



On the Interaction of Inclusion Dependencies with Independence Atoms*

Miika Hannula¹, Juha Kontinen², and Sebastian Link³

¹ University of Auckland, Auckland, New Zealand
m.hannula@auckland.ac.nz

² University of Helsinki, Helsinki, Finland
juha.kontinen@helsinki.fi

³ University of Auckland, Auckland, New Zealand
m.hannula@auckland.ac.nz

Abstract

Inclusion dependencies are one of the most important database constraints. In isolation their finite and unrestricted implication problems coincide, are finitely axiomatizable, PSPACE-complete, and fixed-parameter tractable in their arity. In contrast, finite and unrestricted implication problems for the combined class of functional and inclusion dependencies deviate from one another and are each undecidable. The same holds true for the class of embedded multivalued dependencies. An important embedded tractable fragment of embedded multivalued dependencies are independence atoms. These stipulate independence between two attribute sets in the sense that for every two tuples there is a third tuple that agrees with the first tuple on the first attribute set and with the second tuple on the second attribute set. For independence atoms, their finite and unrestricted implication problems coincide, are finitely axiomatizable, and decidable in cubic time. In this article, we study the implication problems of the combined class of independence atoms and inclusion dependencies. We show that their finite and unrestricted implication problems coincide, are finitely axiomatizable, PSPACE-complete, and fixed-parameter tractable in their arity. Hence, significant expressivity is gained without sacrificing any of the desirable properties that inclusion dependencies have in isolation. Finally, we establish an efficient condition that is sufficient for independence atoms and inclusion dependencies not to interact. The condition ensures that we can apply known algorithms for deciding implication of the individual classes of independence atoms and inclusion dependencies, respectively, to decide implication for an input that combines both individual classes.

1 Introduction

Databases represent information about some domain of the real world. For this purpose, data dependencies provide the main mechanism for enforcing the semantics of the given application

*This research is supported by a Marsden grant from Government funding, administered by the Royal Society of New Zealand, and grant 292767 of the Academy of Finland.

domain within a database system. As such, data dependencies are essential for most data management tasks, including conceptual, logical and physical database design, query and update processing, transaction management, as well as data cleaning, exchange, and integration. The usability of a class \mathcal{C} of data dependencies for these tasks depends critically on the computational properties of its associated implication problem. The implication problem for \mathcal{C} is to decide whether for a given finite set $\Sigma \cup \{\varphi\}$ of data dependencies from \mathcal{C} , Σ implies φ , i.e. whether every database that satisfies all the elements of Σ also satisfies φ . If we require databases to be finite, then we speak of the finite implication problem, and otherwise of the unrestricted implication problem. While the importance of data dependencies continues to hold for new data models, the focus of this article is on the implication problems for important classes of data dependencies in the relational model of data. In this context, data dependency theory is deep and rich [34].

Inclusion dependencies (INDs) constitute one of the most used classes of data dependencies in practice. In particular, INDs are more expressive than foreign keys, thereby capturing Codd's principle of referential integrity on the logical level. An IND $R[A_1, \dots, A_n] \subseteq R'[B_1, \dots, B_n]$, with attribute sequences A_1, \dots, A_n on R and B_1, \dots, B_n on R' , expresses that for each tuple t over R there is some tuple t' over R' such that for all $i = 1, \dots, n$, $t(A_i) = t'(B_i)$ holds. If $n = 1$ we call the IND unary (UIND). As a simple example consider the relation schemata

HEART= $\{p_id, p_name, t_id\}$, and
DISORDER= $\{p_id, t_id, confirmed\}$

in which basic information about patients and medical tests is stored. In particular, HEART stores which medical tests (identified by t_id) for a specific heart disorder were performed on which patients (with ID p_id and name p_name), and DISORDER stores all those tests (identified by t_id) performed on patients (identified by p_id) which have been diagnosed with the disorder. The UIND $\sigma_1 = \text{DISORDER}[p_id] \subseteq \text{HEART}[p_id]$ expresses that each id of a patient who has been diagnosed with a heart disease must also be the id of a patient who has been tested for this heart disease. Similarly, the UIND $\sigma_2 = \text{DISORDER}[t_id] \subseteq \text{HEART}[t_id]$ expresses that each id of a test performed on a patient who has been diagnosed with a heart disease must also be the id of a test used to diagnose this heart disease.

A fundamental result in dependency theory is that the unrestricted and finite implication problems for the class of INDs coincide, are finitely axiomatizable, PSPACE-complete to decide [8], and fixed-parameter tractable in their arity [22, 23]. Combining INDs with functional dependencies (FDs) results in implication problems that differ in the finite and unrestricted case and are both undecidable [9, 28, 29].

Another important expressive class of data dependencies are *embedded multivalued dependencies* (EMVDs). An EMVD $R : X \rightarrow Y \perp Z$ with attribute subsets X, Y, Z of R expresses that the projection $r[XYZ]$ of a relation r over R on the set union XYZ is the join $r[XY] \bowtie r[XZ]$ of its projections on XY and XZ . Another fundamental result in dependency theory is that the unrestricted and finite implication problems for EMVDs differ, each is not finitely axiomatizable [33] and each is undecidable [19, 20]. An important fragment of EMVDs are multivalued dependencies (MVDs), which are a class of full dependencies in which XYZ covers the full underlying set R of attributes. In fact, MVDs are the basis for Fagin's fourth normal form [13]. For the combined class of FDs, MVDs, and UINDs, finite implication is axiomatizable and decidable in cubic time, while unrestricted implication is also axiomatizable and decidable in almost linear time [10, 21].

Another expressive known fragment of EMVDs that is computationally friendly is the class of *independence atoms* (IAs). IAs are EMVDs $R : X \rightarrow Y \perp Z$ where $X = \emptyset$, i.e. expressing that $r[YZ] = r[Y] \bowtie r[Z]$ holds. IAs are denoted by $Y \perp Z$. In our example, the IA $\sigma_3 = \text{HEART} : p_id \perp t_id$ on schema HEART expresses that all patients that are tested for a heart disorder undergo all tests for this disorder. Similarly, the IA $\sigma_4 = \text{DISORDER} : confirmed \perp confirmed$ expresses that all tuples have the same value on attribute *confirmed*. For the individual class of IAs, the finite and unrestricted implication problems coincide, they are finitely axiomatizable and decidable in low-degree polynomial time [24]. Besides their attractive computational features, IAs are interesting for a variety of other reasons: (i) Database researchers studied them as early as 1976 [6], with continued interest over the years [11, 18, 24, 31]. (ii) Geiger, Paz, and Pearl studied IAs in a probabilistic setting [15] where they constitute an important fragment of conditional independencies, which form the foundation for Markov and Bayesian networks. (iii) IAs occur naturally in database practice. For example, the cross product between various tables is computed by the FROM clause in SQL. Naturally, a variety of IAs hold on the resulting table. Recently, Olteanu and Zavodny [30] studied succinct representations of relational data by employing algebraic factorizations using distributivity of Cartesian products over unions. Not surprisingly, one of the core enabling notions of the factorizations is that of independence. (iv) In fact, the concept of independence is fundamental to areas as diverse as causality, bound variables in logic, random variables in statistics, patterns in data, the theory of social choice, Mendelian genetics, and even some quantum physics [2, 3]. In a recent response, the study of logics with IAs as atoms of the language has been initiated [16].

Given the usefulness of EMVDs and INDs for data management, given their computational barriers, and given the attractiveness of IAs as a tractable fragment of EMVDs, it is a natural question to ask how IAs and INDs interact. We aim at helping address this current gap in the existing rich theory of relational data dependencies. Adding further to the challenge it is important to note that IAs still form an embedded fragment of EMVDs, in contrast to MVDs which are a class of full dependencies. Somewhat surprisingly, already the interaction of IAs with just keys is intricate [17, 18]. For example, unrestricted implication is finitely axiomatizable but finite implication is not for keys and unary IAs (those with singleton attribute sets), while the finite and unrestricted implication problems coincide and enjoy a finite axiomatization for IAs and unary keys (those with a singleton attribute set).

