



Libraries and Learning Services

University of Auckland Research Repository, ResearchSpace

Version

This is the Accepted Manuscript version. This version is defined in the NISO recommended practice RP-8-2008 <http://www.niso.org/publications/rp/>

Suggested Reference

Hannula, M., Kontinen, J., & Link, S. (2016). On the finite and general implication problems of independence atoms and keys. *Journal of Computer & System Sciences*, 82(5), 856-877. doi: [10.1016/j.jcss.2016.02.007](https://doi.org/10.1016/j.jcss.2016.02.007)

Copyright

Items in ResearchSpace are protected by copyright, with all rights reserved, unless otherwise indicated. Previously published items are made available in accordance with the copyright policy of the publisher.

For more information, see [General copyright](#), [Publisher copyright](#), [SHERPA/RoMEO](#).

On the Finite and General Implication Problems of Independence Atoms and Keys[☆]

Miika Hannula^a, Juha Kontinen^a, Sebastian Link^b

^a*Department of Mathematics and Statistics, University of Helsinki, Finland*

^b*Department of Computer Science, University of Auckland, New Zealand*

Abstract

We investigate implication problems for keys and independence atoms in relational databases. For keys and unary independence atoms we show that finite implication is not finitely axiomatizable, and establish a finite axiomatization for general implication. The same axiomatization is also sound and complete for finite and general implication of unary keys and independence atoms, which coincide. We show that the general implication of keys and unary independence atoms and of unary keys and general independence atoms is decidable in polynomial time. For these two classes we also show how to construct Armstrong relations. Finally, we establish tractable conditions that are sufficient for certain classes of keys and independence atoms not to interact.

Keywords: Armstrong relation, Axiomatization, Dependence logic, Finite implication, Implication, Independence, Key

1. Introduction

We study two fundamental classes of integrity constraints in relational databases: Keys and independence atoms. Keys are one of the most important classes of integrity constraints as effective data processing largely depends on the identification of data records. Their importance is manifested in the de-facto industry standard for data management, SQL, and they enjoy native support in every real-world database system. The ultimate goal in database normalization is to reduce the given set of integrity constraints to keys and domain constraints only, as this guarantees the absence of data redundancy from any future database instances that comply with these keys, and therefore allows database systems to process updates efficiently [17]. A relation r satisfies the

[☆]Corresponding author: Juha Kontinen. The first two authors were supported by grants 292767 and 264917 of the Academy of Finland. The research is supported by the Marsden Fund Council from Government funding, administered by the Royal Society of New Zealand.

Email addresses: miika.hannula@helsinki.fi (Miika Hannula),
juha.kontinen@helsinki.fi (Juha Kontinen), s.link@auckland.ac.nz (Sebastian Link)

key $\mathcal{K}(X)$ for a set X of attributes, if for all tuples $t_1, t_2 \in r$ it is true that $t_1 = t_2$ whenever t_1 and t_2 have matching values on all the attributes in X .

Independence atoms (IA) are less known in the database community, but have already been introduced under the term *cross product* by Paredaens in 1980 [58]. While different, independence atoms correspond to marginal probabilistic independence statements well-known in statistics and artificial intelligence. Marginal statements were investigated in depth by Geiger, Paz, and Pearl in 1991 [22]. Independence atoms occur naturally in data processing. A relation r satisfies the independence atom $X \perp Y$ between two sets X and Y of attributes, if for all tuples $t_1, t_2 \in r$ there is some tuple $t \in r$ which matches the values of t_1 on all attributes in X and matches the values of t_2 on all attributes in Y . In other words, in relations that satisfy $X \perp Y$, the occurrence of X -values is independent of the occurrence of Y -values. An interesting special case are IAs of the form $X \perp X$ which is satisfied by a given relation if its projection on X contains at most one tuple. In other words, the relation is constant on X . For a simple example of a general independence atom, consider a database schema that stores information about the enrolment of students into a fixed course. The schema records for each enrolled *student* the *year* in which they completed a *prerequisite* course. Intuitively, every student must have completed every prerequisite for the course in some year. For this reason, for every value in the *student* column and every value in the *prerequisite* column there is some value in the *year* column such that these three values together form a tuple. That is, $student \perp prerequisite$ is a constraint that should hold on every meaningful relation over this schema. One of the most fundamental operators in relational algebra is the Cartesian product (or cross product), combining every tuple from one relation with every tuple from a second relation. In SQL, users must specify this database operation in the form of the **FROM** clause. For a minimal example consider two singleton attribute schemata *part* and *supplier* that we join.

<i>part</i>	<i>supplier</i>	<i>part</i>	<i>supplier</i>
engine	Mercedes	engine	Mercedes
	BMW	engine	BMW

The definition of the Cartesian product entails that the resulting relation satisfies the independence atom $part \perp supplier$. It contains redundant data value occurrences in the sense that changing the value to *any* different value will result in the violation of some given constraint. For instance, changing the second occurrence of ‘engine’ to any other value in the example relation above will violate the independence atom $part \perp supplier$. Independence is therefore a major source of data redundancy, a property that largely determines which queries and updates can be processed efficiently [1, 17, 41, 48, 67]. Independence is thus a fundamental concept in database schema design, exhibited for example, by multivalued dependencies. A relation satisfies a multivalued dependency $X \twoheadrightarrow Y \perp R - XY$ over relation schema R if and only if the relation is the lossless join of its projections on XY and $X(R - XY)$. In other words, for each fixed X -value in the relation, the set of associated Y -values is

independent of the set of associated $R - XY$ -values. Multivalued dependencies correspond to saturated conditional independence atoms [23, 26, 51, 55], and capture a large proportion of the integrity constraints specified in practice. They form the foundation for Fagin’s Fourth Normal form [16]. Due to their fundamental importance in everyday data processing in practice, both keys and independence atoms have also received much research interest since the 1970s [6, 13, 14, 17, 22, 30, 31, 32, 34, 42, 49, 50, 52, 58]. The core reasoning problems of data dependencies are their associated implication problems, with about 100 different classes studied so far [65]. Efficient solutions to these problems have important applications, for example, in database design, query and update processing, data cleaning, exchange, integration and security. Section 2 contains some showcases that illustrate the benefit of such solutions to the processing of updates and queries, as well as data privacy.

Given their importance for data processing in practice, given that keys and independence atoms naturally co-exist and given the long and fruitful history of research into relational data dependencies, it is rather surprising that keys and independence atoms have not been studied together. This is particularly true as more expressive classes of dependencies do not have feasible implication problems. In fact, keys are subsumed by numerical dependencies which do not enjoy a finite axiomatization [25], and independence atoms $Y \perp Z$ are subsumed by embedded multivalued dependencies $X \rightarrow Y \perp Z$ [12, 16, 60] as the special case where $X = \emptyset$, but whose implication problem is not finitely axiomatizable [62] and undecidable [35, 36]. A relation satisfies an embedded multivalued dependency $X \rightarrow Y \perp Z$ if and only if the projection of the relation onto XYZ satisfies the multivalued dependency $X \rightarrow Y \perp Z$. While embedded multivalued dependencies are strongly related to probabilistic conditional independence statements [11], Studeny showed that their associated implication problems are different [63, 64]. Studeny also showed that the implication problem of probabilistic conditional independence statements is not finitely axiomatizable, and the proof relies on a circular system of these statements [63, 64]. These remarks show that independence is also a useful notion for probabilistic approaches to certain machine learning problems [4, 15, 55]. Nevertheless these approaches are different from the independence atoms we study here: We do not consider probabilities but are interested in the notion of independence as a class of data dependencies. There are also expressive classes of data dependencies whose implication problem can be decided efficiently. For example, the combined class of functional and multivalued dependencies enjoys an elegant finite axiomatization and is decidable in almost linear time [3, 20]. These results can be extended to the general implication problem of functional, multivalued, and unary inclusion dependencies. On the other hand, the finite implication problem of functional, multivalued, and unary inclusion dependencies can be decided in cubic time in the input, and while it enjoys an elegant axiomatization it requires one cyclic inference rule for each positive integer [10]. Note that functional dependencies extend keys, but multivalued dependencies are full dependencies and cannot express many independence atoms, which are embedded dependencies. In fact, the intersection of multivalued dependencies and independence atoms consists

of multivalued dependencies of the form $X \twoheadrightarrow Y \perp Z$ where $X = \emptyset$, or in other words, of independence atoms of the form $Y \perp Z$ where the underlying relation schema is the union of Y and Z . Keys and independence atoms in isolation enjoy efficient solutions to computational problems: Finite and general implication problems coincide, and are axiomatizable by finite sets of Horn rules [22, 42, 58, 65]. They thus are excellent candidates to push the frontier of axiomatizable classes of data dependencies.

Motivated by real-world applications and the lack of previous research we initiate research on the interaction of key dependencies and independence atoms. As far as we are aware, keys and independence atoms together constitute the first case in which keys are combined with a class of embedded data dependencies that capture the concept of independence. Somewhat surprisingly, the efficient solutions to computational problems that hold for each class in isolation do not carry over to the combined class, even when independence atoms are restricted to the unary case. In fact, we show that for the combined class of keys and independence atoms:

- The finite and the general implication problem differ from one another.
- For keys and unary independence atoms the general implication problem has a 2-ary axiomatization \mathfrak{J} by Horn rules. Here, IAs $X \perp Y$ where X and Y consist of one attribute only are called unary, and an axiomatization is k -ary when every inference rule has at most k premises.
- The finite implication problem for keys and unary independence atoms is not finitely axiomatizable.

While the last result appears to be discouraging for the database practitioner, our research brings forward very useful applications. In fact, all the inference rules that are sound for general implication are also sound for finite implication. This means, in particular, that our inference rules can be exploited to avoid redundant integrity checks when updates are processed, saving more time as the database grows. Furthermore, query optimizers can exploit our inference rules to optimize efficiently the evaluation of queries. Examples of such applications are given in Section 2.

In view of our second and third main result together, it follows that our 2-ary axiomatization \mathfrak{J} for the general implication problem is somewhat ‘as complete as possible’ for finite implication. It means, in particular, our results provide reassurance that we have exhausted a complete list of ‘cheap’ opportunities to benefit our application at hand, for examples, integrity checking or query optimization. As the third result shows, other opportunities for optimization cannot be captured completely by a simple set of rules, but must be dealt with on a case-by-case basis which requires an analysis that is more costly.

