrightarrow Y$, let (x, y, z) and (x, y', z') witness the failure of $X \twoheadrightarrow Y$ in r. Hence, (x, y', z) or (x, y, z') does not appear in r. Of all two-tuple subrelations of r that witness the failure of $X \twoheadrightarrow Y$ in r, let r' be the one which actively satisfies the maximal number of MVDs in Σ. We now show that r' satisfies Σ (which completes the proof, since r' violates $X \twoheadrightarrow Y$).

As in case 1, each FD in Σ is satisfied by r'. Let $U \twoheadrightarrow V$ be an MVD in Σ that is violated by r'; we shall derive a contradiction. Assume that U, V and W form a partition of the relation schema R. As in case 1, the two tuples of r' are U-total and agree on all the attributes in U.

Denote by $\overline{V}$ and $\overline{W}$ those attributes in V and W, respectively, for which the tuples of r' disagree. Since $U \twoheadrightarrow V$ is violated by r', $\overline{V}$ and $\overline{W}$ are both necessarily non-empty. We rewrite (x, y, z) and (x, y', z') as (u, v, w) and (u, v', w'), respectively. Let s_1 be the two-tuple relation consisting of (u, v, w) and (u, v', w), and let s_2 be the two-tuple relation consisting of (u, v, w) and (u, v, w'). Both s_1 and s_2 are subrelations of r since $U \twoheadrightarrow V$ is satisfied by r. They are two-tuple relations since $v \neq v'$ and $w \neq w'$. By Lemma 7 every MVD of Σ that is actively satisfied by r' is also actively satisfied by s_1 and by s_2. Clearly $U \twoheadrightarrow V$ is actively satisfied by s_1 and s_2. If $X \twoheadrightarrow Y$ is violated by s_1 or by s_2, we have derived a contradiction to the maximality of r', and hence the proof is complete. So suppose that $X \twoheadrightarrow Y$ is satisfied by both s_1 and s_2. Then $X \twoheadrightarrow Y$ is actively satisfied by s_1 and by s_2 since all of the tuples in r', s_1 and s_2 are X-total and have the same X-value x. It follows from Lemma 6 that the two tuples in s_1 agree on all attributes in Y or on all attributes in Z. In the former case $\overline{V} \subseteq Z$, since $\overline{V}$ contains all the attributes in which the two tuples in s_1 disagree. In the latter case $\overline{V} \subseteq Y$. Thus we know that $\overline{V} \subseteq Y$ or $\overline{V} \subseteq Z$. Similarly, it follows from our knowledge of s_2 that $\overline{W} \subseteq Y$ or $\overline{W} \subseteq Z$. Since $\overline{V} \subseteq Y$ or $\overline{V} \subseteq Z$, and since $\overline{W} \subseteq Y$ or $\overline{W} \subseteq Z$, there are four possibilities:

1. $\overline{V} \subseteq Y$ and $\overline{W} \subseteq Y$;
2. $\overline{V} \subseteq Y$ and $\overline{W} \subseteq Z$;
3. $\overline{V} \subseteq Z$ and $\overline{W} \subseteq Y$;
4. $\overline{V} \subseteq Z$ and $\overline{W} \subseteq Z$.

Now $\overline{V} \cup \overline{W}$ contains all the attributes on which the two tuples of r' disagree. If (1) were to hold, then the two tuples in r' would agree on all the attributes in Z, and hence the

MVD $X \twoheadrightarrow Y$ would be satisfied by r' (which it is not). Similarly, (4) is impossible. So we have that (2) or (3) holds. We assume without loss of generality that (2) holds. Hence y and y' disagree exactly on all the attributes in $\overline{V}$, and z and z' disagree exactly on all the attributes in $\overline{W}$. Under these conditions (x, y', z) and (u, v', w) are identical, and so are (x, y, z') and (u, v, w'). But this is impossible, since (u, v', w) and (u, v, w') are in r, whereas (x, y', z) or (x, y, z') is not in r. ∎

15 Further equivalences

In this section we will demonstrate how our techniques can be applied to establish further equivalences. It is known from the case of total relations that the equivalences to Boolean implication do not extend to embedded or join dependencies. We demonstrate that our equivalences do extend to Delobel's class of full first-order hierarchical decompositions [81] in the presence of an NFS. We will then utilize the special LP interpretation to introduce the class of Boolean dependencies over incomplete relations, and to extend the equivalence between FDs in the presence of an NFS and the Horn fragment of $\mathcal{S}$-3 logics to arbitrary Boolean dependencies in the presence of an NFS and propositional formulae in $\mathcal{S}$-3 logics. As an application of this equivalence the upper time bounds on the time-complexity of the implication problem, previously established for $\mathcal{S}$-3 logics, transfer directly to the framework of Boolean dependencies. For the special case of an empty NFS, we obtain a detailed analysis of the data, expression and combined complexity from the findings for the Logic of Paradox. Vice versa, our axiomatization and upper time bounds for the implication of FDs and MVDs in the presence of an NFS apply directly to the $\mathcal{S}$-3 implication of the propositional fragment. Next we establish a logical characterization for the notions of a dependency basis and attribute set closure. Finally, we characterize the implication problem for Boolean dependencies in the presence of an arbitrary NFS by the implication problem for Boolean dependencies in the absence of an NFS.