1.1 Motivating Examples

We use a simplified example to illustrate the interaction between IAs and INDs. For this purpose, consider again our two relation schemata HEART and DISORDER. In addition, the set $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ of INDs and IAs has been specified. Not all constraints need to be enforced strictly, e.g., violations of σ_3 may issue alerts about patients that still have to undergo remaining tests. The INDs σ_1 and σ_2 and the IA σ_3 together finitely imply the IND

$$\sigma = \text{DISORDER}[p_id, t_id] \subseteq \text{HEART}[p_id, t_id] .$$

This interaction is very relevant in practice. While the two UINDs σ_1 and σ_2 do not together imply the IND σ , knowing that the IA σ_3 holds on the referenced schema, tells us that σ also holds on the referencing schema. It may be more natural to specify σ in the first place, instead of specifying σ_1 and σ_2 , but enforcing these two UINDs and the IA σ_3 is more efficient than enforcing the binary IND σ and the IA σ_3 [27].

Another area of impact for our results is query optimization. For illustration purposes, suppose we would like to return the *p_id* of people that have undergone all tests listed for the

class	ui = fi	complexity	finite axiomatization
FD	yes [4]	linear time [5]	yes (2-ary) [4]
IND	yes [8]	PSPACE-complete [8]	yes (2-ary) [8]
IA	yes [15, 24, 31]	cubic time [15, 24]	yes (2-ary) [15, 24, 31]
IND+IA	yes	PSPACE-complete	yes (3-ary)

Table 1: Subclasses of IND+IA. We write “ui” and “fi” for unrestricted and finite implication, respectively.

specific heart disorder we consider. We can express this query in SQL, using double-negation, as follows:

```
SELECT H0.p_id FROM HEART H0
WHERE NOT EXISTS
  SELECT * FROM HEART H1
  WHERE NOT EXISTS
    SELECT * FROM HEART H2
    WHERE H2.t_id = H1.t_id AND
          H2.p_id = H0.p_id ;
```

However, if a query optimizer can notice that the IA σ_3 is implied by the set Σ given above, then the query can be rewritten into

```
SELECT p_id
FROM HEART ;
```

While the set of our constraints is weakly acyclic [14], our query is not “path-conjunctive” and the chase & backchase algorithm from [12] cannot be applied.

1.2 Contributions

In this article we investigate the implication problems for the combined class of IAs and INDs. In particular, we make the following contributions.

1. We show that the finite and unrestricted implication problems coincide and we establish a finite axiomatization.
2. We show that the implication problem is PSPACE-complete.
3. We further show that the implication problem is fixed-parameter tractable in the maximum arity of the given dependencies. As the results 1), 2), and 3) already hold for INDs [7, 8, 22, 23], adding IAs to INDs adds significant expressivity without penalties in terms of computational properties. Such gain without pain cannot be taken for granted as the combined class of independence atoms and keys illustrates [17, 18].
4. For the combined class of IAs and INDs we establish a tractable condition sufficient for their non-interaction. The condition ensures that we can apply known algorithms for deciding implication of the individual classes of IAs and INDs, respectively, to decide implication for an input that combines both individual classes.

Organization. In Section 2 we present all the necessary definitions for the article. Section 3 examines the axiomatic characterization of the combined class of INDs and IAs. Section 4 identifies polynomial-time criteria for the non-interaction between INDs and IAs. Finally,

in Section 5 we discuss the computational complexity of the implication problems considered. Some of the proofs can be found in the appendix.

2 Preliminaries

We usually write A, B, C, \dots for attributes, X, Y, Z, \dots for either sets or sequences of attributes, depending on the context. For two sets (sequences) X and Y , we write XY for their union (concatenation). Similarly, we may write A instead of the single element set or sequence that consists of A . The size of a set (or length of a sequence) X is denoted by $|X|$.

A *relation schema* is a set of attributes A , each with a *domain* $\text{Dom}(A)$, and by a *database schema* we denote a pairwise disjoint sequence of relation schemata. A *tuple* over a relation schema R is a function that maps each $A \in R$ to $\text{Dom}(A)$. A *relation* r over R is a non-empty set of tuples over R . To emphasize that r is a relation over R , we sometimes write $r[R]$. A *database* over a database schema R_1, \dots, R_n is a sequence of relations $(r_1[R_1], \dots, r_n[R_n])$. A *finite relation* over R is a non-empty, finite set of tuples over R , and a *finite database* is a sequence of finite relations. For a tuple t and a relation r over R and $X \subseteq R$, $t(X)$ is the *restriction* of t to X , and $r(X)$ is the set of all restrictions $t(X)$ where $t \in r$. If $X = (A_1, \dots, A_n)$ is a sequence of attributes, then we write $t(X)$ for $(t(A_1), \dots, t(A_n))$.

We exclude empty relations from our definition. This is a practical assumption with no effect when single relation schemata are considered only. However, on multiple relations it has an effect, e.g., the rule $\mathcal{UI}3$ in Table 4 becomes unsound.

Syntax and semantics of FDs, INDs, and IAs are as follows. Let $d = (r_1[R_1], \dots, r_n[R_n])$ be a database. For two sequences of distinct attributes $A_1, \dots, A_n \in R_i$ and $B_1, \dots, B_n \in R_j$, $R_i[A_1 \dots A_n] \subseteq R_j[B_1 \dots B_n]$ is an *inclusion dependency* with semantics defined by $d \models R_i[A_1 \dots A_n] \subseteq R_j[B_1 \dots B_n]$ if for all $t \in r_i$ there is some $t' \in r_j$ such that $t(A_1) = t'(B_1), \dots, t(A_n) = t'(B_n)$. For two (not necessarily disjoint) sets of attributes $X, Y \subseteq R_i$, $R_i : X \perp Y$ is an *independence atom* with semantics: $d \models R_i : X \perp Y$ if for all $t, t' \in r_i$ there exists $t'' \in r_i$ such that $t''(X) = t(X)$ and $t''(Y) = t'(Y)$. For two sets of attributes $X, Y \subseteq R_i$, $R_i : X \rightarrow Y$ is a *functional dependency* with semantics: $d \models R_i : X \rightarrow Y$ if for all $t, t' \in r_i$, $t(X) = t'(X)$ implies $t(Y) = t'(Y)$. We may exclude relation schemata from the notation if they are clear from the context (e.g. write $X \perp Y$ instead of $R_i : X \perp Y$). A *disjoint independence atom* (DIA) is an IA $X \perp Y$ where $X \cap Y$ is empty. We say that an IND is *k-ary* if it is of the form $A_1 \dots A_k \subseteq B_1 \dots B_k$. An IA $X \perp Y$ and an FD $X \rightarrow Y$ are called *k-ary* if $\max\{|X|, |Y|\} = k$. A class of dependencies is called *k-ary* if it contains at most *k-ary* dependencies. We add “U” to a class name to denote its unary subclass, e.g., UIND denotes the class of all unary INDs. Similarly, for $k \geq 2$ we add “k” to a class name to denote its *k-ary* subclass. We use “+” to denote unions of classes, e.g., IND+IA denotes the class of all inclusion dependencies and independence atoms. Note that the semantics of IAs implies:

$$\begin{aligned} * \quad & d \models R_i : X \perp X, \text{ if} \\ & \text{for all } s, s' \in r_i \text{ it holds that } s(X) = s'(X). \end{aligned}$$

Hence, unary FDs of the form $\emptyset \rightarrow A$ and unary IAs of the form $A \perp A$ are also called *constancy atoms* (CAs).