We remark that the rules $\mathcal{R}1 - \mathcal{R}5$ of \mathfrak{J} characterize the implication problem for the class of independence atoms alone [42, 58]. Notably, $\mathcal{R}1, \mathcal{R}2, \mathcal{R}4, \mathcal{R}5$ characterize also the implication problem for the class of marginal probabilistic independence statements [22, Theorem 3], in which the sets of given random

variables are assumed to be disjoint. The extra rule $\mathcal{R3}$ captures our more general notion in which the sets of attributes may intersect non-trivially (also defined in [57]). It is therefore not surprising that our axiomatization \mathfrak{J} shows strong similarities with the *semi-graphoids* which are sound for the implication of general conditional independence atoms [11, 23, 26, 56].

Our results are somewhat similar to those known for the combined class of functional dependencies (FDs) and inclusion dependencies (INDs). While both classes in isolation have matching finite and general implication problems and enjoy finite axiomatizations, the finite and the general implication problems differ for the combined class of FDs and unary INDs already [8]. For FDs and unary INDs the general implication problem has a 2-ary axiomatization by Horn rules [10], while their finite implication problem is not finitely axiomatizable [8]. Interestingly, key dependencies are strictly subsumed by FDs. It is also known that both implication problems are undecidable for FDs and INDs [9, 54], but decidable for FDs and unary INDs [10]. The containment problem for conjunctive queries has also been investigated in the presence of functional and inclusion dependencies [37]. Query answering on inconsistent and incomplete databases with keys and inclusion dependencies has also been studied in [7].

The results, described so far, have been announced in [29]. In the current article, we provide complete proofs and further examples that motivate and illustrate these results. In the following, we explain the new contributions that we added to the current article.

We show that the combined class of unary keys and independence atoms enjoys desirable properties. Firstly, the combined class is important in practice as most real-world database schema designs choose to rely on unary keys only. The thinking is to avoid the specification of natural keys as their enforcement may prohibit the entry of important data that violate the natural key as an exception. Instead, relation schemata are augmented by a single surrogate attribute which serves as a unary (artificial) key. We hasten to point out that natural keys should be dealt with in any case, particularly to ensure consistency with respect to natural keys that capture semantically meaningful business rules. Nevertheless, the reliance on unary keys is a phenomenon that is common practice. Secondly, we show for this class that the associated finite and general implication problems coincide. Thirdly, we show that the axiomatization \mathfrak{J} for the general implication problem of general keys and unary independence atoms is also sound and complete for the implication problem of unary keys and general independence atoms.

We also consider the computational complexity of the general implication problems of keys and unary independence atoms and of unary keys and general independence atoms. By utilizing our complete axiomatization, we show that these implication problems can be decided in polynomial time.

In addition, we establish two conditions that allow us to exploit previous research on the individual class of keys and the individual class of independence atoms in the context of the combined class of keys and independence atoms. The conditions identify situations in which there is no interaction between the given set Σ_K of keys and the given set Σ_I of independence atoms. That is, for every

key φ it holds that $\Sigma_K \cup \Sigma_I$ implies φ if and only if Σ_K implies φ , and for every independence atom φ it holds that $\Sigma_K \cup \Sigma_I$ implies φ if and only if Σ_I implies φ . As our conditions can be verified in polynomial time, instances of the combined implication problems that comply with our condition can be solved efficiently by using tools to decide implication for either keys or independence atoms only. Again, our research fits in with previous research on the combined class of functional and inclusion dependencies, for which several sufficient conditions are known that guarantee no interaction between them [46, 47].

Finally, we investigate the concept of Armstrong relations [18] in our setting. A relation is Armstrong for a given set of constraints from a class \mathcal{C} when it satisfies all of the given constraints but violates all those constraints from \mathcal{C} which are not implied by the given set. In this sense, an Armstrong relation is a concise representation of a given constraint set in the form of a data sample. Armstrong relations have been shown to be useful for the acquisition of semantically meaningful constraints for a given application domain [43]. We show how to construct i) infinite Armstrong relations for every given set of keys and unary independence atoms where implication refers to general implication, and ii) finite Armstrong relations for every given set of independence atoms and unary keys.

Our work originated from recent developments in the area of dependence logic, which constitutes a novel approach to the study of various notions of dependence and independence that is intimately linked with databases and their data dependencies [24, 66]. It has been shown recently, e.g., that the general implication problem of so-called conditional independence atoms and inclusion atoms can be finitely axiomatized in this context [28]. For databases, this result establishes a finite axiomatization (utilizing implicit existential quantification) of the general implication problem for inclusion, functional, and embedded multivalued dependencies taken together. This result is similar to the axiomatization of the general implication problem for FDs and INs [54]. A comparison to our work shows how the axiomatizability for expressive classes of data dependencies can be achieved by relaxing the notion of an axiomatization, while the same class is not axiomatizable with respect to the traditional notion.

Organization. In Section 2, we motivate our research into implication problems for keys and independence atoms with examples from integrity checking, query optimization, and data privacy which can exploit our inference rules to their advantage. We fix the notation for keys, independence atoms, and implication problems in Section 3. A 2-ary finite axiomatization \mathfrak{J} for the general implication problem of keys and unary independence atoms is established in Section 4. The difference between finite and general implication for keys and unary independence atoms, as well as the impossibility of having any k -ary axiomatization for the finite implication problem is established in Section 5. The finite and general implication problems for the class of unary keys and general independence atoms are shown to coincide in Section 6, where we also show that our set \mathfrak{J} of inference rules is sound and complete for these implication problems. The construction of Armstrong relations for sub-classes of keys and independence atoms is discussed in Section 7. In Section 8, we define a poly-

nomial time algorithm for deciding the general implication problem of keys and unary independence atoms. Furthermore, in Section 9 we establish two conditions on the given sets of keys and independence atoms that guarantees that there is no interaction between them. These conditions can be verified in polynomial time. We conclude in Section 10 where we also list some open problems and comment on future work.

2. Motivating Examples

In this section we present three showcases that illustrate how an understanding of the interaction of keys and independence atoms can benefit various areas of data processing. Our complete axiomatizations guarantee us that we can exhaust all implied constraints for these showcases. While this cannot be guaranteed by our incomplete axiomatizations, they still allow us to infer a large number of implied constraints using our inference rules. Any remaining implied constraints that cannot ‘simply’ be inferred by our inference rules must be detected by a case by case analysis.

2.1. Query Optimization

In this section we give some minimal examples that illustrate the use of some inference rules to optimize queries. The examples illustrate how the formalization of some commonsense reasoning in the form of inference rules can benefit query processing.

As a first example, consider a simple relation schema `SUPPLIES` with two attributes `supplier` and `part`. Suppose we want to know for each part how many distinct suppliers supply the part. A naïve SQL query Q would be

```
SELECT    part, COUNT(DISTINCT supplier)
FROM      SUPPLIES
GROUP BY  part ;
```

Here, the command `DISTINCT` is used to eliminate duplicate suppliers. In data processing duplicate elimination is time-consuming and not executed by default. However, duplicate elimination in query Q is redundant for the following reason. The `GROUP BY` clause uses parts to partition the input relation over `SUPPLIES` into sub-relations. That is, the `GROUP BY` clause causes each sub-relation to satisfy the independence atom $part \perp part$, which means that all tuples of the same sub-relation have the same value on `part`. As the input relation over `SUPPLIES` satisfies the key $\mathcal{K}(supplier, part)$, so does each of the sub-relations. However, the key $\mathcal{K}(supplier, part)$ and the independence atom $part \perp part$ together imply the key $\mathcal{K}(supplier)$. Hence, there are no duplicate `suppliers` in any sub-relation and Q can be replaced by the more efficient query Q' .

```
SELECT    part, COUNT(supplier)
FROM      SUPPLIES
GROUP BY  part ;
```


in which no time is wasted on finding non-existing duplicate suppliers in each sub-relation. The example illustrates how special independence atoms of the form $A \perp A$ are already practically relevant: They apply to every sub-relation generated by the `GROUP BY A` in SQL. The fact that the key $\mathcal{K}(\text{supplier}, \text{part})$ and the independence atom $\text{part} \perp \text{part}$ together imply the key $\mathcal{K}(\text{supplier})$ is just an instance of the following sound inference rule

$$\frac{X \perp X \quad \mathcal{K}(XY)}{\mathcal{K}(Y)}.$$

The inference rule formalizes the commonsense reasoning that Y -values are unique ($\mathcal{K}(Y)$) whenever XY -values are unique ($\mathcal{K}(XY)$) and the X -values are constant ($X \perp X$). It is important to observe that this line of reasoning is not limited to SQL-based database systems. In fact, `GROUP BY` clauses are also integral to SPARQL [2] and PIG Latin queries [21], and any reasonable query language that supports aggregation.

For an example with a general independence atom take the relation schema `CAR` with attributes `model`, `vehicle`, `part`. A naïve query that returns for the `model` Ferrari all combinations of `vehicles` and `parts`, we write:

```
SELECT  c1.vehicle, c2.part
FROM    CAR c1, CAR c2
WHERE   c1.model='Ferrari' AND c1.model=c2.model ;
```

Knowing that the IA $\text{vehicle} \perp \text{part}$ holds for each given model (in particular, for Ferrari's), we can rewrite this into:

```
SELECT  vehicle, part
FROM    CAR
WHERE   model='Ferrari' ;
```

saving a join. It is important to stress in this example that this optimization is only possible when the underlying relation satisfies the IA $\text{vehicle} \perp \text{part}$. If this is not the case, then we need the original self-join query to ensure that we also return a combination of vehicles and parts that do not occur together in some tuple of the relation.