15.1 Extension to Delobel's Full First-Order Hierarchical Decompositions

It is known that, already for the special case of total relations, the equivalences of Theorem 3 do not extend to join or embedded dependencies [21]. Delobel introduced the class of full first-order hierarchical decompositions (FOHDs) as an important subclass of join dependencies [81]. We will now extend the class of FOHDs to the context of the "no information" null marker, and show under which translation to propositional formulae the equivalences of Theorem 3 apply to the class of FOHDs.

An FOHD over a relation schema R is an expression $X : \{Y_1, \ldots, Y_k\}$ where $X, Y_1, \ldots, Y_k$ are all subsets of R such that $XY_1 \cdots Y_k = R$. We assume without loss of generality that the attribute sets $X, Y_1, \ldots, Y_k$ are mutually disjoint and that $k \geq 2$. The FOHD $X : \{Y_1, \ldots, Y_k\}$ over R is satisfied by a relation r over R if and only if $r_X[R] = r_X[XY_1] \bowtie \cdots \bowtie r_X[XY_k]$. An MVD $X \twoheadrightarrow Y$ is satisfied by a relation r if and only if the binary FOHD $X : \{Y, R - XY\}$ is satisfied by r.

For the FOHD $X : \{Y_1, \ldots, Y_k\}$, denoted by φ, let φ' denote its corresponding $\mathcal{L}$-formula:

$$\left(\bigwedge_{A \in X} A'\right) \rightarrow \left(\bigvee_{i=1}^{k} \left(\bigwedge_{B \in \bigcup_{1 \leq j \leq k, j \neq i} Y_j} B'\right)\right).$$

Theorem 5 *Let $\Sigma \cup \{\varphi\}$ be a set of FDs and FOHDs, and let R_s be an NFS over some relation schema R. Let $\mathcal{L}$ be the set of propositional variables that corresponds to R, $\mathcal{S}$ the set of propositional variables that corresponds to R_s, and let $\Sigma' \cup \{\varphi'\}$ denote the set of $\mathcal{L}$-formulae that corresponds to $\Sigma \cup \{\varphi\}$. Then the following statements are equivalent:*

1. $\Sigma \models_{R_s} \varphi$,
2. $\Sigma' \models_{LP_{\mathcal{S}}} \varphi'$,
3. $\Sigma' \models^3_{\mathcal{S}} \varphi'$, *and*
4. $(\Sigma[lhs(\varphi)R_s])' \models_{BL} \varphi'$.

Proof The result follows from Theorem 3, Corollary 6 and Corollary 7 since a relation r satisfies an FOHD $X : \{Y_1, \ldots, Y_k\}$ if and only if for all $i = 1, \ldots, k-1$ it is the case that r satisfies the MVD $X \twoheadrightarrow Y_i$. ∎

Example 11 *Let $R = ASLC$, $R_s = ALC$, $\Sigma = \{A \rightarrow S, AL \rightarrow C, S : \{L, AC\}\}$ as in Example 2. The relation*

Article	Supplier	Location	Cost
Kiwi	***ni***	*Maunganui*	*1.50*
Kiwi	***ni***	*Taranaki*	*2.50*

shows that the FOHD $A : \{S, L, C\}$, denoted by φ, is not implied by Σ in the presence of R_s. As an illustration of Theorem 5 we note that the $LP_{\mathcal{S}}$ interpretation that assigns $\mathbb{T}$ to A', $\mathbb{P}$ to S', and $\mathbb{F}$ to L' and C', is a model of Σ' but not a model of the formula $\varphi' = A' \rightarrow ((L' \wedge C') \vee (S' \wedge C') \vee (S' \wedge L'))$. ∎

15.2 Boolean dependencies

As an application of our special *LP* interpretation we introduce the class of Boolean dependencies (BDs) in the presence of an NFS. This class subsumes Atzeni and Morfuni's class of FDs in the presence of an NFS [25, 23], and the class of BDs over total relations (where the NFS $R_s = R$) [21]. Note that MVDs are not BDs.

Remark 6 *Boolean dependencies have not been very popular in database design yet. However, the following extension of them allows designers to express quite useful constraints. Define for each attribute $A \in R$ an equivalence relation E_A on the domain of A. Define further that the Boolean dependency $A \rightarrow B \vee \neg C$ holds if and only if for*

any two tuples t_1 and t_2 we have $(t_1(A), t_2(A)) \in E_A$, then either $(t_1(B), t_2(B)) \in E_B$ or $(t_1(C), t_2(C)) \notin E_C$. Using this generalization, we can, e.g., express the following statement about insurance policy holders. Assume that we have attributes Age, Premium, and Sex, and that we define equivalence relations for age groups and premium groups. Then we can express the constraint if two policy holders belong to the same age group, then their premiums are in the same class or they are of different sexes [123].