The *restriction* of a dependency σ to a set of attributes R , written $\sigma \upharpoonright R$, is $X \cap R \rightarrow Y \cap R$ for an FD σ of the form $X \rightarrow Y$, and $X \cap R \perp Y \cap R$ for an IA σ of the form $X \perp Y$. If σ is an IND of the form $A_1 \dots A_n \subseteq B_1 \dots B_n$ and i_1, \dots, i_k lists $\{i = 1, \dots, n : A_i \in R \text{ and } B_i \in R\}$, then $\sigma \upharpoonright R = A_{i_1} \dots A_{i_k} \subseteq B_{i_1} \dots B_{i_k}$. For a set of dependencies Σ , the restriction of Σ to R , written $\Sigma \upharpoonright R$, is the set of all $\sigma \upharpoonright R$ where $\sigma \in \Sigma$. Let A and B be attributes from R . By

$\sigma(R : A \mapsto B)$ we denote dependencies obtained from σ by replacing any number of occurrences of A with B .

A set \mathfrak{R} of rules of the form $\sigma_1, \dots, \sigma_n \Rightarrow \sigma$ is called an *axiomatization*. A rule of the previous form is called *n-ary*, and an axiomatization consisting of at most *n-ary* rules is called *n-ary*. A *deduction* from a set of dependencies Σ by an axiomatization \mathfrak{R} is a sequence of dependencies $(\sigma_1, \dots, \sigma_n)$ where each σ_i is either an element of Σ or follows from $\sigma_1, \dots, \sigma_{i-1}$ by an application of a rule in \mathfrak{R} . In such an occasion we write $\Sigma \vdash_{\mathfrak{R}} \sigma$, or simply $\Sigma \vdash \sigma$ if \mathfrak{R} is known.

Given a finite set of database dependencies $\Sigma \cup \{\sigma\}$, the (finite) unrestricted implication problem is to decide whether all (finite) databases that satisfy Σ also satisfy σ , written $\Sigma \models \sigma$ ($\Sigma \models_{\text{fin}} \sigma$). An axiomatization \mathfrak{R} is *sound* for the unrestricted implication problem of a class of dependencies \mathcal{C} if for all finite sets $\Sigma \cup \{\sigma\}$ of dependencies from \mathcal{C} , $\Sigma \vdash_{\mathfrak{R}} \sigma \Rightarrow \Sigma \models \sigma$; it is *complete* if $\Sigma \models \sigma \Rightarrow \Sigma \vdash_{\mathfrak{R}} \sigma$. Soundness and completeness for finite implication are defined analogously.

We assume that all our axiomatizations are attribute-bounded. A sound and complete axiomatization is said to be *attribute-bounded* if it does not introduce new attributes, i.e., any implication of σ by Σ can be verified by a deduction in which only attributes from Σ or σ appear [9]. It is easy to see that a finite (attribute-bounded) axiomatization gives rise to a decision procedure for the associated implication problem. The converse is not necessarily true; join dependencies constitute a class that is associated with a decidable implication problem, yet they lack finite axiomatization [32]. Consider then the class FD+IND+IA. Clearly, both sets $\{(\Sigma, \sigma) \mid \Sigma \models \sigma\}$ and $\{(\Sigma, \sigma) \mid \Sigma \not\models_{\text{fin}} \sigma\}$ are recursively enumerable; the first via reduction to the validity problem of first-order logic, and the second by checking through whether some finite relation satisfies $\Sigma \cup \{\neg\sigma\}$. Consequently, given a subclass \mathcal{C} of FD+IND+IA, the unrestricted and finite implication problems for \mathcal{C} are decidable whenever these two problems coincide.

In our completeness proof we utilize the chase technique (see, e.g., [1]). The chase provides a general tool for reasoning about various dependencies as well as for optimizing conjunctive queries. Given an implication problem for σ by Σ , the starting point of the chase is a simple database falsifying σ . For instance, the chase for independence atoms starts with a unirelational database consisting of two rows that disagree on all attributes. Using some dedicated set of chase rules, this initial database is then completed to another database satisfying Σ . If the new database satisfies also σ , then one concludes that the implication holds. For some classes, such as embedded multivalued dependencies, the chase does not necessarily terminate. In those cases only a semi-decision procedure is obtained.

Axiomatizations. Tables 2, 3, and 4 present the axiomatizations considered in this article. The axiomatization of Table 2 is sound and complete for independence atoms alone [18, 24]. Table 3 depicts the sound and complete axiomatization of inclusion dependencies introduced in [7, 8]. Table 4 presents rules describing interaction between inclusion dependencies and independence atoms.

We conclude this section by stating the soundness of the axioms in Tables 2, 3, and 4. The proof is a straightforward exercise and left to the reader.

Theorem 1. *The axiomatization $\mathfrak{A} \cup \mathfrak{B} \cup \mathfrak{C}$ is sound for the unrestricted and finite implication problems of IND+IA.*

3 IAs+INDs

In this section we establish a set of inference rules that is proven sound and complete for the unrestricted and finite implication problems of IND+IA. This axiomatization consists of rules \mathcal{C} describing interaction between the two classes (see Table 4) and two sets \mathfrak{A} and \mathfrak{B} of complete rules for both classes in isolation (see Tables 2 and 3, resp.). Furthermore, as a consequence of the completeness proof we obtain that the finite and unrestricted implication problems coincide for IND+IA.

$\frac{}{\emptyset \perp X}$ (trivial independence, $\mathcal{I}1$)	$\frac{X \perp Y}{Y \perp X}$ (symmetry, $\mathcal{I}2$)
$\frac{X \perp YZ}{X \perp Y}$ (decomposition, $\mathcal{I}3$)	$\frac{X \perp Y \quad XY \perp Z}{X \perp YZ}$ (exchange, $\mathcal{I}4$)
$\frac{X \perp Y \quad Z \perp Z}{X \perp YZ}$ (weak composition, $\mathcal{I}5$)	

Table 2: Axiomatization \mathfrak{A} for IAs .

We start with the following simplifying lemma which reduces one finite IND+IA-implication problem to another that is not associated with any constancy atoms, i.e., IAs of the form $X \perp X$.

Lemma 2. *Let Σ be a set of IAs and INDs over schema R_1, \dots, R_n , and let $\mathcal{C} := \bigcup_{i=1}^n \{A \in R_i \mid \Sigma \vdash R_i : A \perp A\}$. Let Σ_0 and σ_0 be the restrictions of Σ and σ to the attributes not in \mathcal{C} ,*

$\frac{}{R[X] \subseteq R[X]}$ (reflexivity, $\mathcal{U}1$)
$\frac{R[X] \subseteq R'[Y] \quad R'[Y] \subseteq R''[Z]}{R[X] \subseteq R''[Z]}$ (transitivity, $\mathcal{U}2$)
$\frac{R[A_1 \dots A_n] \subseteq R'[B_1 \dots B_n]}{R[A_{i_1} \dots A_{i_m}] \subseteq R'[B_{i_1} \dots B_{i_m}]}$ (projection and permutation, $\mathcal{U}3$) (*) i_j are pairwise distinct and from $\{1, \dots, n\}$

Table 3: Axiomatization \mathfrak{B} for INDs

$\frac{R[X] \subseteq R'[Z] \quad R[Y] \subseteq R'[W] \quad R'[Z \perp W]}{R[XY] \subseteq R'[ZW]}$ (concatenation, $\mathcal{U}1$)
$\frac{R[XY] \subseteq R'[ZW] \quad R'[ZW] \subseteq R[XY] \quad R'[Z \perp W]}{R[X \perp Y]}$ (transfer, $\mathcal{U}2$)
$\frac{R[X] \subseteq R'[Y] \quad R' : Y \perp Y}{R'[Y] \subseteq R[X]}$ (symmetry, $\mathcal{U}3$)
$\frac{R[X] \subseteq R'[Y] \quad R' : Y \perp Y}{R : X \perp X}$ (constancy, $\mathcal{U}4$)
$\frac{R[A] \subseteq R'[C] \quad R[B] \subseteq R'[C] \quad R' : C \perp C \quad \sigma}{\sigma(R : A \mapsto B)}$ (equality, $\mathcal{U}5$)