The last example showed the importance of independence atoms alone for query optimization. For an example that involves the combined class of keys and independence atoms, consider the relation schema `HEALTH` with attributes `date`, `patient` and `status`. Here, a `HEALTH` record captures the status of a patient on a certain date. For simplicity, we assume that the domain of `status` is either 'admitted' or 'released', that is, we record when a patient has been admitted to a hospital, or released. We may ask the query that returns patients that were admitted and released on the same date:

```
SELECT  DISTINCT date, patient
FROM    HEALTH
GROUP BY  date, patient
HAVING  COUNT(DISTINCT status) > 1 AND
        COUNT(patient) > 1;
```

Here, the clause `HAVING COUNT(DISTINCT status) > 1` ensures that only groups of *date, patient* are considered in which the patient on that date has two different statuses (i.e., admitted and released). In other words, each group of *date, patient* satisfies the independence atom $status \perp patient$. Each group of *date, patient* also violates the independence atom $status \perp status$, since there are two different values present in each group. The second condition `COUNT(patient) > 1` ensures that the number of (duplicate) patients in each group is more than 1, i.e., each group violates the key $\mathcal{K}(patient)$. The constraints $status \perp patient$, $\mathcal{K}(patient)$, and $status \perp status$ form an instance of our inference rule

$$\frac{X \perp Y \quad \mathcal{K}(X)}{Y \perp Y}.$$

Informally, this inference rule formalizes the commonsense reasoning that Y -values are constant ($Y \perp Y$) whenever X -values and Y -values are independent ($X \perp Y$), and Y -values are unique ($\mathcal{K}(Y)$). In fact, as each group satisfies $status \perp patient$ and violates $status \perp status$, it follows by this rule that each group must also violate $\mathcal{K}(patient)$. For this reason, the query above can be simplified to:

```
SELECT    DISTINCT date, patient
FROM      HEALTH
GROUP BY  date, patient
HAVING    COUNT(DISTINCT status) > 1;
```

We emphasize that it may be common sense to not include the condition `COUNT(patient) > 1` in the original query. Nevertheless, the example shows how commonsense reasoning is expressed in the form of an inference rule, and how it can be applied in SQL.

2.2. Integrity Enforcement

Integrity enforcement ensures that updates of a given database result in a new database that complies with every element of the given set Σ of integrity constraints that have been specified to hold. For this purpose, the database system must validate for each integrity constraint σ in Σ whether the new database satisfies σ . This can be a time-consuming task and the larger the database the more time is required. It is therefore important to keep the validation at a minimum necessary, in particular, Σ should not contain any ‘redundant’ constraints σ , i.e. constraints $\sigma \in \Sigma$ for which $\Sigma - \{\sigma\} \models_{(FIN)} \sigma$ holds. Our results concerning the general and finite implication problem for keys and unary independence atoms ensure that we can remove redundant constraints that can be inferred using our inference rules. The negative results we establish on the completeness of an axiomatization for finite implication also apply to the removal of redundant constraints. In particular, for finite implication we cannot expect to use a finite axiomatization to remove all redundant constraints but only those constraints that can be ‘easily’ identified as redundant. In the finite case, the decidability of the problem whether a given set of our constraints is redundant is open. For

the class of unary keys and general independence atoms, we can identify all redundant constraints. As a simple example, if Σ is given by $status \perp patient$, $status \perp status$ and $k(patient)$, then $status \perp status$ can be removed from Σ as it is (finitely) implied by $status \perp patient$ and $k(patient)$. In this example it is also guaranteed that none of the remaining rules is redundant.

2.3. Protecting Privacy under Inference Attacks

Our final example illustrates the significance of keys and independence atoms for data privacy under smart inference attacks. Assume all patients (with some given condition) receive the same set of therapies, i.e. we have an independence atom $patient \perp therapy$ on the schema TREATMENT. If a user asks

```
SELECT  patient, therapy
FROM    TREATMENT
WHERE   patient='Bob' AND therapy='Radiation' ;
```

then direct access to (Bob, Radiation) would be prohibited to some users due to privacy concerns by Bob. However, if a smart user asks the following two queries

```
SELECT  patient
FROM    TREATMENT
WHERE   patient='Bob';
```

and

```
SELECT  therapy
FROM    TREATMENT
WHERE   therapy='Radiation';
```

then access control is bypassed and returns answers that include 'Bob' and 'Radiation', respectively. The user can conclude from both answers that Bob has been treated with Radiation due to the independence atom. Based on the independence atom the same user should only see the correct answer to one of the queries. In addition, if the answer to the first user query contains 'Bob' just once, then the user can infer that 'Radiation' was the *only* therapy that 'Bob' underwent. This is because $k(patient)$ (here in the case where the patient is 'Bob') and $patient \perp therapy$ together imply $therapy \perp therapy$. So, reasoning about independence atoms (and keys) can help detect and prevent inference attacks by smart users.

3. Preliminaries

3.1. Definitions

A *relation schema* R is a set of symbols A called *attributes*, each equipped with a domain $\text{Dom}(A)$ representing the possible values that can occur in the column named A . A tuple t over R is a mapping $R \rightarrow \bigcup_{A \in R} \text{Dom}(A)$ where $t(A) \in \text{Dom}(A)$ for each $A \in R$. For a tuple t over R and $R' \subseteq R$, $t(R')$ is

the restriction of t on R' . A *relation* r over R is a set of tuples t over R . If $R' \subseteq R$ and r is a relation over R , then we write $r(R')$ for $\{t(R') : t \in r\}$. If $A \in R$ is an attribute and r is a relation over R , then we write $r(A = a)$ for $\{t \in r : t(A) = a\}$. For sets of attributes X and Y , we often write XY for $X \cup Y$, and denote singleton sets of attributes $\{A\}$ by A . Also, for a relation schema $A_1 \dots A_n$, a relation $r(A_1 \dots A_n)$ is sometimes identified with the set notation $\{(a_1, \dots, a_n) \mid \exists t \in r : t(A_i) = a_i \text{ for } 1 \leq i \leq n\}$.

3.2. Independence Atoms and Keys

Let R be a relation schema and $X \subseteq R$. Then $\mathcal{K}(X)$ is an *R-key*, given the following semantic rule for a relation r over R :

- $r \models \mathcal{K}(X)$ if and only if for all $t, t' \in r$: if $t(X) = t'(X)$, then $t = t'$.

Let R be a relation schema and $X, Y \subseteq R$. Then $X \perp Y$ is an *R-independence atom*, given the following semantic rule for a relation r over R :

- $r \models X \perp Y$ if and only if for all $t, t' \in r$ there exists a $t'' \in r$ such that $t''(X) = t(X) \wedge t''(Y) = t'(Y)$.

Remark. $r \models X \perp X$ means that only one value exists for X in r .

An independence atom $X \perp Y$ is called unary if X and Y are single attributes. Similarly, a key $\mathcal{K}(A)$ is called unary if A is a single attribute. *R-keys* and *R-independence atoms* are together called *R-constraints*. If Σ is a set of *R-constraints* and $R' \subseteq R$, then we write $\Sigma \upharpoonright R'$ for the subset of all constraints of Σ that only mention attributes contained in R' .

3.3. Implication Problems

For a set $\Sigma \cup \{\phi\}$ of independence atoms and keys we say that Σ *implies* ϕ , written $\Sigma \models \phi$, if every relation that satisfies every element in Σ also satisfies ϕ . We write $\Sigma \models_{\text{FIN}} \phi$, if every finite relation that satisfies every element in Σ also satisfies ϕ . We say that ϕ is a *k-ary* (finite) *implication* of Σ , if there exists $\Sigma' \subseteq \Sigma$ such that $|\Sigma'| \leq k$ and $\Sigma' \models \phi$ ($\Sigma' \models_{\text{FIN}} \phi$).

In this article we consider the axiomatizability of the so-called finite and the general implication problem for independence atoms and keys. The general implication problem for independence atoms and keys is defined as follows.

PROBLEM:	General implication problem for independence atoms and keys
INPUT:	Relation schema R , Set $\Sigma \cup \{\varphi\}$ of independence atoms and keys over R
OUTPUT:	Yes, if $\Sigma \models \varphi$; No, otherwise

The finite implication problem is defined analogously by replacing $\Sigma \models \phi$ with $\Sigma \models_{\text{FIN}} \phi$.

In general for a set \mathfrak{R} of inference rules, we denote by $\Sigma \vdash_{\mathfrak{R}} \phi$ the *inference* of ϕ from Σ . That is, there is some sequence $\gamma = [\sigma_1, \dots, \sigma_n]$ of independence atoms and keys such that $\sigma_n = \phi$ and every σ_i is an element of Σ or results from an application of an inference rule in \mathfrak{R} to some elements in $\{\sigma_1, \dots, \sigma_{i-1}\}$. A set \mathfrak{R} of inference rules is said to be *sound* for the general implication problem of independence atoms and keys, if for every R and for every set Σ , $\Sigma \vdash_{\mathfrak{R}} \phi$ implies that $\Sigma \models \phi$. A set \mathfrak{R} is called *complete* for the general implication problem if $\Sigma \models \phi$ implies that $\Sigma \vdash_{\mathfrak{R}} \phi$. The (finite) set \mathfrak{R} is said to be a (finite) *axiomatization* of the general implication for independence atoms and keys if \mathfrak{R} is both sound and complete. These notions are defined analogously for the finite implication problem. For $k \geq 1$, a rule is called *k-ary* if it is of the form

$$\frac{A_1 \quad A_2 \quad \dots \quad A_{k-1} \quad A_k}{B} \quad (1)$$

A set of inference rules is called *k-ary* if it consists of at most *k-ary* rules. In this paper the set of 2-ary inference rules \mathfrak{J} for IAs and keys, depicted in Figure 1, will be studied. Note that in the context of unary IAs the rules $\mathcal{R}4$ and $\mathcal{R}5$ become redundant. It is also worth noting that the rules $\mathcal{R}1 - \mathcal{R}5$ are sound and complete for the implication problem of IAs alone [42, 57, 58], and the unary rules $\mathcal{R}6$ and $\mathcal{R}7$ are sound and complete for the implication problem of keys alone [65]. It is also straightforward to show that for unary IAs alone the 2-ary set of rules $\mathcal{R}1 - \mathcal{R}3$ form a sound and complete axiomatization which yields a 1-ary axiomatization since the rule $\mathcal{R}3$ can be replaced with

$$\frac{X \perp X}{X \perp Y}.$$

On the other hand, there exists no 1-ary axiomatization for the combined class of keys and unary IAs; evidently the rule $\mathcal{R}9$ does not reduce to any set of 1-ary rules.