The class of *Boolean dependencies* (BDs) over a relation schema R is defined as the propositional language $\mathcal{B}_R := R^*$ over R. An *agreement* over R is a function $\omega : R \to \{\mathbb{D}, \mathbb{W}, \mathbb{S}\}$. For two distinct tuples t_1, t_2 over R we define the agreement $\omega_{\{t_1,t_2\}}$ of t_1 and t_2 by

$$\omega_{\{t_1,t_2\}}(A) = \begin{cases} \mathbb{S} & \text{, if } A \in ag^s(t_1, t_2) \\ \mathbb{W} & \text{, if } A \in ag^w(t_1, t_2) \\ \mathbb{D} & \text{, if } A \notin ag(t_1, t_2) \end{cases}$$

for all $A \in R$. Intuitively, the definition carries the following meaning: $\omega_{\{t_1,t_2\}}(A) = \mathbb{S}$ when t_1 and t_2 *s*trongly agree on A, $\omega_{\{t_1,t_2\}}(A) = \mathbb{W}$ when t_1 and t_2 *w*eakly agree on A, and $\omega_{\{t_1,t_2\}}(A) = \mathbb{D}$ when t_1 and t_2 *d*isagree on A. We can extend an agreement ω over R to a function $\Omega : \mathcal{B}_R \to \{\mathbb{D}, \mathbb{W}, \mathbb{S}\}$ as follows:

1. if $\varphi = A \in R$ let $\Omega(\varphi) := \omega(A)$,
2. if $\varphi = (\neg\psi)$ let $\Omega(\varphi) := \neg\Omega(\psi)$,
3. if $\varphi = (\varphi_1 \vee \varphi_2)$ let $\Omega(\varphi) := \Omega(\varphi_1) \vee \Omega(\varphi_2)$, and
4. if $\varphi = (\varphi_1 \wedge \varphi_2)$ let $\Omega(\varphi) := \Omega(\varphi_1) \wedge \Omega(\varphi_2)$.

On the right-hand side of these definitions, $\neg$, $\vee$, and $\wedge$ denote the truth functions defined by Table 3 where $\mathbb{F}$, $\mathbb{P}$ and $\mathbb{T}$ are replaced by $\mathbb{D}$, $\mathbb{W}$ and $\mathbb{S}$, respectively.

For a relation r and a BD φ over relation schema R we say that r *satisfies* φ, denoted by $\models_r \varphi$, if and only if for all tuples $t_1, t_2 \in r$ the following holds: if $t_1 \neq t_2$, then $\Omega_{\{t_1,t_2\}}(\varphi) \in \{\mathbb{W}, \mathbb{S}\}$. In particular, BDs subsume FDs: a relation r satisfies the FD $\{A_1, \ldots, A_n\} \to \{B_1, \ldots, B_m\}$ [25, 23] if and only if r satisfies the BD $(A_1 \wedge \cdots \wedge A_n) \to (B_1 \wedge \cdots \wedge B_m)$.

Let $\phi : R \to \mathcal{L}$ be a bijection between a relation schema R and its corresponding set $\mathcal{L} = \{A' \mid A \in R\}$ of propositional variables. We extend ϕ to a mapping Φ from $\mathcal{B}_R$ to the set $\mathcal{L}^*$. As before, let $\varphi' = \Phi(\varphi)$ and $\Sigma' = \{\sigma' \mid \sigma \in \Sigma\}$. We define

1. if $\varphi = A$, then $\varphi' = A'$,
2. if $\varphi = (\neg\psi)$, then $\varphi' = (\neg\psi')$,
3. if $\varphi = (\varphi_1 \vee \varphi_2)$, then $\varphi' = (\varphi_1' \vee \varphi_2')$, and
4. if $\varphi = (\varphi_1 \wedge \varphi_2)$, then $\varphi' = (\varphi_1' \wedge \varphi_2')$.

We want to show that for any set $\Sigma \cup \{\varphi\}$ of Boolean dependencies there is an R_s-total relation r that satisfies Σ and violates φ if and only if there is an $LP_{\mathcal{S}}$ model ω'_r of Σ' that is not an $LP_{\mathcal{S}}$ model of φ'. The following result is the counter-part of Lemma 8. Note that the definition of satisfaction for BDs implies directly that their implication is equivalent to that in the world of two-tuple relations. However, since the proof of the following lemma is considerably simpler than that of Lemma 8 we include it here.