Table 4: Axiomatization \mathcal{C} for IAs and INDs

and let Σ_1 be obtained from Σ by

- (1) replacing $R_i : X \perp Y \in \Sigma$ with $R_i : X \setminus \mathcal{C} \perp Y \setminus \mathcal{C}$,
- (2) adding $R_i : A_1 \dots A_j \perp A_{j+1} \dots A_m (R_i \setminus \mathcal{C})$, where A_1, \dots, A_m is some enumeration of $R_i \cap \mathcal{C}$ and $j = 1, \dots, m$,
- (3) adding $R_j[B] \subseteq R_i[A]$ if $\Sigma \vdash R_i[A] \subseteq R_j[B] \wedge R_j : B \perp B$.

Then $\Sigma \vdash \Sigma_0 \cup \Sigma_1 \cup \{R_i : R_i \cap \mathcal{C} \perp R_i \cap \mathcal{C} \mid i = 1, \dots, n\}$ and

- (i) σ is an IA: $\Sigma \models_{\text{fin}} \sigma \Rightarrow \Sigma_0 \models_{\text{fin}} \sigma_0$,
- (ii) σ is an IND: $\Sigma \models_{\text{fin}} \sigma \Rightarrow \Sigma_1 \models_{\text{fin}} \sigma$.

Proof. Clearly we have that $\Sigma \vdash \Sigma_0 \cup \Sigma_1 \cup \{R_i : R_i \cap \mathcal{C} \perp R_i \cap \mathcal{C} \mid i = 1, \dots, n\}$. For claim (i) note that any finite database $d = (r_1, \dots, r_n)$ satisfying $\Sigma_0 \cup \{\neg\sigma_0\}$ can be extended to a one satisfying $\Sigma \cup \{\neg\sigma\}$ by replacing in each $r_i \in d$ each tuple t with all tuples t' such that $t'(A) = 0$ for $A \in R_i \cap \mathcal{C}$ and $t'(A) \in \{0, t(A)\}$ for $A \in R_i \setminus \mathcal{C}$, where 0 is a value not appearing in d .

Next we show claim (ii). Assuming a finite database $d' = (r'_1[R_1], \dots, r'_n[R_n])$ satisfying $\Sigma_1 \cup \{\neg\sigma\}$ for σ of the form $R_l[X] \subseteq R_{l'}[Y]$, we construct a finite database $d = (r_1[R_1], \dots, r_n[R_n])$ satisfying $\Sigma \cup \{\neg\sigma\}$. Let $t \in r'_l$ be such that $t(X) \neq t'(Y)$ for all $t' \in r'_{l'}$. Let t_0 be an extension of $t(R_l \cap \mathcal{C})$ to \mathcal{C} such that, for $A \in R_l \cap \mathcal{C}$ and $B \in R_l \cap \mathcal{C}$, $t_0(A) = t(B)$ if $\Sigma \vdash R_l[A] \subseteq R_l[B]$, and otherwise $t_0(A)$ is any value from $r'_l(A)$. Note that we may assume without losing generality that t_0 is well-defined, i.e., for no distinct $B, B' \in R_l \cap \mathcal{C}$, $\Sigma \vdash R_l[B] \subseteq R_l[B']$. For this, define an equivalence class \sim on $R_l \cap \mathcal{C}$ such that $B \sim B'$ if $\Sigma \vdash R_l[B] \subseteq R_l[B']$. Using $\mathcal{UT5}$ it is then straightforward to show that $\Sigma \models_{\text{fin}} \sigma \Rightarrow \Sigma^* \models_{\text{fin}} \sigma^*$ and $\Sigma^* \vdash \sigma^* \Rightarrow \Sigma \vdash \sigma$ where $\Sigma^* \cup \{\sigma^*\}$ is the set of constraints obtained from $\Sigma \cup \{\sigma\}$ by replacing attributes in $R_l \cap \mathcal{C}$ with their equivalence classes.

Now, define $r_i := r'_i(R_i \setminus \mathcal{C}) \times \{t_0(R_i \cap \mathcal{C})\}$, for $i = 1, \dots, n$. Since $t \in r_l$ and $r_{l'} \subseteq r'_l$, by items (2,3) and the construction, we obtain that $d' \not\models R_l[X] \subseteq R_{l'}[Y]$. It is also easy to see by the construction that all IAs in Σ remain true in d . Assume then that $R_i[X_1 \dots X_m] \subseteq R_j[Y_1 \dots Y_m] \in \Sigma$, and let $t \in r_i$. Since $Y_i \in \mathcal{C}$ implies $X_i \in \mathcal{C}$ and $t_0(X_i) = t_0(Y_i)$, we can assume that $Y_1, \dots, Y_m \notin \mathcal{C}$. Hence, $r_j(Y_1 \dots Y_m) = r'_j(Y_1 \dots Y_m)$. Again, $r_i \subseteq r'_i$ by (2,3) and the construction, and $d' \models R_i[X_1 \dots X_m] \subseteq R_j[Y_1 \dots Y_m]$; hence we obtain that $d \models R_i[X_1 \dots X_m] \subseteq R_j[Y_1 \dots Y_m]$. This concludes case (ii) and the proof. \square

The following lemma will be also helpful in the sequel.

Lemma 3. $XY \perp UV$ can be deduced from $XU \perp YV$, $X \perp U$, and $Y \perp V$ by rules $\mathcal{I2}, \mathcal{I3}, \mathcal{I4}$.

Proof. The following deduction shows the claim:

$$\begin{array}{c}
\frac{XU \perp YV}{Y \perp V \quad YV \perp XU} \mathcal{I2} \\
\frac{Y \perp V \quad YV \perp XU}{Y \perp XUV} \mathcal{I4} \\
\frac{Y \perp XUV}{Y \perp UV} \mathcal{I3} \\
\frac{Y \perp UV}{UV \perp Y} \mathcal{I2} \\
\frac{X \perp U \quad XU \perp YV}{X \perp YUV} \mathcal{I4} \\
\frac{X \perp YUV}{YUV \perp X} \mathcal{I2} \\
\frac{UV \perp XY}{XY \perp UV} \mathcal{I4}
\end{array}$$

\square

Using the previous lemmata we can now state the completeness result. The proof is divided into three subcases in which either CA, IND, or IA consequences are considered. By Lemma 2 we may consider only IND+DIA-implication in the latter two cases. These cases are proved by a chase argument that generalizes the completeness proof of IND-axioms presented in [8].

Theorem 4. *The axiomatization $\mathfrak{A} \cup \mathfrak{B} \cup \mathfrak{C}$ is sound and complete for the unrestricted and finite implication problems of IA+IND.*

Proof. By Theorem 1 the axiomatization is sound. For completeness with respect to both implication problems, it suffices to show that finite implication entails derivability. For this, notice that unrestricted implication entails finite implication. Hence, assume that $\Sigma \models_{\text{FIN}} \sigma$ for a finite set $\Sigma \cup \{\sigma\}$ of IAs and INDs over database schema R_1, \dots, R_n . Let $\mathcal{C} := \bigcup_{i=1}^n \{A \in R_i \mid \Sigma \vdash R_i : A \perp A\}$. By $\mathcal{I}1 - \mathcal{I}3$ we may assume without losing generality that σ is either a CA, a DIA, or an IND. Hence, we show that $\Sigma \vdash \sigma$ in these three cases.