4. General Implication

In this section we will show that the set of axioms \mathfrak{J} in Table 1 is sound and complete for the general implication problem of unary independence atoms and arbitrary keys taken together. It is straightforward to check the soundness of the axioms \mathfrak{J} .

Theorem 4.1. *The axioms \mathfrak{J} are sound for the general implication problem of independence atoms and keys.*

Proof. Assume R is a relation schema and Σ consists of R -keys and R -independence atoms. Let r be a relation over R . We will show using induction on the length of derivation $[\sigma_1, \dots, \sigma_n]$ from Σ that if $r \models \Sigma$ then $r \models \sigma_i$ for $1 \leq i \leq n$. We consider only the cases where σ_n has been obtained by Rule $\mathcal{R}8$ or $\mathcal{R}9$.

We consider first the case where σ_n has been obtained by applying Rule $\mathcal{R}8$ to σ_j and σ_k . Then σ_n , σ_j , and σ_k are of the form $\mathcal{K}(Y)$, $X \perp X$, and $\mathcal{K}(XY)$,

$\frac{}{\emptyset \perp X}$ (trivial independence, $\mathcal{R}1$)	$\frac{X \perp Y}{Y \perp X}$ (symmetry, $\mathcal{R}2$)
$\frac{X \perp X \quad Y \perp Z}{XY \perp Z}$ (constancy, $\mathcal{R}3$)	$\frac{X \perp YZ}{X \perp Y}$ (decomposition, $\mathcal{R}4$)
$\frac{X \perp Y \quad XY \perp Z}{X \perp YZ}$ (exchange, $\mathcal{R}5$)	$\overline{\mathcal{K}(R)}$ (trivial key, $\mathcal{R}6$)
$\frac{\mathcal{K}(X)}{\mathcal{K}(XY)}$ (upward closure, $\mathcal{R}7$)	$\frac{X \perp X \quad \mathcal{K}(XY)}{\mathcal{K}(Y)}$ (1st composition, $\mathcal{R}8$)
$\frac{X \perp Y \quad \mathcal{K}(X)}{Y \perp Y}$ (2nd composition, $\mathcal{R}9$)	

Table 1: The Set of Axioms \mathfrak{J} for Independence Atoms and Keys in Database Relations

respectively. By the induction hypothesis, $r \models \sigma_j$ and $r \models \sigma_k$. We will show $r \models \sigma_n$. For that end, let $t_1, t_2 \in r$ such that $t_1(Y) = t_2(Y)$. By the induction hypothesis $r \models X \perp X$ implying that the attribute X has only one value in r . Therefore it holds that $t_1(XY) = t_2(XY)$. Again by assumption $r \models \mathcal{K}(XY)$, thus $t_1 = t_2$ as wanted.

Let us then consider Rule $\mathcal{R}9$. Now σ_n, σ_j , and σ_k are of the form $Y \perp Y$, $X \perp Y$, and $\mathcal{K}(X)$, respectively. By the induction hypothesis, $r \models \sigma_j$ and $r \models \sigma_k$. We will show $r \models \sigma_n$. Let $t_1, t_2 \in r$. We need to show that $t_1(Y) = t_2(Y)$. For a contradiction, assume that $t_1(Y) \neq t_2(Y)$. Since $r \models \sigma_j$, there exists $t_3 \in r$ such that $t_3(X) = t_1(X)$ and $t_3(Y) = t_2(Y)$. Hence there exists $t_1, t_3 \in r$ such that $t_1(X) = t_3(X)$ and $t_1 \neq t_3$ contradicting the assumption that $r \models \mathcal{K}(X)$. Therefore, we must have $t_1(Y) = t_2(Y)$, and $r \models Y \perp Y$, as was to be shown. \square

Next we will show that the set of axioms \mathfrak{J} is complete for the general implication problem of unary independence atoms and arbitrary keys. This result is obtained by an infinite relation construction. Later we will see that no finite counter-example can here be constructed; this is implied by Theorem 5.10.

Theorem 4.2. *Assume that R is a relation schema and $\Sigma \cup \{\phi\}$ consists of R -keys and unary R -independence atoms. Then $\Sigma \vdash_{\mathfrak{J}} \phi$ iff $\Sigma \models \phi$.*

Proof. Assume to the contrary that $\Sigma \not\vdash_{\mathfrak{J}} \phi$. We show the theorem by constructing a countably infinite relation r witnessing $\Sigma \not\models \phi$. Let $\Sigma_1 \cup \Sigma_K$ be the partition

of Σ to independence atoms and keys, respectively. Let $X_1 \perp Y_1, \dots, X_m \perp Y_m$ be an enumeration of Σ_I . Let $R' := \{A \in R : \Sigma \vdash_{\exists} A \perp A\}$. We notice that by the assumption and rules $\mathcal{R}2 - \mathcal{R}4, \mathcal{R}7 - \mathcal{R}8, \Sigma' \not\vdash_{\exists} \phi'$ and $\{A \in R - R' : \Sigma' \vdash_{\exists} A \perp A\} = \emptyset$ where $\Sigma' \cup \{\phi'\}$ is the set of atoms obtained from $\Sigma \cup \{\phi\}$ by projecting to attributes in $R - R'$. It easy to see that then $\Sigma' \not\vdash \phi' \Rightarrow \Sigma \not\vdash \phi$. Hence we may assume without loss of generality that $R' = \emptyset$.

The relation r is now constructed as follows. We first define an increasing chain (with respect to \subseteq) of finite relations r_n , for $n \geq 0$, such that

1. $r_n \models \Sigma_K$ and
2. $r_n \models X_l \perp Y_l$ if $n \geq 1$ and $n = l \pmod{m}$.

Then letting $r := \bigcup_{n \geq 0} r_n$, we obtain that $r \models \Sigma$. Regarding ϕ , we have two cases: ϕ is either of the form

- (i) $\mathcal{K}(X)$ or
- (ii) $X \perp Y$.

We consider the construction of r_n separately in these two cases.

Case (i). Consider first the case where $\phi = \mathcal{K}(X)$ for some $X \subseteq R$. We construct the relations r_n inductively as follows:

The base case. Assume first that $n = 0$. We let $r_0 := \{t_0, t_1\}$ where for all $A \in R$, $t_0(A) := 0$ and

$$\bullet t_1(A) := \begin{cases} 0 & \text{if } A \in X, \\ 1 & \text{otherwise.} \end{cases}$$

Now item 2 is holds trivially. For item 1, if $r_0 \not\models \mathcal{K}(Y)$ for some $\mathcal{K}(Y) \in \Sigma_K$, then $Y \subseteq X$ and hence ϕ is derivable by $\mathcal{R}7$, contrary to the assumption.

The inductive step. Assuming that r_n is a finite relation satisfying items 1-2, we construct a finite relation r_{n+1} satisfying the same conditions. Assume that $l = n + 1 \pmod{m}$. If $r_n \models X_l \perp Y_l$, then we let $r_{n+1} := r_n$. Otherwise, let $(a_1, b_1), \dots, (a_k, b_k)$ be an enumeration of $r_n(X_l) \times r_n(Y_l) \setminus r_n(X_l Y_l)$, and assume that M is the maximal number occurring in r_n . We then let r_{n+1} be obtained by extending r_n with tuples s_i , for $1 \leq i \leq k$, such that for all $A \in R$,

$$s_i(A) = \begin{cases} a_i & \text{if } A = X_l, \\ b_i & \text{if } A = Y_l, \\ M + i & \text{otherwise.} \end{cases} \quad (2)$$

Note that r_{n+1} is well-defined due to the assumption $R' = \emptyset$, and that item 2 follows from the definition. For item 1, assume to the contrary that $r_{n+1} \not\models \mathcal{K}(Z)$ for some $\mathcal{K}(Z) \in \Sigma_K$. Then, by the definition of r_{n+1} , and since $r_n \models \mathcal{K}(Z)$ by the induction assumption, we obtain that $Z = X_l$ or $Z = Y_l$. Then with one application of $\mathcal{R}9$ one derives from $\mathcal{K}(Z)$ and $X_l \perp Y_l$ either $X_l \perp X_l$ or $Y_l \perp Y_l$ which both contradict the assumption $R' = \emptyset$. Hence item 1 follows.

By the above construction, taking $r := \bigcup_{n \geq 0} r_n$, we obtain that $r \models \Sigma$. Also $r \not\models \phi$ since $r_0 \not\models \phi$.

Case (ii). Consider next the case where $\phi = X \perp Y$. We add the following condition to the construction of r_n :

3. for no $t \in r_n : t(XY) = t_0(X)t_1(Y)$.

The base case. Assume first that $n = 0$. We let $r_0 := \{t_0, t_1\}$ where for all $A \in R$, $t_0(A) := 0$ and $t_1(A) := 1$. Note that r_0 satisfies items 1 and 2 trivially, and item 3 follows since neither X nor Y is empty (otherwise one derives ϕ by $\mathcal{R}1$).

The inductive step. Assume that r_n is a finite relation satisfying items 1-3. The relation r_{n+1} is now constructed as in case (i). Since items 1 and 2 hold, it suffices to show item 3. Assume that ϕ is $X \perp Y$. Assume to the contrary that for some $t \in r_{n+1} \setminus r_n : t(XY) = t_0(X)t_1(Y)$. Then by the definition of r_{n+1} , $XY \subseteq X_l Y_l$. Moreover, by the assumption and the definition of t_0 and t_1 , it follows that X and Y are two distinct attributes. Hence $X \perp Y$ is either $X_l \perp Y_l$ or $Y_l \perp X_l$. Since $X_l \perp Y_l \in \Sigma$, we then, by $\mathcal{R}2$, obtain that $\Sigma \vdash_{\mathfrak{J}} \phi$ which contradicts the assumption. Hence, item 3 also holds. This concludes the construction of the relations r_n .