Lemma 9 *Let $\Sigma \cup \{\varphi\}$ be a set of BDs over the relation schema R, and let r be some relation over R that satisfies Σ and violates φ. Then there is a two-tuple subrelation $r' \subseteq r$ such that r' satisfies Σ and violates φ.*

Proof Since r violates the BD φ there are two tuples $t_1, t_2 \in r$ such that $t_1 \neq t_2$ and $\Omega_{\{t_1,t_2\}}(\varphi) = \mathbb{D}$. Let $r' := \{t_1, t_2\}$. We know that r' satisfies Σ since r satisfies Σ and $r' \subseteq r$. Consequently, r' is a two-tuple subrelation of r that satisfies Σ and violates φ. ∎

Lemma 9 tells us that for deciding the implication problem $\Sigma \models \varphi$ it suffices to examine two-tuple relations (instead of arbitrary finite relations). For two-tuple relations $\{t_1, t_2\}$, however, we can define a corresponding LP interpretation $\omega'_{\{t_1,t_2\}}$. For two tuples t_1, t_2 over the relation schema R let $\omega'_{\{t_1,t_2\}}$ denote the following *special* LP interpretation of $\mathcal{L}$:

$$\omega'_{\{t_1,t_2\}}(A') = \begin{cases} \mathbb{T} & \text{, if } \omega_{\{t_1,t_2\}}(A) = \mathbb{S} \\ \mathbb{P} & \text{, if } \omega_{\{t_1,t_2\}}(A) = \mathbb{W} \\ \mathbb{F} & \text{, if } \omega_{\{t_1,t_2\}}(A) = \mathbb{D} \end{cases} .$$

The following lemma justifies the definition of the special LP interpretation.

Lemma 10 *Let r be a two-tuple relation over the relation schema R, and let φ denote a BD over R. Then we have*

- *$\Omega_r(\varphi) = \mathbb{S}$ if and only if $\Omega'_r(\varphi') = \mathbb{T}$,*
- *$\Omega_r(\varphi) = \mathbb{W}$ if and only if $\Omega'_r(\varphi') = \mathbb{P}$, and*
- *$\Omega_r(\varphi) = \mathbb{D}$ if and only if $\Omega'_r(\varphi') = \mathbb{F}$.*

Proof The lemma can be proven by an induction over the structure of φ. ∎

Indeed, a two-tuple relation r satisfies a BD φ if and only if ω'_r is an LP model of the corresponding $\mathcal{L}$-formula φ'.

Corollary 8 *Let r be a two-tuple relation over the relation schema R, and let φ denote a BD over R. Then r satisfies φ if and only if ω'_r is an LP model of φ'.* ∎

In fact, Lemma 9 and Corollary 8 allow us to establish the anticipated equivalence between the implication of BDs in the presence of an NFS and the $LP_{\mathcal{S}}$ implication of propositional formulae. However, to obtain the equivalence we need to assume that duplicate tuples can occur.

Theorem 6 *Let $\Sigma \cup \{\varphi\}$ be a set of BDs over the relation schema R, and let R_s denote an NFS over R. Let $\mathcal{L}$ denote the set of propositional variables that corresponds to R, $\mathcal{S}$ the set of variables that corresponds to R_s, and $\Sigma' \cup \{\varphi'\}$ the set of $\mathcal{L}$-formulae that corresponds to $\Sigma \cup \{\varphi\}$. Under the assumption that relations may contain duplicate tuples, the following are equivalent:*

1. $\Sigma \models_{R_s} \varphi$,
2. $\Sigma \models_{2,R_s} \varphi$, *and*
3. $\Sigma' \models_{LP_{\mathcal{S}}} \varphi'$.

Proof The equivalence between (1) and (2) is an immediate consequence of Lemma 9.

For (3) implies (2) suppose (2) does not hold. Then there is some two-tuple relation r over R that satisfies Σ and R_s but violates φ. Since r is R_s-total it follows immediately from the definition of the special LP interpretation that ω'_r is an $LP_{\mathcal{S}}$ interpretation. Following Corollary 8, ω'_r is an $LP_{\mathcal{S}}$ model of Σ' but not an $LP_{\mathcal{S}}$ model of φ'. Consequently, (3) does also not hold.

For (2) implies (3) suppose that (3) does not hold. Then there is an $LP_{\mathcal{S}}$ interpretation ω' that is a model of Σ' but not a model of φ'. Define an agreement $\omega : R \to \{\mathbb{D}, \mathbb{W}, \mathbb{S}\}$ by

$$\omega(A) := \begin{cases} \mathbb{S} & \text{, if } \omega'(A') = \mathbb{T} \\ \mathbb{W} & \text{, if } \omega'(A') = \mathbb{P} \\ \mathbb{D} & \text{, if } \omega'(A') = \mathbb{F} \end{cases} .$$

Let t_1, t_2 be two tuples over R such that for all $A \in R$ we have $\omega_{\{t_1,t_2\}}(A) = \omega(A)$. In particular, if $\omega'(A') = \mathbb{F}$, then let $t_1(A)$ and $t_2(A)$ be distinct elements from $dom(A) - \{\texttt{ni}\}$. We conclude that in the case where one of t_1, t_2 is subsumed by the other, then $t_1(A) = t_2(A)$ holds for all $A \in R$. Moreover, following the definition of the agreement ω it is true that t_1 and t_2 are R_s-total since ω' is an $LP_{\mathcal{S}}$ interpretation. According to Corollary 8 we know then that r satisfies Σ and R_s, but r violates φ. Therefore, (2) does also not hold. ∎

The following example illustrates the use of duplicate tuples in a counter-example.