1) σ is a constancy atom. Assume that σ is of the form $R_l : A \perp A$, and assume to the contrary that $\Sigma \not\vdash \sigma$. First let \mathcal{I} be the set of attributes B for which there is $i = 1, \dots, n$ such that $\Sigma \vdash R_i[A] \subseteq R_i[B]$. Then let $d = (r_1, \dots, r_n)$ be the database where $r_i := {}^{R_i \cap \mathcal{I}}\{0, 1\} \times {}^{R_i \setminus \mathcal{I}}\{0\}$. We show that $d \models \Sigma \cup \{\neg\sigma\}$ which contradicts the assumption that $\Sigma \models_{\text{FIN}} \sigma$. It is easy to see that d satisfies $\neg\sigma$, so assume that $R_i[A_1 \dots A_m] \subseteq R_j[B_1 \dots B_m] \in \Sigma$. Since $A_i = A_j \Rightarrow B_i = B_j$ by the assumption and $A_i \in \mathcal{I} \Rightarrow B_i \in \mathcal{I}$ by $\mathcal{U}2$, it follows by the construction that $d \models R_i[A_1 \dots A_m] \subseteq R_j[B_1 \dots B_m]$.

Assume then that $R_i : XZ \perp YZ \in \Sigma$. By the construction, $d \models R_i : X \perp Y$, so it suffices to show that $d \models R_i : \emptyset \rightarrow B$, for $B \in Z$. If $d \not\models R_i : B \perp B$, then by the construction $\Sigma \vdash R_l[A] \subseteq R_l[B]$. Moreover by $\mathcal{U}3$, $\Sigma \vdash R_i[B] \subseteq R_l[A]$, and by $\mathcal{U}4$, $\Sigma \vdash R_l : A \perp A$, contrary to the assumption. Hence $d \models R_i : B \perp B$ which concludes the proof of $d \models \Sigma \cup \{\neg\sigma\}$ and the case of σ being a constancy atom.

2) σ is a disjoint independence atom. Assume that σ is a DIA of the form $R_l : A_1 \dots A_h \perp A_{h+1}, \dots, A_{h+k}$. By Lemma 2 we may assume that Σ is a set of DIAs and INDs. Define first a database $d = (r_1[R_1], \dots, r_n[R_n])$ such that

- $r_l = \{s, s'\}$ where s and s' map all attributes in R_l to 0 except that $s(A_i) = i$ for $i = 1, \dots, h$ and $s'(A_i) = i$ for $i = h + 1, \dots, h + k$;
- $r_i = \{t\}$ where t maps all attributes in R_i to 0, for $i \neq l$.

The idea is to extend d to a database $d'' = (r''_1, \dots, r''_n)$ such that $d'' \models \Sigma$ and

- * if $t \in r''_i$ is such that $t(B_1) = i_1, \dots, t(B_m) = i_m$ and $0 < i_1 < \dots < i_m$, then $\Sigma \vdash R_l[A_{i_1} \dots A_{i_m}] \subseteq R_i[B_1 \dots B_m]$.

We let d'' be the result of chasing d by Σ over the following two chase rules, i.e., d'' is obtained by applying rules (i-ii) to d repeatedly until this is no more possible.

- (i) Assume that $R[Y] \subseteq R'[Z] \in \Sigma$ and $t \in r[R]$ is such that for no $t' \in r'[R']$, $t(Y) = t'(Z)$. Then extend r' with t_{new} that maps Z pointwise to $t(Y)$ and otherwise maps attributes in R' to 0.
- (ii) Assume that $R : Y \perp Z \in \Sigma$ and $t_0, t_1 \in r[R]$ are such that for no $t_2 \in r[R]$, $t_2(Y) = t_0(Y)$ and $t_2(Z) = t_1(Z)$. Then extend r with t_{new} that agrees with t_0 on Y , with t_1 on Z , and maps every other attribute in R to 0.

Note that since the range of the assigned values is finite, the process terminates. Hence, d'' is a finite model of Σ , and therefore by the assumption it satisfies σ . It is also straightforward to verify, using $\mathcal{U}1, \mathcal{U}3$ at the initial stage, $\mathcal{U}2, \mathcal{U}3$ in items (i,ii), and $\mathcal{I}2, \mathcal{I}3, \mathcal{U}1$ in item (ii), that * is satisfied.

Since d'' satisfies σ , we find a tuple $t \in r''_l$ mapping A_i to i , for $i = 1, \dots, n$. Hence, it suffices to show that, given any sequence $\vec{d} = (d_1, \dots, d_m)$, where $d_1 = d$ and d_{i+1} is obtained from d_i by applying (i) or (ii), and any $X \subseteq \{1, \dots, h + k\}$, if there is $t \in r_l$, for $d_m = (r_1, \dots, r_n)$, such that $t(A_i) = i$ if $i \in X$, then $\Sigma \vdash R_l : X \cap A_1 \dots A_h \perp X \cap A_{h+1} \dots A_{h+k}$. We show this claim

by induction on the number of applications of (ii) in \vec{d} .

Assume first that no application of (ii) occurs. Then t cannot combine any values from both s and s' and hence S cannot intersect both $\{1, \dots, h\}$ and $\{h+1, \dots, h+k\}$. Consequently, $R_l : X \cap A_1 \dots A_h \perp X \cap A_{h+1} \dots A_{h+k}$ is derivable by $\mathcal{I}1, \mathcal{I}2$.

Let us then show the induction step. We prove the claim for $S = \{1, \dots, h+k\}$; the general case is then analogous. Assume that the number of applications of (ii) in $\vec{d} = (d_1, \dots, d_m)$ is greater than 0 and let d_k be the database obtained by applying (ii) for the last time, say with regards to some $R : X \perp Y$ and tuples t_0, t_1, t_{new} . We start by defining tuples t_i, t'_i and databases d'_i , for $i = k, \dots, m$. First let t_i be the new tuple introduced in d_i . Note that then t_k is t_{new} from step k . In addition, for $i = k$, define t'_k as t_0 from step k , and d'_k as d_{k-1} . If $i = k+1, \dots, m$, define d'_i and t'_i inductively as follows:

- If d_i is obtained by applying (i) to some t_j listed in t_k, \dots, t_{i-1} (or t not listed in t_k, \dots, t_{i-1}), then let d'_i be obtained by applying (i) to t'_j (or t), and let t'_i be the new tuple added to d'_i . If applying (i) above is not possible, due to the condition given in (i), then define d'_i as d'_{i-1} , and t'_i as any tuple t' violating the condition.

We now let $\vec{d}' := (d_1, \dots, d_{k-1}, d'_k, \dots, d'_m)$. We may assume that $t(A_1 \dots A_{h+k}) \notin r'_l(A_1 \dots A_{h+k})$, for $d'_m = (r'_1, \dots, r'_n)$; otherwise the claim follows by the induction assumption. Since $d_{k-1} \subseteq d'_m$, an analogous claim holds for d_{k-1} . Therefore, and since only (i) is applied after step k , (d_k, \dots, d_m) must include a (possibly empty) chain of applications of (i) copying $1, \dots, h+k$ from $t_k(B_1), \dots, t_k(B_{h+k})$ to $t(A_1), \dots, t(A_{h+k})$, for some $B_1, \dots, B_{h+k} \in XY$. Note that the values are copied from t_k because otherwise one finds $t(A_1 \dots A_{h+k})$ from $r'_l(A_1 \dots A_{h+k})$ contrary to the assumption. Now, using repeatedly $\mathcal{U}2, \mathcal{U}3$ we obtain that

$$\Sigma \vdash R_i[B_1 \dots B_{h+k}] \subseteq R_l[A_1 \dots A_{h+k}]. \quad (1)$$

Moreover, since $*$ is satisfied with regards to d_m , we conclude that

$$\Sigma \vdash R_l[A_1 \dots A_{h+k}] \subseteq R_i[B_1 \dots B_{h+k}]. \quad (2)$$

Let $B_{i_1} \dots B_{i_a} B_{i_{a+1}} \dots B_{i_b} B_{i_{b+1}} \dots B_{i_c} B_{i_{c+1}} \dots B_{i_d}$ relist $B_1 \dots B_{h+k}$ so that

- $B_{i_1}, \dots, B_{i_b} \in X$ and $B_{i_{b+1}}, \dots, B_{i_d} \in Y$
- $\{i_1, \dots, i_a, i_{b+1}, \dots, i_c\} = \{1, \dots, h\}$ and $\{i_{a+1}, \dots, i_b, i_{c+1}, \dots, i_d\} = \{h+1, \dots, h+k\}$.