Now, taking $r := \bigcup_{n \geq 0} r_n$, we again obtain that $r \models \Sigma$. Also $r \not\models \phi$ follows from item 3. This concludes the proof of Theorem 4.2. □

The existence of a complete axiomatization without arity restrictions for independence atoms remains open. In particular, our construction of a counter-model does not carry over to the general case since the interaction between independence atoms and keys is more complicated as shown by the following example. It is an interesting open problem to determine whether the general implication problem is finitely axiomatizable.

Example 4.3. It is easy to check that any relation satisfying $A \perp BC$ and $\mathcal{K}(AB)$, must also satisfy the functional dependency $B \rightarrow C$. Therefore,

$$\{A \perp BC, \mathcal{K}(AB), B \perp D\} \models BC \perp D$$

but it is also straightforward to see that $\{A \perp BC, \mathcal{K}(AB), B \perp D\} \not\models_{\mathfrak{J}} BC \perp D$. Therefore, the system \mathfrak{J} does not completely axiomatize the general implication problem of keys and arbitrary independence atoms.

5. Finite Implication

In Section 5.1 we will show that general and finite implication do not coincide for keys and unary independence atoms. Using these results, we will show in Section 5.2 that for no k there exists a k -ary axiomatization of the corresponding finite implication problem whereas 2-ary axiomatization exists for the general implication problem.

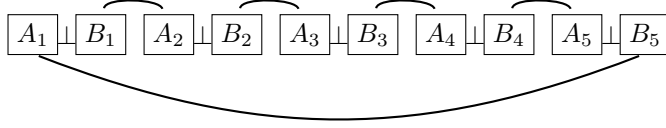


Figure 1: Σ_5

5.1. Separating Finite and General Implication

For $n \geq 2$, let $R_n := \{A_i, B_i : 1 \leq i \leq n\}$ be a relation schema, and let $\Sigma_n := \{A_i \perp B_i : 1 \leq i \leq n\} \cup \{\mathcal{K}(B_i A_{s(i)}) : 1 \leq i \leq n\}$ where s denotes the successor function that maps i to $i + 1$, for $i < n$, and n to 1. Then each Σ_n can be thought of as a smiley face of $n - 1$ eyes. For Σ_5 , this is illustrated in Figure 1 where each pair of attributes connected by an edge represents a key of Σ_5 . In this section we show in Lemmata 5.1 and 5.2 that $\Sigma_n \models_{\text{FIN}} \mathcal{K}(A_1 B_1)$, for $n \geq 2$, and $\Sigma_2 \not\models \mathcal{K}(A_1 B_1)$. In Section 5.2 we show that $\mathcal{K}(A_1 B_1)$ cannot be deduced from Σ_n in any sound $(2n - 1)$ -ary axiomatization for the finite implication problem of independence atoms and keys. Since this holds for all n , no finite axiomatization exists for the finite implication.

Lemma 5.1. *For $n \geq 2$, $\Sigma_n \models_{\text{FIN}} \mathcal{K}(A_1 B_1)$.*

Proof. Let $n \geq 2$, and let r be a finite relation over R_n such that $r \models \Sigma_n$. We show that $r \models \mathcal{K}(A_1 B_1)$. First note that since $r \models \mathcal{K}(B_n A_1)$, we obtain that

$$|r| = |r(B_n A_1)| \leq |r(B_n)| \cdot |r(A_1)|. \quad (3)$$

Let then $2 \leq i \leq n$, and assume that $|r(B_i)| = m$. Then since $r \models A_i \perp B_i$, each member of $r(A_i)$ has at least m repetitions in r , that is, $|r(A_i = b)| \geq m$ for each $b \in r(A_i)$. Therefore, and since $r \models \mathcal{K}(B_{i-1} A_i)$, we obtain that $|r(B_{i-1})| \geq m$. From this and the assumption that $|r(B_i)| = m$, it follows that $|r(B_i)| \leq |r(B_{i-1})|$. Therefore, we conclude that $|r(B_n)| \leq |r(B_1)|$ in which case $|r| \leq |r(B_1)| \cdot |r(A_1)|$ by (3). But now since $r \models A_1 \perp B_1$, we obtain that $|r(B_1)| \cdot |r(A_1)| = |r(B_1 A_1)|$ from which the claim follows. \square

The following lemma can be proved by constructing a counter example for $\Sigma_2 \models \mathcal{K}(A_1 B_1)$, similar to the one presented in the proof of Theorem 4.2.

Lemma 5.2. $\Sigma_2 \not\models \mathcal{K}(A_1 B_1)$.

Proof. We construct a countably infinite relation r over R_2 witnessing $\Sigma \not\models \mathcal{K}(A_1 B_1)$. For this we will inductively define an increasing chain (with respect to \subseteq) of finite relations r_n over R_2 such that $r_1 \not\models \mathcal{K}(A_1 B_1)$ and, for $n \geq 1$,

$$1. \ r_n \models \begin{cases} \mathcal{K}(B_2 A_1), \\ \mathcal{K}(B_1 A_2), \end{cases}$$

$$2. r_n \models \begin{cases} A_1 \perp B_1 & \text{if } n \text{ is odd,} \\ A_2 \perp B_2 & \text{if } n \text{ is even.} \end{cases}$$

Then, letting $r := \bigcup_{n \geq 1} r_n$, we obtain that $r \models \Sigma$ and $r \not\models \phi$. The construction of relations r_n is done as follows:

The base case. We first let $r_1(A_1B_1A_2B_2) := \{(0, 0, 0, 0), (0, 0, 1, 1)\}$. Then $r_1 \models \mathcal{K}(B_2A_1)$, $r_1 \models \mathcal{K}(B_1A_2)$ and $r_1 \models A_1 \perp B_1$.

The inductive step. Assume that $r_n(A_1B_1A_2B_2)$ is a finite relation satisfying the induction assumption; we will construct a finite relation r_{n+1} also satisfying the induction assumption. Assume first that $n + 1$ is even. Let m be the maximal number occurring in r_n , and let $(a_1, b_1), \dots, (a_k, b_k)$ enumerate the set $(r_n(A_2) \times r_n(B_2)) \setminus r_n(A_2B_2)$. Note that this set is non-empty because otherwise, by the induction assumption, we would obtain a finite relation r witnessing $\Sigma_2 \not\models \mathcal{K}(A_1B_1)$, contrary to Lemma 5.1. We then let

$$r_{n+1}(A_1B_1A_2B_2) := r_n \cup \{(m+i, m+i, a_i, b_i) : 1 \leq i \leq k\}.$$

By the construction and the induction assumption, it is straightforward to check that items 1 and 2 hold. In the case where $n + 1$ is odd, the relation r_{n+1} can be constructed analogously. This concludes the construction and the proof. \square

Hence, from Lemmata 5.1 and 5.2, we directly obtain the following theorem.

Theorem 5.3. *For keys and unary independence atoms taken together, the finite implication problem and the general implication problem do not coincide.*

5.2. Non-axiomatizability of Finite Implication

In this section we will show that for no k there exists a k -ary axiomatization of the finite implication problem for unary independence atoms and keys taken together. First recall from the previous section the definition of Σ_n over a relation schema R_n . We denote by $\text{Cl}(\Sigma_n)$ the set of all independence atoms and keys over R_n that are derivable from Σ_n using the rules $\mathcal{R}1, \mathcal{R}2, \mathcal{R}7$, i.e.,

$$\text{Cl}(\Sigma_n) = \Sigma_n \cup \{\mathcal{K}(D) : C \subseteq D \subseteq R_n, \mathcal{K}(C) \in \Sigma_n\} \cup \{B_i \perp A_i : 1 \leq i \leq n\} \cup \{A \perp \emptyset, \emptyset \perp A : A \subseteq R_n\}.$$

Recall that the rules $\mathcal{R}1, \mathcal{R}2, \mathcal{R}7$ state existence of trivial IAs, symmetry of IAs, and upward closure of keys. We will now show that $\text{Cl}(\Sigma_n)$ is closed under $(2n - 1)$ -ary finite implication. Hence, and since $\mathcal{K}(A_1B_1) \notin \text{Cl}(\Sigma_n)$, it follows that no sound $(2n - 1)$ -ary axiomatization of independence atoms and keys allows a deduction of $\mathcal{K}(A_1B_1)$ from Σ_n .

Note that $\mathcal{R}1, \mathcal{R}2, \mathcal{R}7$ are all 1-ary rules, i.e., rules of the form (1) for $k = 1$. Hence the above claim follows if one can show that every $(2n - 1)$ -ary finite implication of Σ_n is included in $\text{Cl}(\Sigma_n)$. Therefore it suffices to prove the following theorem which states that given any subset $\Sigma' \subseteq \Sigma_n$ of size $2n - 1$, and any consequence ϕ of Σ' , we find that $\phi \in \text{Cl}(\Sigma_n)$

Theorem 5.4. *Let $n \geq 2$, $\Sigma' := \Sigma_n \setminus \{\psi\}$ where $\psi \in \Sigma_n$, and let ϕ be an R_n -key or a unary R_n -independence atom such that $\Sigma' \models_{\text{FIN}} \phi$. Then $\phi \in \text{Cl}(\Sigma_n)$.*

The proof of Theorem 5.4 is divided into four cases, covered in Lemmata 5.6, 5.7, 5.8, and 5.9. In each case ϕ (or ψ) is fixed either as a key or a unary independence atom.

The first case where both ϕ and ψ are keys is essentially proved in Lemma 5.5 where, given a $\mathcal{K}(D) \notin \text{Cl}(\Sigma_n)$, we construct a finite relation r satisfying $\Sigma' := \Sigma_n \setminus \{\mathcal{K}(B_n A_1)\}$ and violating $\mathcal{K}(D)$. For the construction of r , we will first define tuples t, t' such that for all $X \in R_n$, $t(X) = t'(X)$ if and only if $X \in D$. Then r will be obtained by extending $\{t, t'\}$ inductively over columns as follows. Assume that r is constructed up to X_i where X_i is the i th member of $A_1, B_1, \dots, A_n, B_n$. Then we have two cases for X_{i+1} . If $t(X_i X_{i+1}) \neq t'(X_i X_{i+1})$, then we will define $r(X_i X_{i+1})$ so that $r(X_i X_{i+1}) = r(X_i) \times r(X_{i+1})$ and $r \models \mathcal{K}(X_i X_{i+1})$. If $t(X_i X_{i+1}) = t'(X_i X_{i+1})$, then by $\mathcal{K}(D) \notin \text{Cl}(\Sigma_n)$ we obtain that $X_i X_{i+1} = A_j B_j$, and therefore r must satisfy $X_i \perp X_{i+1}$. Again, $r(X_i X_{i+1})$ will be a cartesian product but this time we must include repetitions for $X_i X_{i+1}$ in r . We will start the proof with a careful investigation of the cardinalities $|r(X_i)|$ that enables the above construction.