Example 12 *Let $R = ASLC$ denote the relation schema* Supplies*, let $R_s = ALC$, let Σ consist of the FDs $A \to S$ and $A \to C$, and let φ denote the BD $A \to \neg L$. The BD says that the same article is not to be delivered from the same location more than once. The following relation r*

Article	Supplier	Location	Cost
Kiwi	***ni***	*Taranaki*	*2.50*
Kiwi	***ni***	*Taranaki*	*2.50*

shows that Σ does not imply φ in the presence of R_s. Indeed, the same article can be delivered from the same location more than once (by the same supplier at the same cost). For ω'_r we obtain $\omega'_r(A') = \mathbb{T}$, $\omega'_r(S') = \mathbb{P}$, $\omega'_r(L') = \mathbb{T}$ and $\omega'_r(C') = \mathbb{T}$. Indeed, ω'_r is an $LP_{\{A',L',C'\}}$ interpretation that is a model of Σ' but not a model of φ'. ∎

Remark 7 *For the special case where $R_s = R$ it is known [124] that the BD $\phi'_R = \bigvee_{A \in R} \neg A$ has the following property: the relation $r = \{t_1, t_2\}$ is a set (and not a bag) if and only if the special truth assignment ω'_r satisfies $\phi'_R = \bigvee_{A \in R} \neg A'$. In the general case, however, it is no longer true that there is a BD ϕ'_R that characterizes subsumption-free two-tuple relations. In fact, for $R = A$ both $r_1 = \{a, \boldsymbol{ni}\}$ and $r_2 = \{a, a'\}$ satisfy the same BDs over R, but r_2 is subsumption-free whereas r_1 is not.* ∎

For the special case where $R_s = R$ it is known [124] that for every relation schema R and for every set $\Sigma \cup \{\varphi\}$ of Boolean dependencies over R it is true that $\Sigma \models_R \varphi$ if and only if $\Sigma' \cup \{\phi'_R\} \models_{LP_{\mathcal{L}}} \varphi'$ where $\mathcal{L}$ corresponds to the NFS R. According to Theorem 6 one may suspect that a similar result holds in the general case of an arbitrary NFS R_s. The following proposition shows that this already fails for the special case where $R_s = \emptyset$.

Proposition 5 *There is a relation schema R, a singleton FD set Σ and a BD φ over R such that $\Sigma \models \varphi$, but $\Sigma' \cup \{\phi'_R\} \not\models_{LP} \varphi'$.*

Proof Let $R = \{A(\text{rticle}), L(\text{ocation}), C(\text{ost})\}$, $\Sigma = \{A \to C\}$ and $\varphi = A \to \neg L$. Suppose that φ is violated by some relation r that satisfies Σ. Since r violates φ there are two distinct tuples $t_1, t_2 \in r$ such that $t_1(A) = t_2(A)$, t_1, t_2 are A-total and $t_1(L) = t_2(L)$. Since r satisfies Σ it follows that $t_1(C) = t_2(C)$ holds as well. That is, r must be a bag and not a relation. Consequently, there is no relation that satisfies Σ and violates φ. We conclude that $\Sigma \models \varphi$.

On the other hand, the LP interpretation ω' with $\omega'(A') = \mathbb{T} = \omega'(L')$ and $\omega'(C') = \mathbb{P}$ shows that $\Sigma' \cup \{\neg A' \vee \neg L' \vee \neg C'\} \not\models_{LP} \varphi'$. ∎

15.3 Complexity considerations

15.3.1 Boolean dependencies in the absence of a null-free subschema

As a consequence of Theorem 6 we obtain worst-case time-complexity results for the implication problem of BDs. This problem has been investigated in depth for the logic LP, and Table 5 provides a summary of the results [82]. In fact, the problem has been analyzed with respect to three different notions of complexity defined by Vardi [125]. For data complexity, Σ is the input and φ has fixed size. For expression complexity, φ is the input and Σ has fixed size. For combined complexity, Σ and φ are the input. The complexity results also distinguish the input with respect to its syntactic form. A BD is in *Conjunctive Normal Form* (CNF) when it is a single conjunction of clauses, where a *clause* is a disjunction of literals (i.e. an attribute A or its negation $\neg A$). A BD is in *Disjunctive Normal Form* (DNF) when it is a single disjunction of conjunctions of literals. In Table 5, the symbol "Any" means that no assumption is made on the syntactic form of the BDs. For a discussion of these results we refer the reader to [82].