Since $\Sigma \vdash R_i : B_{i_1} \dots B_{i_b} \perp B_{i_{b+1}} \dots B_{i_d}$ by $\mathcal{I}2, \mathcal{I}3$, we obtain using (1), (2) and $\mathcal{U}3, \mathcal{U}I2$ that

$$\Sigma \vdash R_l : A_{i_1} \dots A_{i_b} \perp A_{i_{b+1}} \dots A_{i_d}. \quad (3)$$

By construction, (d'_k, \dots, d'_m) copies i_1, \dots, i_b from the sequence $t'_k(B_{i_1}), \dots, t'_k(B_{i_b})$ to $t'(A_{i_1}), \dots, t'(A_{i_b})$, for some $t' \in r'_l$. Since \vec{d}' contains less applications of (ii) than \vec{d} , it follows by the induction assumption that $\Sigma \vdash R_l : A_{i_1} \dots A_{i_a} \perp A_{i_{a+1}} \dots A_{i_b}$. It follows by an analogous argument that $\Sigma \vdash R_l : A_{i_{b+1}} \dots A_{i_c} \perp A_{i_{c+1}} \dots A_{i_d}$. By Lemma 3, these two and (3) imply that $\Sigma \vdash R : A_1 \dots A_h \perp A_{h+1} \dots A_{h+k}$. This concludes the induction proof and the case of σ being a disjoint IA.

3) σ is an inclusion dependency. Assume that σ is an IND of the form $R_l[X] \subseteq R_{l'}[Y]$. By Lemma 2 we may assume that Σ is a set of IAs and disjoint INDs. We let $d = (r_1, \dots, r_n)$ where $r_1, \dots, r_{l-1}, r_{l+1}, \dots, r_n$ are single rows of 0's, and $r_l = \{s\}$ for a tuple $s : A_i \mapsto i$, where $R_l = \{A_1, \dots, A_m\}$. It suffices then to chase d by Σ with rules (i,ii), and show that the resulting database d'' satisfies $*$. Since this is analogous to the previous case, we omit the proof here. \square

We obtain the following corollaries. For the first corollary, note that in the last two cases of the previous proof none of the rules $\mathcal{I}5, \mathcal{UI}3, \mathcal{UI}4, \mathcal{UI}5$ are applied. The second corollary follows directly from the previous theorem which shows that the same axiomatization characterizes both implication problems.

Corollary 5. *The axiomatization $\{\mathcal{I}1, \mathcal{I}2, \mathcal{I}3, \mathcal{I}4\} \cup \{\mathcal{UI}1, \mathcal{UI}2\} \cup \mathfrak{B}$ is sound and complete for the unrestricted and finite implication problems of $\text{DIA} + \text{IND}$.*

Corollary 6. *The finite and unrestricted implication problems of $\text{IND} + \text{IA}$ coincide.*

One consequence of Theorem 4 is that the implication problem for $\text{UIND} + \text{CA}$ by $\text{IND} + \text{IA}$ can be determined by considering only interaction between UINDs and CAs. For a set of dependencies Σ , define $\Sigma_{\text{CA}} := \{A \perp A \mid R : AX \perp AY \in \Sigma\}$ and $\Sigma_{\text{UIND}} := \{R[A_i] \subseteq R'[B_i] \mid R[A_1 \dots A_n] \subseteq R'[B_1 \dots B_n] \in \Sigma, i = 1, \dots, n\}$. The following theorem now formulates this idea.

Theorem 7. *Let Σ be a set of INDs and IAs, and let σ be a UIND or a CA. The following are equivalent:*

- (1) $\Sigma \models \sigma$,
- (2) $\Sigma_{\text{UIND}} \cup \Sigma_{\text{CA}} \models \sigma$,
- (3) σ is derivable from $\Sigma_{\text{UIND}} \cup \Sigma_{\text{CA}}$ by $\mathcal{U}1, \mathcal{U}2, \mathcal{UI}3, \mathcal{UI}4$.

Proof. It is clear that (3) \Rightarrow (2) \Rightarrow (1). We show that (1) \Rightarrow (3). By Theorem 4, there is a deduction $(\sigma_1, \dots, \sigma_m)$ from Σ by $\mathfrak{A} \cup \mathfrak{B} \cup \mathfrak{C}$ such that $\sigma_m = \sigma$. It is a straightforward induction to show that for all $i = 1, \dots, m$:

- If σ_i is $R : A \perp A$, then σ_i satisfies (3).
- If σ_i is $R[A_1 \dots A_n] \subseteq R'[B_1 \dots B_n]$, then $\sigma_j := R[A_j] \subseteq R'[B_j]$ satisfies (3), for $j = 1, \dots, n$.

It is worth noting that every application of $\mathcal{UI}5$, where $\sigma(R : A \mapsto B)$ is a UIND or CA, can be simulated by $\mathcal{U}2, \mathcal{UI}3, \mathcal{UI}4$. All the other cases are straightforward and left to the reader. \square

4 Polynomial-Time Conditions for Non-Interaction

The interaction-freeness between the class $\text{FD} + \text{IND}$ has been well-studied in the literature [25, 26]. Here, we examine the frontiers for tractable reasoning about the class $\text{IND} + \text{IA}$. The idea is to establish sufficient criteria for the non-interaction between IAs and INDs. We define non-interaction between two classes as follows.

Definition 8. *Let Σ_0 and Σ_1 be two sets of dependencies from classes \mathcal{C}_0 and \mathcal{C}_1 , respectively. We say that Σ_0, Σ_1 have no interaction with respect to unrestricted (finite) implication if*

- for σ from \mathcal{C}_0 , σ is (finitely) implied by Σ_0 iff σ is (finitely) implied by $\Sigma_0 \cup \Sigma_1$.
- for σ from \mathcal{C}_1 , σ is (finitely) implied by Σ_1 iff σ is (finitely) implied by $\Sigma_0 \cup \Sigma_1$.

Let us now define two syntactic criteria for describing non-interaction. We say that an IA $X \perp Y$ splits an IND $Z \subseteq W$ if both $X \cap W$ and $Y \cap W$ are non-empty. We show that lacking splits implies non-interaction for $\text{IND} + \text{IA}$. The proof is rather straightforward by using the complete axiomatization.