Lemma 5.5. *Let $n \geq 2$, and let $D \subseteq R_n$ be such that $\mathcal{K}(D) \notin \text{Cl}(\Sigma_n)$. Then there exists a finite relation r and $t_0, t_1 \in r$ such that $r \models \Sigma_n \setminus \{\mathcal{K}(B_n A_1)\}$, $t_0(X) = 0$ for all $X \in R_n$, and*

$$t_1(X) = \begin{cases} 0 & \text{if } X \in D, \\ 1 & \text{if } X \in R_n \setminus D. \end{cases}$$

Proof. Let $n \geq 2$, and let $D \subseteq R_n$ be such that $\mathcal{K}(D) \notin \text{Cl}(\Sigma_n)$. We define a finite relation $r = \{t_0, \dots, t_{m-1}\}$ where $m := 2^{n+2}$, and r, t_0, t_1 satisfy the claim. We construct r inductively over columns. Let $a_i := 2^i$, $b_i := 2^{n+1-i}$, for $1 \leq i \leq n$, and let

- $t(X) = 0$ for all $X \in R_n$,
- $t'(X) = \begin{cases} 0 & \text{if } X \in D, \\ 1 & \text{if } X \in R_n \setminus D. \end{cases}$

The idea is to build inductively on i a relation $r_i = \{t_0, t_1, \dots, t_{m-1}\}$ over R_i such that $r_i \models \Sigma' \upharpoonright R_i$, $t_0 = t \upharpoonright R_i$, $t_1 = t' \upharpoonright R_i$, and $t_0(A_i B_i), \dots, t_{m-1}(A_i B_i)$ lists two copies of $\{0, \dots, a_i - 1\} \times \{0, \dots, b_i - 1\}$. Then $r := r_n$ is as wanted.

First for $i = 1$, we define r_i as a set of tuples t_0, \dots, t_{m-1} over R_i where $t_0(A_1 B_1), \dots, t_{m-1}(A_1 B_1)$ lists two copies of $\{0, \dots, a_i - 1\} \times \{0, \dots, b_i - 1\}$ and $t_0 = t \upharpoonright R_i$, and $t_1 = t' \upharpoonright R_i$. Note that the size of r_1 collapses to $m/2$. However, later each r_i will be of size m .

Assume then that $r_i = \{t_0, \dots, t_{m-1}\}$ is defined. By the induction assumption each value of $r_i(B_i)$ has a_{i+1} many repetitions in $t_0(B_i), \dots, t_{m-1}(B_i)$. Hence, we let first t'_i extend t_i with a value for A_{i+1} such that $t'_i(B_i A_{i+1}), \dots,$

$t'_{m-1}(B_i A_{i+1})$ lists $\{0, \dots, b_i - 1\} \times \{0, \dots, a_{i+1} - 1\}$ without repetitions. Then we let $r_{i+1} := \{u_0, \dots, u_{m-1}\}$ where each u_i extend t'_i with a value for B_{i+1} such that $u_0(A_{i+1}, B_{i+1}), \dots, u_{m-1}(A_{i+1}, B_{i+1})$ lists two copies of $\{0, \dots, b_i - 1\} \times \{0, \dots, a_{i+1} - 1\}$. Then we obtain that $r_{i+1} \models \Sigma' \upharpoonright R_i$. Also it easy to see that for any t and t' , r_{i+1} can be constructed so that $u_0 = t \upharpoonright R_{i+1}$ and $u_1 = t' \upharpoonright R_{i+1}$. This concludes the construction of relations r_i and the proof. \square

The proof of the first case is now easy.

Lemma 5.6. *Let $n \geq 2$, $\Sigma' := \Sigma_n \setminus \{\psi\}$ where $\psi \in \Sigma_n$ is a key, and assume that ϕ is an R_n -key such that $\Sigma' \models_{\text{FIN}} \phi$. Then $\phi \in \text{Cl}(\Sigma_n)$.*

Proof. By symmetry, we may assume that $\psi = \mathcal{K}(B_n A_1)$. Then Σ' is as in Figure 2. Let us assume to the contrary that $\phi \notin \text{Cl}(\Sigma_n)$ where $\phi = \mathcal{K}(D)$ for

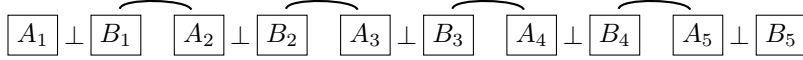


Figure 2: Σ' in case $n = 5$

some $D \subseteq R_n$. Then by Lemma 5.5 there exists a finite relation r over R_n such that $r \models \Sigma'$ and $r \not\models \phi$. Therefore, $\Sigma' \not\models_{\text{FIN}} \phi$ which shows the claim. \square

The remaining cases are stated in the following lemmata. In the next case ψ is an independence atom and ϕ is a key.

Lemma 5.7. *Let $n \geq 2$, $\Sigma' := \Sigma_n \setminus \{\psi\}$ where $\psi \in \Sigma_n$ is a unary independence atom, and assume that ϕ is an R_n -key such that $\Sigma' \models_{\text{FIN}} \phi$. Then $\phi \in \text{Cl}(\Sigma_n)$.*

Proof. By symmetry, we may assume that $\psi = A_1 \perp B_1$. Let us assume to the contrary that $\phi \notin \text{Cl}(\Sigma_n)$ where $\phi = \mathcal{K}(D)$ for some $D \subseteq R_n$. We will show that $\Sigma' \not\models_{\text{FIN}} \phi$. First we define $\Sigma^* := \Sigma_n \setminus \{\mathcal{K}(B_n A_1)\}$. Then by Lemma 5.5, there exists a finite relation $r^* = \{t_0, t_1, \dots, t_{m-1}\}$ such that $r^* \models \Sigma^*$, $t_0(X) = 0$ for all $X \in R_n$, and

$$t_1(X) = \begin{cases} 0 & \text{if } X \in D, \\ 1 & \text{if } X \in R_n \setminus D. \end{cases}$$

We let r be obtained from r^* by replacing, for $0 \leq i \leq m-1$, $t_i(A_1)$ with

- i if $i \neq 1$,
- $\begin{cases} 0 & \text{if } i = 1 \text{ and } B_n \notin D, \\ 1 & \text{if } i = 1 \text{ and } B_n \in D. \end{cases}$

The construction of r is illustrated in Figure 3 in case $B_n \notin D$. From the definition of r and the fact that $A_1 B_n \not\subseteq D$ it follows that $r \not\models \mathcal{K}(D)$ and $r \models \Sigma^* \setminus \{A_1 \perp B_1\}$. For $r \models \Sigma'$, we still need to show that $r \models \mathcal{K}(B_n A_1)$.

	A_1	B_1	\dots	\dots	\dots	\dots	A_n	B_n
t_0	0							0
t_1	0							1
t_2	2							y_2
t_3	3							y_3
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
t_{m-2}	$m-2$							y_{m-2}
t_{m-1}	$m-1$							y_{m-1}

Figure 3: r in case $B_n \notin D$

Because of the definition of $t_i(A_1)$ in r , $\mathcal{K}(B_n A_1)$ could be violated only in $\{t_0, t_1\}$. In that case we would have $t_1(A_1 B_1) = 00$ in r which contradicts with the definitions. Hence we obtain that $r \not\models \mathcal{K}(D)$ which concludes the proof. \square

In the third case ψ is a key and ϕ is an independence atom.

Lemma 5.8. *Let $n \geq 2$, $\Sigma' := \Sigma_n \setminus \{\psi\}$ where $\psi \in \Sigma_n$ is a key, and assume that ϕ is a unary R_n -independence atom such that $\Sigma' \models_{\text{FIN}} \phi$. Then $\phi \in \text{Cl}(\Sigma_n)$.*

Proof. By symmetry, we may assume that $\psi = \mathcal{K}(B_n A_1)$. Hence Σ' is as in Figure 2. Assume to the contrary that $\phi \notin \text{Cl}(\Sigma_n)$. We will show that $\Sigma' \not\models_{\text{FIN}} \phi$. Due to $\mathcal{R}2$ and by symmetry of Σ' , it suffices to consider only the cases where $\phi = A_i \perp Y$, for some $1 \leq i \leq n$ and $Y \in R_n \setminus \{B_i\}$.

So let $1 \leq i \leq n$. We will construct two finite relations r and r' such that

1. $r \models \Sigma'$ and $r' \models \Sigma'$,
2. $r \not\models \begin{cases} A_i \perp A_j & \text{for } j \leq i, \\ A_i \perp B_j & \text{for } j > i, \end{cases}$
3. $r' \not\models \begin{cases} A_i \perp A_j & \text{for } j > i, \\ A_i \perp B_j & \text{for } j < i. \end{cases}$

We let $r := \{t_0, t_1, t_2, t_3\}$ (see Figure 4) where we define, for $X \in R_n$,

- $t_0(X) = 0$,
- $t_1(X) = \begin{cases} 0 & \text{if } X = A_j \text{ for } j \leq i, \\ & \text{or } X = B_j \text{ for } j > i, \\ 1 & \text{otherwise,} \end{cases}$
- $t_2(X) = \begin{cases} 0 & \text{if } X = B_i, \\ 1 & \text{otherwise,} \end{cases}$
- $t_3(X) = \begin{cases} 0 & \text{if } X = B_j \text{ for } j < i, \\ & \text{or } X = A_j \text{ for } j > i, \\ 1 & \text{otherwise.} \end{cases}$

	A_1	B_1	\dots	\dots	A_{i-1}	B_{i-1}	A_i	B_i	A_{i+1}	B_{i+1}	\dots	\dots	A_n	B_n
t_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_1	0	1	0	1	0	1	0	1	1	0	1	0	1	0
t_2	1	1	1	1	1	1	1	0	1	1	1	1	1	1
t_3	1	0	1	0	1	0	1	1	0	1	0	1	0	1

Figure 4: r

Then we let $r' := \{t_0, t_4\}$ (see Figure 5) where we define, for $X \in R_n$,

$$\bullet t_4(X) = \begin{cases} 0 & \text{if } X = B_j \text{ for } j \geq i, \\ & \text{or } X = A_j \text{ for } j < i, \\ 1 & \text{otherwise.} \end{cases}$$

	A_1	B_1	\dots	\dots	A_{i-1}	B_{i-1}	A_i	B_i	A_{i+1}	B_{i+1}	\dots	\dots	A_n	B_n
t_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_4	0	1	0	1	0	1	1	0	1	0	1	0	1	0

Figure 5: r'

It is straightforward to check that items 1-3 hold. This concludes the proof of Lemma 5.8. \square

In the last case both ψ and ϕ are independence atoms.