15.3.2 Boolean dependencies in the presence of a null-free subschema

Let Σ be an arbitrary set of BDs over R in NNF, and let φ be an arbitrary BD in CNF. The following findings establish NFSs as an effective mechanism to balance the

Table 5: Time-complexities for Deciding BDs

$\Sigma \models \varphi$				
Σ	φ	$\Sigma? \models \varphi?$ (Combined)	$\Sigma_0 \models \varphi?$ (Expression)	$\Sigma? \models \varphi_0$ ("Data")
Any	Any	coNP-complete	coNP-complete	$\mathcal{O}(\|\Sigma\|)$
Any	CNF	$\mathcal{O}(\|\Sigma\| \times \|\varphi\|)$	$\mathcal{O}(\|\varphi\|)$	$\mathcal{O}(\|\Sigma\|)$
Any	DNF	coNP-complete	coNP-complete	$\mathcal{O}(\|\Sigma\|)$
CNF	Any	coNP-complete	coNP-complete	$\mathcal{O}(\|\Sigma\|)$
CNF	CNF	$\mathcal{O}(\|\Sigma\| \times \|\varphi\|)$	$\mathcal{O}(\|\varphi\|)$	$\mathcal{O}(\|\Sigma\|)$
CNF	DNF	coNP-complete	coNP-complete	$\mathcal{O}(\|\Sigma\|)$
DNF	Any	coNP-complete	coNP-complete	$\mathcal{O}(\|\Sigma\|)$
DNF	CNF	$\mathcal{O}(\|\Sigma\| \times \|\varphi\|)$	$\mathcal{O}(\|\varphi\|)$	$\mathcal{O}(\|\Sigma\|)$
DNF	DNF	coNP-complete	coNP-complete	$\mathcal{O}(\|\Sigma\|)$

expressiveness and efficiency of various entailment relations. They follow immediately from results established for $\mathcal{S}$-3 logics [79, Theorems 4.4 and 4.6] and Theorem 6.

Corollary 9 *For every set Σ of BDs in NNF, and every BD φ in CNF, and every NFSs R_s and R'_s over R such that $R_s \subseteq R'_s$, if $\Sigma \models_{R_s} \varphi$, then $\Sigma \models_{R'_s} \varphi$.* ∎

Corollary 9 establishes the monotonicity for the family of the entailment relations $\models_{R_s}$. That is, by declaring attributes as `NOT NULL`, a data administrator enforces at least all of the previously enforced data dependencies. In Figure 2, this is illustrated as a potential increase in expressiveness.

Corollary 10 *The implication problem $\Sigma \models_{R_s} \varphi$ for sets Σ of BDs in NNF, BDs φ in CNF and NFS R_s over relation schemata R can be decided in time $\mathcal{O}(|\Sigma| \times |\varphi| \times 2^{|R_s|})$.* ∎

Corollary 10 establishes a uniform complexity for deciding the family of entailment relations $\models_{R_s}$ in terms of the NFSs R_s. Therefore, by declaring attributes as `NULL`, a data administrator can utilize more efficient algorithms for deciding the associated implication problem. In Figure 2, this is illustrated as a potential increase in efficiency.

15.4 Characterizing the notions of dependency basis and attribute set closure

As an application of Theorem 3 we generalize Sagiv, Delobel, Parker and Fagin's logical characterization of a dependency basis and attribute set closure from the special case where $R_s = R$ [21] to an arbitrary NFS R_s.

Theorem 7 *Let Σ denote a set of FDs and MVDs, and R_s an NFS over the relation schema R. Let $\mathcal{L}$ denote the set of propositional variables that corresponds to R, $\mathcal{S}$ the set of variables that corresponds to R_s, and Σ' the set of $\mathcal{L}$-formulae that corresponds to Σ.*

Let X and W be disjoint subsets of R. Let $\omega'_{W'}$ denote the following $LP_{\mathcal{S}}$ interpretation of $\mathcal{L}$:

$$\omega'_{W'}(A') = \begin{cases} \mathbb{T} & , \text{ if } A \in X((R-W) \cap R_s) \\ \mathbb{P} & , \text{ if } A \in (R-W) - R_s \\ \mathbb{F} & , \text{ if } A \in W \end{cases} .$$

If $W \in DepB_{\Sigma,R_s}(X)$ and W and X^*_{Σ,R_s} are disjoint, then $\omega'_{W'}$ is an $LP_{\mathcal{S}}$ model of Σ'. If $\omega'_{W'}$ is an $LP_{\mathcal{S}}$ model of Σ', then W is contained in one set of $DepB_{\Sigma,R_s}(X)$ and W is disjoint from X^*_{Σ,R_s}.