Theorem 9. *Let Σ_{IND} and Σ_{IA} be respectively sets of INDs and IAs. If no IA in Σ_{IA} splits any IND in Σ_{IND} , then Σ_{IND} and Σ_{IA} have no interaction with respect to unrestricted (finite) implication.*

Proof. Assume that σ is implied by $\Sigma_{\text{IND}} \cup \Sigma_{\text{IA}}$ (recall that here finite and unrestricted implication coincide). By Theorem 4, σ can be deduced from $\Sigma_{\text{IND}} \cup \Sigma_{\text{IA}}$ by $\mathfrak{A} \cup \mathfrak{B} \cup \mathfrak{C}$. Given the condition, no rules from \mathfrak{C} can be applied in the deduction. Since only rules in \mathfrak{B} (in \mathfrak{A}) produce fresh INDs (IAs), the claim follows. \square

5 Complexity Results

Next we examine the computational complexity of the discussed implication problems. We show that adding IAs to the class of INDs involves no trade-off in terms of losing desirable computational properties. Theorem 16 shows that, alike for INDs [8], the implication problem for IND+IA is PSPACE-complete. First, however, we show that all unary INDs and constancy atoms implied by a set of INDs and IAs can be recognized in linear time. This follows by Theorem 12, presented in [10] for a more general class of embedded implicational dependencies and UINDs.

Definition 10 ([10]). *For any set Δ of IAs and set Σ of UINDs, we define the set Y called the singlevalued span of Δ and Σ to be the minimum set of attributes Y that satisfies the two conditions:*

- (1) *if $\Delta \cup \{Y \perp Y\} \models A \perp A$, then add A to Y ,*
- (2) *if attribute B is in Y and $A \subseteq B$ is in Σ , then add A to Y .*

Definition 11 ([10]). *For any set A of IAs, any set Z of UINDs, and Y the singlevalued span of Δ and Σ , we define the sets Δ'' and Σ'' , called the unrestricted extensions of Δ and Σ , by $\Delta'' = \Delta \cup \{Y \perp Y\}$ and $\Sigma'' = \Sigma \cup \{A \subseteq B : B \subseteq A \text{ in } \Sigma, A \text{ in } Y\}$.*

Theorem 12 ([10]). *Let Δ be a set of IAs, Σ set of UINDs, Y the singlevalued span, and Δ'', Σ'' the unrestricted extensions of Δ, Σ . For any IA δ and any UIND σ , we have*

- $\Delta \cup \Sigma \models \sigma \Leftrightarrow \Sigma'' \models \sigma$,
- $\Delta \cup \Sigma \models \delta \Leftrightarrow \Delta'' \models \delta$.

We can now show that implication of unary INDs and constancy atoms by INDs and IAs is linear-time decidable.

Theorem 13. *The unrestricted and finite implication problems for CA+UIND by IND+IA is linear-time decidable.*

Proof. Let Σ be a set of INDs and IAs, and let σ be a UIND and τ a CA. By Theorem 7, $\Sigma \models \rho \Leftrightarrow \Sigma_{\text{UIND}} \cup \Sigma_{\text{CA}} \models \rho$, for $\rho \in \{\sigma, \tau\}$. Let Y be the singlevalued span of $\Sigma_{\text{UIND}} \cup \Sigma_{\text{CA}}$ described in Definition 10, and let $\Sigma_0 := \Sigma_{\text{UIND}} \cup \{A \subseteq B : B \subseteq A \in \Sigma_{\text{UIND}}, A \in Y\}$ and $\Sigma_1 := \Sigma_{\text{CA}} \cup \{\emptyset \rightarrow A : A \in Y\}$. By Theorem 12,

- $\Sigma_{\text{UIND}} \cup \Sigma_{\text{CA}} \models \sigma \Leftrightarrow \Sigma_0 \models \sigma$,
- $\Sigma_{\text{UIND}} \cup \Sigma_{\text{CA}} \models \tau \Leftrightarrow \Sigma_1 \models \tau$.

The singlevalued span Y and the deductive closure of Σ_0 can be computed in linear time by reducing to graph reachability. For the latter, note that $\mathcal{U}1, \mathcal{U}2$ form a complete axiomatization for UINDs. Moreover, $\Sigma_1 \models \emptyset \rightarrow A$ iff $A \in Y$. We conclude that in both cases implication can be tested in linear time. \square

Next we turn to the full class IND+IA and use graphs again to characterize the associated implication problem. Recall by Corollary 6 that the unrestricted and finite implication problems coincide for IND+IA.

Definition 14. Let $\Sigma \cup \{\sigma\}$ be a set of INDs and IAs, and assume that σ is of the form $R[X] \subseteq S[Y]$ (or $R[X_1 \perp X_2]$ where $X = X_1 X_2$). Then we let $H_{\Sigma, \sigma}$ be a graph that has nodes $\{\tau_1, \dots, \tau_k\}$, for $k \leq |X|$, where τ_i is an IND of the form $R[A_i] \subseteq R'[B_i]$ and the concatenation $A_1 \dots A_n$ is a permutation (without repetition) of X . Two nodes v, v' are connected by a directed edge $v \rightarrow v'$ if one of the following holds:

- There exists τ_0, τ_1, τ such that $v \setminus \{\tau\} = v' \setminus \{\tau_0, \tau_1\}$ where τ is a permutation of $R[U_0 U_1] \subseteq R'[V_0 V_1]$, $\tau_0 = R[U_0] \subseteq R'[V_0]$, $\tau_1 = R[U_1] \subseteq R'[V_1]$.
- There exists τ_0, τ_1, τ such that $v \setminus \{\tau_0, \tau_1\} = v' \setminus \{\tau\}$ where $\tau_0 = R[U_0] \subseteq R'[V_0]$, $\tau_1 = R[U_1] \subseteq R'[V_1]$, $\tau = R[U_0 U_1] \subseteq R'[V_0 V_1]$, and for some $W_0 \supseteq V_0$ and $W_1 \supseteq V_1$, $R'[W_0 \perp W_1] \in \Sigma$.
- There exists τ, τ' such that $v \setminus \{\tau\} = v' \setminus \{\tau'\}$ where $\tau = R[U] \subseteq R'[V]$, $\tau' = R[U] \subseteq R''[W]$, and $R'[V] \subseteq R''[W]$ is a projection and permutation of some IND in Σ .

If σ is $R[X_1 \perp X_2]$, then we define $v_{\text{start}} := \{R[X_1] \subseteq R[X_1], R[X_2] \subseteq R[X_2]\}$ and $v_{\text{end}} := \{R[X] \subseteq R[X]\}$. If σ is $R[X] \subseteq S[Y]$, then $v_{\text{start}} := \{R[X] \subseteq R[X]\}$ and $v_{\text{end}} := \{R[X] \subseteq S[Y]\}$.

Lemma 15. Let $\Sigma \cup \{\sigma\}$ be a set of INDs and DIAs. Then $\Sigma \models \sigma$ iff $H_{\Sigma, \sigma}$ contains a directed path from v_{start} to v_{end} .

Proof. Assuming $\Sigma \models \sigma$, the required path is found by backtracking a successful chase of d by Σ where the chase rules and d are defined as in cases 2) and 3) in the proof of Theorem 4.

For the other direction, let d be a database satisfying Σ . Assume first that σ is an IA of the form $R[X_1 \perp X_2]$, and let $t, t' \in r[R]$. Now X_1 and X_2 are disjoint, so we can define a mapping t_0 that agrees with t on X_1 and with t' on X_2 . It suffices to show, given a directed path in $H_{\Sigma, \sigma}$ from v_{start} to $\{\tau_1, \dots, \tau_k\}$, that for each τ_i of the form $R[U_i] \subseteq R'[V_i]$ there is $t_i \in r'[R']$ such that $t_0(U_i) = t_i(V_i)$. Since this is a straightforward induction, we leave the proof to the reader. The case where σ is an IND is analogous. \square

PSPACE-completeness of the IND+IA-implication is now showed by reducing to graph reachability in $H_{\Sigma, \sigma}$.

Theorem 16. The unrestricted (finite) implication problem for IND+IA is complete for PSPACE.