Lemma 5.9. *Let $n \geq 2$, $\Sigma' := \Sigma_n \setminus \{\psi\}$ where $\psi \in \Sigma_n$ is a unary independence atom, and assume that ϕ is a unary R_n -independence atom such that $\Sigma' \models_{\text{FIN}} \phi$. Then $\phi \in \text{Cl}(\Sigma_n)$.*

Proof. By symmetry, we may assume that $\psi = A_1 \perp B_1$. Assume to the contrary that $\phi \notin \text{Cl}(\Sigma_n)$. We will show that $\Sigma' \not\models_{\text{FIN}} \phi$. Analogously to the proof of Lemma 5.8, it suffices to consider only the cases where $\phi = A_i \perp Y$, for some $1 \leq i \leq n$ and $Y \in R_n \setminus \{B_i\}$. Let $1 \leq i \leq n$. We will construct four relations r_0, r_1, r_2, r_3 such that

1. $r_i \models \Sigma'$ for $i = 0, 1, 2, 3$,
2. $r_0 \not\models A_i \perp A_j$ for $1 \leq j \leq n$,
3. $r_1 \not\models A_1 \perp B_j$ for $1 < j$,

and if $1 < i$,

4. $r_2 \not\models A_i \perp B_j$ for $j < i$,
5. $r_3 \not\models A_i \perp B_j$ for $i < j$.

For the constructions, we first define tuples t_0, \dots, t_6 as follows:

- $\bullet t_0(X) = 0$,

- $t_1(X) = \begin{cases} 0 & \text{if } X = B_j \text{ for } j > 1, \\ 1 & \text{otherwise.} \end{cases}$
- $t_2(X) = \begin{cases} 0 & \text{if } X = A_j \text{ for } j > 1, \\ 1 & \text{otherwise.} \end{cases}$
- $t_3(X) = \begin{cases} 0 & \text{if } X = A_j \text{ for } 1 < j < i, \text{ or} \\ & X = B_j \text{ for } i \leq j \leq n, \\ 1 & \text{otherwise.} \end{cases}$
- $t_4(X) = \begin{cases} 0 & \text{if } X = A_1, \\ & \text{or } X = B_j \text{ for } j \leq i, \\ 1 & \text{otherwise,} \end{cases}$
- $t_5(X) = \begin{cases} 0 & \text{if } X = A_j \text{ for } 1 < j \leq i, \\ & \text{or } X = B_j \text{ for } i < j, \\ 1 & \text{otherwise,} \end{cases}$
- $t_6(X) = \begin{cases} 0 & \text{if } X = A_j \text{ for } i < j, \\ 1 & \text{otherwise.} \end{cases}$

Then we let $r_0 := \{t_0, t_1\}$, $r_1 := \{t_0, t_2\}$, $r_2 := \{t_0, t_3\}$, and $r_3 := \{t_0, t_4, t_5, t_6\}$ which are illustrated in Figure 6-9, respectively.

	A_1	B_1	A_{i-1}	B_{i-1}	A_i	B_i	A_{i+1}	B_{i+1}	A_n	B_n
t_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_1	1	1	1	0	1	0	1	0	1	0	1	0	1	0

Figure 6: r_0

	A_1	B_1	A_{i-1}	B_{i-1}	A_i	B_i	A_{i+1}	B_{i+1}	A_n	B_n
t_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_2	1	1	0	1	0	1	0	1	0	1	0	1	0	1

Figure 7: r_1

	A_1	B_1	A_{i-1}	B_{i-1}	A_i	B_i	A_{i+1}	B_{i+1}	A_n	B_n
t_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_3	1	1	0	1	0	1	1	0	1	0	1	0	1	0

Figure 8: r_2

Recall that in the last two cases the presupposition is that $1 < i$. Also in the last case $i < n$ since $i < j \leq n$. Again, it is straightforward to check that items 1-5 hold. This concludes the proof of Lemma 5.9.

□

	A_1	B_1	\dots	\dots	A_{i-1}	B_{i-1}	A_i	B_i	A_{i+1}	B_{i+1}	\dots	\dots	A_n	B_n
t_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t_4	0	0	1	0	1	0	1	0	1	1	1	1	1	1
t_5	1	1	0	1	0	1	0	1	1	0	1	0	1	0
t_6	1	1	1	1	1	1	1	1	0	1	0	1	0	1

Figure 9: r_3

From Lemmata 5.6, 5.7, 5.8 and 5.9 we obtain Theorem 5.4. Using this we can prove the following theorem. Note that the theorem denies existence of any sort of k -ary axiomatization, finite or infinite.

Theorem 5.10. *For no natural number k , there exists a sound and complete k -ary axiomatization of the finite implication problem for unary independence atoms and keys taken together.*

Proof. Let \mathfrak{A} be a sound k -ary axiomatization for unary independence atoms and keys over some natural number k , and let n be such that $n \geq 2$ and $2n > k$. First we observe that the closure of Σ_n under k -ary finite implication is $\text{Cl}(\Sigma_n)$. For this, first note that every instance of $\text{Cl}(\Sigma_n)$ can be derived from Σ_n using a sound 1-ary rule $\mathcal{R}1$, $\mathcal{R}2$, or $\mathcal{R}7$. On the other hand, assume that $\Sigma' \models \phi$ where $\Sigma' \subseteq \text{Cl}(\Sigma_n)$ is a subset of size k and ϕ is a key or a unary independence atom. Since each instance of Σ' is derivable from at most one instance of Σ_n with single application of $\mathcal{R}1$, $\mathcal{R}2$, or $\mathcal{R}7$, we find a subset $\Sigma'' \subseteq \Sigma_n$ of size at most k such that $\Sigma'' \models \Sigma'$, and hence $\Sigma'' \models \phi$. Since Σ'' is a proper subset of Σ_n , we obtain by Theorem 5.4 that $\phi \in \text{Cl}(\Sigma_n)$. Hence, the closure of Σ_n under k -ary finite implication is exactly $\text{Cl}(\Sigma_n)$.

Since \mathfrak{A} is k -ary and $\mathcal{K}(A_1 B_1) \notin \text{Cl}(\Sigma_n)$, it follows that $\Sigma_n \not\models_{\mathfrak{A}} \mathcal{K}(A_1 B_1)$. On the other hand, $\Sigma_n \models_{\text{FIN}} \mathcal{K}(A_1 B_1)$ by Theorem 5.1. Hence \mathfrak{A} is not complete which shows the claim. \square

Let us then turn to the case where independence atoms have no arity restrictions. Fix $X \perp Y$ as a R_n -independence atom of arity greater than 1. Using Decomposition and Symmetry (rules $\mathcal{R}4$ and $\mathcal{R}2$), one then finds a unary $A \perp B \notin \text{Cl}(\Sigma_n)$ such that $\{X \perp Y\} \models A \perp B$. Now by Theorem 5.4, given any proper subset $\Sigma' \subset \Sigma_n$, it follows that $\Sigma' \not\models A \perp B$, thus implying that $\Sigma' \not\models X \perp Y$. Hence, a R_n -independence atom ϕ is a consequence of Σ' only if it is unary. It is easy to see that this observation together with the proof of Theorem 5.10 shows the following generalization.

Theorem 5.11. *For no natural number k , there is a sound and complete k -ary axiomatization of the finite implication problem for independence atoms and keys taken together.*

6. Unary Keys and General Independence Atoms

Considering our results for the combined class of general keys and unary independence atoms, we show now that the combined class of unary keys and

general independence atoms enjoys more desirable properties. Indeed, the associated finite and general implication problems coincide for this class, and are finitely 2-ary axiomatizable by the axiomatization \mathfrak{J} (see Table 1 in page 10) for the combined class of general keys and unary independence atoms. For proving Theorem 6.2, it suffices to construct a finite model of $\Sigma \cup \{\neg\phi\}$, given $\Sigma \not\vdash_{\mathfrak{J}} \phi$. If ϕ is a key, then the model consists of two rows which differ from another only on derivably key attributes. If ϕ is an independence atom, then using Theorem 6.1 we first obtain a finite model of the restriction of $\{X \perp Y : X \perp Y \in \Sigma\} \cup \{\neg\phi\}$ to the attributes that are neither derivably keys nor derivably constants. A straightforward extension to the remaining attributes results then in a model of $\Sigma \cup \{\neg\phi\}$.

Theorem 6.1 ([42]). *Rules $\mathcal{R}1 - \mathcal{R}5$ of the axiomatization \mathfrak{J} are sound and complete for the finite implication problem of independence atoms.*

Theorem 6.2. *Assume that R is a relation schema and $\Sigma \cup \{\phi\}$ consists of R -independence atoms and unary R -keys. Then the following are equivalent:*

1. $\Sigma \vdash_{\mathfrak{J}} \phi$,
2. $\Sigma \models \phi$,
3. $\Sigma \models_{\text{FIN}} \phi$.

Proof. It is straightforward to see that (1) \Rightarrow (2) and (2) \Rightarrow (3). We show that (3) \Rightarrow (1). Assume to the contrary that $\Sigma \not\vdash_{\mathfrak{J}} \phi$, we show by constructing a counterexample that $\Sigma \not\models_{\text{FIN}} \phi$. Let $U := \{A \in R : \Sigma \vdash_{\mathfrak{J}} \mathcal{K}(A)\}$ and $V := \{B \in R : \Sigma \vdash_{\mathfrak{J}} B \perp B\}$. We consider the cases where ϕ is either a unary key or an independence atom in the following.