Proof Let $W \in DepB_{\Sigma,R_s}(X)$, and let W and X^*_{Σ,R_s} be disjoint. We show that $\omega'_{W'}$ is an $LP_{\mathcal{S}}$ model of Σ'. The proof of Theorem 2 defines a two-tuple relation r which satisfies Σ and R_s. Without loss of generality, let $W := W_i$ where W_i denotes the set from $DepB_{\Sigma,R_s}(X)$ in the proof of Theorem 2. According to Lemma 3 and Lemma 5 it follows that ω'_r is an $LP_{\mathcal{S}}$ model of Σ'. However, we have $\omega'_{W'} = \omega'_r$. This shows that $\omega'_{W'}$ is an $LP_{\mathcal{S}}$ model of Σ'.

Let $\omega'_{W'}$ be an $LP_{\mathcal{S}}$ model of Σ'. We show first that W is contained in one set of $DepB_{\Sigma,R_s}(X)$. Suppose to the contrary that W is not contained in any set of $DepB_{\Sigma,R_s}(X)$. Then there is a set $V \in DepB_{\Sigma,R_s}(X)$ such that $V \cap W \neq \emptyset$ and $W \cap (R-V) \neq \emptyset$. It follows that $\omega'_{W'}$ is not an $LP_{\mathcal{S}}$ model of the formula that corresponds to $X \twoheadrightarrow V$. However, Theorems 1 and 2 show that $X \twoheadrightarrow V$ is implied by Σ in the presence of R_s, and Theorem 3 shows that $\omega'_{W'}$ is an $LP_{\mathcal{S}}$ model of the formula that corresponds to $X \twoheadrightarrow V$ since $\omega'_{W'}$ is an $LP_{\mathcal{S}}$ model of Σ'. This is a contradiction. Consequently, W is contained in one set of $DepB_{\Sigma,R_s}(X)$.

We show now that W is disjoint from X^*_{Σ,R_s}. If W is not disjoint from X^*_{Σ,R_s}, then W, as an element of $DepB_{\Sigma,R_s}(X)$, must be a singleton set according to Theorem 1, say $W = A$. The attribute A is a member of X^*_{Σ,R_s}, and therefore there must be an FD $Y \to Z$ in Σ such that $A \in Z - Y$ and $Y \subseteq XR_s$ (otherwise $X \to A$ cannot be derived from Σ, cf. Lemma 2). Since $Y \subseteq XR_s$ and $A = W$ it follows that $\omega'_{W'}$ is not an $LP_{\mathcal{S}}$ model of $Y \to A$. This, however, is a contradiction since $Y \to A \in \Sigma$ and $\omega'_{W'}$ is an $LP_{\mathcal{S}}$ model of Σ'. Consequently, W is disjoint from X^*_{Σ,R_s}. ∎

Intuitively, $\omega'_{W'}$ is the special LP interpretation ω'_r induced by the two-tuple relation $r := r_\varphi$ in our completeness proof of $\mathfrak{D}$ (where $W = W_i$), cf. Table 2. Note that $\omega'_{W'}$ reduces to the propositional truth assignment defined in [21] for the special case where $R_s = R$. Moreover, $\omega'_{W'}$ is equivalent to an $\mathcal{S}$-3 interpretation, cf. Proposition 1.

15.5 Boolean and $\mathcal{S}$-3 implication

In [63] Sagiv presents an algorithm for deciding the implication problem $\Sigma \models_R \varphi$ for sets $\Sigma \cup \{\varphi\}$ of FDs and MVDs over R. The algorithm can be implemented to run in time $\mathcal{O}(\bar{p}_\Sigma \times |\Sigma|)$ where $\bar{p}_\Sigma$ is the number of sets in $DepB_{\Sigma,R}(lhs(\varphi))$ that have non-empty intersection with the right-hand side of φ. Using Corollary 2 we can apply Sagiv's algorithm to decide implication in the presence of an NFS R_s in time $\mathcal{O}(|\Sigma| + \bar{p}_{\Sigma[lhs(\varphi)R_s]} \times |\Sigma[lhs(\varphi)R_s]|)$. Following Corollary 4, we note that our

$$\mathcal{O}(|\Sigma| + \min\{k_{\Sigma[lhs(\varphi)R_s]}, \log \bar{p}_{\Sigma[lhs(\varphi)R_s]}\} \times |\Sigma[lhs(\varphi)R_s]|)$$

algorithm for deciding $\Sigma \models_{R_s} \varphi$ can be applied directly to decide $\Sigma' \models^3_{\mathcal{S}} \varphi'$ for the corresponding fragment $\mathcal{F}$ of Cadoli and Schaerf's $\mathcal{S}$-3 logics, cf. Corollary 6, a fragment not studied previously to our knowledge. It follows that the axiomatization $\mathfrak{D}$ for the implication of FDs and MVDs in the presence of an NFS R_s also applies to $\mathcal{S}$-3 implication in $\mathcal{F}$. For any set $\Sigma' \cup \{\varphi'\}$ of formulae in $\mathcal{F}$ and any $\mathcal{S} \subseteq \mathcal{L}$, $\Sigma' \models^3_{\mathcal{S}} \varphi'$ if and only if $\Sigma'[lhs(\varphi')\mathcal{S}] \models_{BL} \varphi'$. Here, $\Sigma'[lhs(\varphi')\mathcal{S}]$ is the set of formulae in Σ' whose set of variables in the antecedent is a subset of $lhs(\varphi')\mathcal{S}$, and $lhs(\varphi')$ is the set of variables in the antecedent of φ'. Note that this confirms a result by Cadoli and Schaerf [79].