Proof. The lower bound follows by the fact that the implication problem for INDs alone is PSPACE-complete [8]. For the upper bound, let $\Sigma \cup \sigma$ be a set of INDs and IAs where σ is an IA (or an IND). Construct first $\Sigma_0 \cup \{\sigma_0\}$ (Σ_1) as described in Lemma 2. By Theorem 13 this can be done in polynomial time. Then non-deterministically check whether there is a directed path in H_{Σ_0, σ_0} ($H_{\Sigma_1, \sigma}$) from v_{start} to v_{end} . Since this requires only polynomial amount of space, we conclude by Savitch' theorem that the implication problem is in PSPACE. \square

Note that there are only polynomially many nodes in $H_{\Sigma, \sigma}$, given that σ is of fixed arity. Hence, by Lemma 2 and Theorem 13 we obtain the following corollary.

Corollary 17. The unrestricted (finite) implication problem for σ by Σ , where $\Sigma \cup \sigma$ is a set of INDs and IAs, is fixed-parameter tractable in the arity of σ .

Actually, the choice of the parameter in the above corollary is not optimal. By Theorem 13, tractability is preserved if only the number of non-constant attributes in an IA σ is fixed. Moreover, assume that σ is an IND of the form $A_1 \dots A_{h+k} \subseteq B_1 \dots B_{h+k}$ where B_i is constant for $i \leq h$. Then the implication problem for σ is fixed-parameter tractable in k since by $\mathcal{UI1}$ it suffices to test whether $A_{h+1} \dots A_{h+k} \subseteq B_{h+1} \dots B_{h+k}$ and $A_i \subseteq B_i$, for $i \leq h$, are all implied.

6 Conclusion and Outlook

We have investigated the implication problem of the combined class of independence atoms and inclusion dependencies. We have shown that the finite and unrestricted versions of this problem coincide, are finitely axiomatizable, PSPACE-complete, and fixed-parameter tractable in their arity. The results retain the computational properties of the individual class of inclusion dependencies but add significant expressivity by the addition of independence atoms. This cannot be taken for granted, as the combined class of independence atoms and keys exhibits [17, 18], for example. We have further established an efficient condition that guarantees the non-interaction of independence atoms with inclusion dependencies. The condition ensures that we can apply known algorithms for deciding implication of the individual classes of independence atoms and inclusion dependencies, respectively, to decide implication for an input that combines both individual classes.

In future work we will investigate the same problems for the combined class of functional dependencies, independence atoms, and inclusion dependencies. This will both be interesting and challenging as the implication problems for the class of functional and inclusion dependencies deviate in the finite and unrestricted case, and are each undecidable [9, 28, 29].

References

- [1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
- [2] Samson Abramsky. Contextual semantics: From quantum mechanics to logic, databases, constraints, and complexity. *Bulletin of the EATCS*, 113, 2014.
- [3] Samson Abramsky, Georg Gottlob, and Phokion G. Kolaitis. Robust constraint satisfaction and local hidden variables in quantum mechanics. In *IJCAI*, 2013.
- [4] William W. Armstrong. Dependency Structures of Data Base Relationships. In *Proc. of IFIP World Computer Congress*, pages 580–583, 1974.
- [5] Catriel Beeri and Philip A. Bernstein. Computational problems related to the design of normal form relational schemas. *ACM Trans. Database Syst.*, 4(1):30–59, 1979.
- [6] Jean-Marc Cadiou. On semantic issues in the relational model of data. In *MFCS*, pages 23–38, 1976.
- [7] Marco A. Casanova, Ronald Fagin, and Christos H. Papadimitriou. Inclusion dependencies and their interaction with functional dependencies. In *PODS*, pages 171–176, 1982.
- [8] Marco A. Casanova, Ronald Fagin, and Christos H. Papadimitriou. Inclusion dependencies and their interaction with functional dependencies. *J. Comput. Syst. Sci.*, 28(1):29–59, 1984.
- [9] Ashok K. Chandra and Moshe Y. Vardi. The implication problem for functional and inclusion dependencies is undecidable. *SIAM Journal on Computing*, 14(3):671–677, 1985.
- [10] Stavros S. Cosmadakis, Paris C. Kanellakis, and Moshe Y. Vardi. Polynomial-time implication problems for unary inclusion dependencies. *J. ACM*, 37(1):15–46, 1990.
- [11] Claude Delobel. Normalization and hierarchical dependencies in the relational data model. *ACM Trans. Database Syst.*, 3(3):201–222, 1978.

- [12] Alin Deutsch, Lucian Popa, and Val Tannen. Physical data independence, constraints, and optimization with universal plans. In *VLDB*, pages 459–470, 1999.
- [13] Ronald Fagin. Multivalued dependencies and a new normal form for relational databases. *ACM Transactions on Database Systems*, 2:262–278, September 1977.
- [14] Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.
- [15] Dan Geiger, Azaria Paz, and Judea Pearl. Axioms and algorithms for inferences involving probabilistic independence. *Information and Computation*, 91(1):128–141, 1991.
- [16] Erich Grädel and Jouko A. Väänänen. Dependence and independence. *Studia Logica*, 101(2):399–410, 2013.
- [17] Miika Hannula, Juha Kontinen, and Sebastian Link. On independence atoms and keys. In *CIKM*, pages 1229–1238, 2014.
- [18] Miika Hannula, Juha Kontinen, and Sebastian Link. On the finite and general implication problems of independence atoms and keys. *J. Comput. Syst. Sci.*, 82(5):856–877, 2016.
- [19] Christian Herrmann. On the undecidability of implications between embedded multivalued database dependencies. *Information and Computation*, 122(2):221 – 235, 1995.
- [20] Christian Herrmann. Corrigendum to ”On the undecidability of implications between embedded multivalued database dependencies”. *Inf. Comput.*, 204(12):1847–1851, 2006.
- [21] Paris C. Kanellakis. Elements of relational database theory. In *Handbook of Theoretical Computer Science*, pages 1073–1156. 1990.
- [22] Henning Köhler and Sebastian Link. Inclusion dependencies reloaded. In *CIKM*, pages 1361–1370, 2015.
- [23] Henning Köhler and Sebastian Link. Inclusion dependencies and their interaction with functional dependencies in SQL. *J. Comput. Syst. Sci.*, 85:104–131, 2017.
- [24] Juha Kontinen, Sebastian Link, and Jouko A. Väänänen. Independence in database relations. In *WoLLIC*, pages 179–193, 2013.
- [25] Mark Levene and George Loizou. How to prevent interaction of functional and inclusion dependencies. *Inf. Process. Lett.*, 71(3-4):115–125, 1999.
- [26] Mark Levene and George Loizou. Guaranteeing no interaction between functional dependencies and tree-like inclusion dependencies. *Theor. Comput. Sci.*, 254(1-2):683–690, 2001.
- [27] Mozhgan Memari and Sebastian Link. Index design for enforcing partial referential integrity efficiently. In *EDBT*, pages 217–228, 2015.
- [28] John C. Mitchell. The implication problem for functional and inclusion dependencies. *Information and Control*, 56(3):154–173, 1983.
- [29] John C. Mitchell. Inference rules for functional and inclusion dependencies. In *PODS*, pages 58–69, 1983.
- [30] Dan Olteanu and Jakub Závodný. Size bounds for factorised representations of query results. *ACM Trans. Database Syst.*, 40(1):2, 2015.
- [31] Jan Paredaens. The interaction of integrity constraints in an information system. *J. Comput. Syst. Sci.*, 20(3):310–329, 1980.
- [32] S. V. Petrov. Finite axiomatization of languages for representations of system properties: Axiomatization of dependencies. *Inf. Sci.*, 47(3):339–372, April 1989.
- [33] Douglas Stott Parker Jr. and Kamran Parsaye-Ghomi. Inferences involving embedded multivalued dependencies and transitive dependencies. In *SIGMOD*, pages 52–57, 1980.
- [34] B Thalheim. *Dependencies in relational databases*. Teubner, 1991.