15.6 The power of reasoning without null-free subschemata

The following theorem shows that reasoning about BDs in the presence of an NFS can be simulated by reasoning about BDs in the absence of an NFS. It explains why our correspondences to the Logic of Paradox provide a very general tool for reasoning about data dependencies. For a clause φ let $Attr(\varphi)$ denote the set of attributes that occur in φ.

Theorem 8 *Let $\Sigma \cup \{\varphi\}$ be a set of BDs over relation schema R where φ is a clause, and let R_s denote an NFS over R. Then the following two statements are equivalent:*

1. $\Sigma \models_{R_s} \varphi$
2. $\Sigma \cup \{\neg\varphi\} \models \bigvee_{A \in R_s \cup Attr(\varphi)} (A \wedge \neg A)$.

Proof According to Theorem 6 it suffices to show the equivalence in the world of two-tuple relations. Without loss of generality let φ denote the clause $\neg A_1 \vee \cdots \vee \neg A_n \vee B_1 \vee \cdots \vee B_m$.

For (2) implies (1) we assume that (1) does not hold. Consequently, there is some two-tuple relation r over R such that r satisfies Σ and the NFS R_s but r violates φ. The violation of φ by r means that $\Omega_r(\varphi) = \mathbb{D}$, but this means that $\omega_r(A_i) = \mathbb{S}$ for all $i = 1, \ldots, n$ and $\omega_r(B_j) = \mathbb{D}$ for all $j = 1, \ldots, m$. In particular we know that r satisfies Σ and $\neg\varphi$. The satisfaction of R_s by r means that $\omega_r(A) \in \{\mathbb{D}, \mathbb{S}\}$ for all $A \in R_s$. Consequently, for all $A \in R_s \cup Attr(\varphi)$ we have $\omega_r(A) \in \{\mathbb{D}, \mathbb{S}\}$. Hence, $\Omega_r(A \wedge \neg A) = \mathbb{D}$ for all $A \in R_s \cup Attr(\varphi)$. That is, r violates $\bigvee_{A \in R_s \cup Attr(\varphi)} (A \wedge \neg A)$. We have shown that (2) does also not hold.

For (1) implies (2) we assume that (2) does not hold. Consequently, there is some two-tuple relation $r = \{t_1, t_2\}$ over R such that r satisfies $\Sigma \cup \{\neg\varphi\}$ but r violates

$$\bigvee_{A \in R_s \cup Attr(\varphi)} (A \wedge \neg A).$$

The latter implies that $\Omega_r(A \wedge \neg A) = \mathbb{D}$ for all $A \in R_s \cup Attr(\varphi)$. Now we change the relation r for every attribute $A \in R_s$ where $\omega_r(A) = \mathbb{D}$ holds and where we have either $t_1(A) = \texttt{ni}$ or $t_2(A) = \texttt{ni}$. Without loss of generality let $t_1(A) = \texttt{ni}$. Then we replace $t_1(A)$ by a new value from $dom(A) - \{\texttt{ni}, t_2(A)\}$. For the resulting relation r' we have that $\omega_{r'} = \omega_r$ holds. Moreover, r' satisfies R_s. Consequently, r' satisfies Σ and R_s but it violates φ. Consequently, (1) does also not hold. ∎

We give an example to illustrate the correspondence established in Theorem 8.

Example 13 *Let* $R = ASLC$, $R_s = ALC$, $\Sigma = \{A \to S, AL \to C, S \twoheadrightarrow L\}$ *and* $\varphi = \neg A \vee L \vee (S \wedge C)$. *Note that the satisfaction of* φ *is equivalent to that of the two clauses* $\varphi_1 = \neg A \vee L \vee S$ *and* $\varphi_2 = \neg A \vee L \vee C$. *The relation*

Article	Supplier	Location	Cost
Kiwi	***ni***	*Maunganui*	*1.50*
Kiwi	***ni***	*Taranaki*	*2.50*

satisfies Σ *and* R_s, *but violates* φ_2 *and therefore violates* φ. *One can also see that* r *satisfies* $\Sigma \cup \{\neg\varphi_2\}$ *and violates* $(A \wedge \neg A) \vee (L \wedge \neg L) \vee (C \wedge \neg C)$. ∎