Assume first that $\phi = \mathcal{K}(C)$ for some $C \in R$. Then we let $r := \{t_0, t_1\}$ where

- $t_0(X) = 0$ for $X \in R$,
- $t_1(X) = \begin{cases} 0 & \text{if } X \in R \setminus U, \\ 1 & \text{if } X \in U. \end{cases}$

Since $C \notin U$, we obtain that $r \not\models \mathcal{K}(C)$. Also for any $\mathcal{K}(A) \in \Sigma$, $r \models \mathcal{K}(A)$ since $A \in U$. Let then $X \perp Y \in \Sigma$, we show that $r \models X \perp Y$. If $X \cap U \neq \emptyset$, then using rules $\mathcal{R}7$ and $\mathcal{R}9$, we obtain that $\Sigma \vdash_{\mathfrak{J}} Y \perp Y$, and therefore, by rules $\mathcal{R}2$ and $\mathcal{R}4$, $Y \subseteq V$. Since U and V are disjoint (otherwise, using rules $\mathcal{R}7$ and $\mathcal{R}8$, one could deduce $\mathcal{K}(C)$), Y is constant in r . Hence we conclude that $r \models X \perp Y$. Since the case where $Y \cap U \neq \emptyset$ is analogous, and the case where $XY \cap U = \emptyset$ is trivial, this concludes the case where $\phi = \mathcal{K}(C)$.

Assume then that $\phi = C \perp D$ for some $C, D \subseteq R$. First note that $U \cap CD = \emptyset$ because otherwise one could deduce $C \perp D$ using rules $\mathcal{R}1$, $\mathcal{R}2$, $\mathcal{R}3$, $\mathcal{R}7$ and $\mathcal{R}9$. Let $\Sigma' := \{A \setminus UV \perp B \setminus UV : A \perp B \in \Sigma\}$. Then $\Sigma' \not\vdash_{\mathfrak{J}} C \setminus V \perp D \setminus V$ because otherwise one could deduce $C \perp D$ with $\mathcal{R}1$, $\mathcal{R}2$ and $\mathcal{R}3$. By Theorem 6.1 there exists a finite relation $r' := \{t_1, \dots, t_n\}$ over $R \setminus UV$ such that $r' \models \Sigma'$ and $r' \not\models C \setminus V \perp D \setminus V$. We then let r be obtained from r' by extending each t_i to UV as follows: $t_i(A) = i$ for $A \in U$, and $t_i(B) = 0$ for $B \in V$. Note that r is

well defined since U and V are again disjoint (otherwise, using rules $\mathcal{R}1$, $\mathcal{R}3$ and $\mathcal{R}9$, one could deduce $C \perp D$). Since $r \not\models C \perp D$, it suffices to show that $r \models \Sigma$. If $\mathcal{K}(A) \in \Sigma$, then $r \models \mathcal{K}(A)$ by $A \in U$. Let $X \perp Y \in \Sigma$. The cases where U intersects with X or Y are analogous to the previous case. Hence assume that $XY \cap U = \emptyset$. Then $X \setminus V \perp Y \setminus V \in \Sigma'$, and therefore $r \models X \setminus V \perp Y \setminus V$ by the construction. Since also $r \models V \perp V$, we conclude that $r \models X \perp Y$. This concludes the case where $\phi = C \perp D$ and the proof. \square

7. Armstrong relations

In this section we consider Armstrong relations for different classes of IAs and keys. Given a relation schema R , and a set Σ of constraints in a class \mathcal{C} , an Armstrong relation for the (finite) implication problem of \mathcal{C} is a relation that satisfies all constraints in \mathcal{C} and violates all constraints in \mathcal{C} not (finitely) implied by Σ . We say that the (finite) implication problem for a class \mathcal{C} of constraints enjoys Armstrong relations if for all relation schemata R and all sets Σ of constraints in \mathcal{C} , Σ has an Armstrong relation for the (finite) implication problem of \mathcal{C} .

The concept of Armstrong relations is motivated by theory and practice. In terms of theory, Armstrong relations embody a stronger notion of completeness in the sense that a single relation must violate all non-implied constraints. This means that an Armstrong relation can be regarded as an exact representation of the given set of constraints. In fact, given a set Σ of constraints from class \mathcal{C} , and an Armstrong relation db for Σ in \mathcal{C} , the problem of deciding whether any given constraint φ in \mathcal{C} is implied by Σ reduces to the problem of deciding whether φ is satisfied by db . These properties of Armstrong relations have also been shown useful in practice, where they can be used to identify constraints that are semantically meaningful for a given application domain. This task is paramount for the design of any database schema and therefore the processing and application of data as a whole. The task is also challenging as database designers cannot be expected to know much about the application domain. It is therefore helpful for the designers to consult with domain experts. However, domain experts cannot be expected to know anything about database constraints, which raises the question of how the designers should communicate effectively with the domain experts. This communication mismatch between designers and domain experts may be addressed by Armstrong relations. For example, instead of explaining their current understanding of the application domain by referring to the set Σ of constraints they perceive as meaningful, the designers may use an algorithm to compute an Armstrong relation for Σ and inspect the relation together with the domain experts. The domain experts can then *notice* flaws with the perceptions, and point these flaws out to the designers. This improves the designer's understanding of the application domain. Indeed, for keys and functional dependencies, empirical evidence suggests that the process of using Armstrong relations in the requirements acquisition phase does indeed lead to the discovery of additional constraints that are meaningful for the application domain [43, 45].

In the following we present one result for the general implication problem for the class of keys and unary independence atoms, and one result for the implication problem for the class of independence atoms and unary keys.

Our first main result shows how to construct for every given set of keys and unary independence atoms an infinite Armstrong relation. As the finite implication problem for this class is different from the general implication problem, see Theorem 5.3, it is impossible in general to construct finite Armstrong relations.

Theorem 7.1. *The implication problem for keys and unary independence atoms enjoys infinite Armstrong relations.*

Proof. Let R be a relation schema and Σ a set of keys and unary IAs over R . We construct an infinite relation r satisfying exactly those keys and unary IAs that are implied by Σ . Let $R' := \{A \in R : \Sigma \models A \perp A\}$ and Σ' the restriction of Σ to $R \setminus R'$. Now if r' is an Armstrong relation for Σ' and r an extension to R obtained from r' by extending its tuples with constant values for R' , then using the sound rules in Table 1 we notice that $\Sigma \models \phi \Leftrightarrow \Sigma' \models \phi'$ and $r \models \phi \Leftrightarrow r' \models \phi'$, where ϕ' is the restriction of ϕ to $R \setminus R'$. Hence, and since $\Sigma' \not\models A \perp A$ for $A \in R \setminus R'$, we may without loss of generality assume that $R' = \emptyset$.

Let then $\mathcal{K}(X_1), \dots, \mathcal{K}(X_n)$ list all R -keys that are not implied by Σ . We define $r' := \bigcup_{i=1}^n r'_i$ where each r'_i consists of two tuples t_i, t'_i defined as follows for $A \in R$:

$$\begin{aligned} \bullet t_i(A) &= \begin{cases} 0 & \text{if } A \in X_i, \\ i & \text{otherwise,} \end{cases} \\ \bullet t'_i(A) &= \begin{cases} 0 & \text{if } A \in X_i, \\ n+i & \text{otherwise.} \end{cases} \end{aligned}$$

Then $r' \not\models \mathcal{K}(X_i)$ for all $i = 1, \dots, n$. Let $Y_1 \perp Z_1, \dots, Y_m \perp Z_m$ list all R -independence atoms implied by Σ . We let $r := \bigcup_{i=0}^{\infty} r_i$ where $r_0 := r' \cup \{c, c'\}$ for constant tuples c and c' mapping all attributes of R to -1 and -2 , respectively. Note that then $r \not\models A \perp A$ for all $A \in R$. The remaining r_i are constructed inductively. For the claim it then suffices to confirm that for all i ,

- (i) $r_i \models \phi$ if ϕ is a key implied by Σ ,
- (ii) $r_i \models Y_k \perp Z_k$ if $i \geq 1$ and $k = i \pmod{n}$,
- (iii) there exists no unary IA $X \perp Y$ not implied by Σ and $t \in r_i$ such that $t(X) = -1$ and $t(Y) = -2$.

The base case. First consider the case where $i = 0$. Items (ii) and (iii) are evident so we show only item (i). Assume that $\Sigma \models \mathcal{K}(X)$. Then $X \not\subseteq X_i$ for all $i = 1, \dots, n$ and hence by the construction each $t \in r_0$ maps some $A \in X$ to a unique number $-2, -1, 1, \dots, 2n$, i.e., such that it differs from all $t'(A)$ where $t' \neq t$.

The inductive case. If r_i is defined and $r_i \models Y_k \perp Z_k$ for $k = i + 1 \pmod{n}$, then we let $r_{i+1} := r_i$. Otherwise, let $(a_1, b_1), \dots, (a_l, b_l)$ enumerate $r(Y_k) \times r(Z_k) \setminus r(Y_k Z_k)$, and let M be the greatest number appearing in r_i . We then let r_{i+1} be obtained by extending r_i with tuples t_j , for $1 \leq j \leq l$, such that for all $A \in R$,

$$t_j(A) = \begin{cases} a_j & \text{if } A = Y_k, \\ b_j & \text{if } A = Z_k, \\ M + j & \text{otherwise.} \end{cases} \quad (4)$$

Checking items (i)-(iii) proceeds then as in Case (ii) in the proof of Theorem 4.2. This concludes the proof. \square

Theorem 7.1 establishes the general existence of infinite Armstrong relations for every set of keys and unary independence atoms. As the requirement of having infinite relations is necessary for the general implication problem, we cannot simply use these Armstrong relations for acquiring semantically meaningful keys and unary independence atoms. In the future, one may think about effective ways of presenting carefully chosen snippets of these relations to help with the acquisition. Since finite implication is the practically relevant case, the interesting open question remains whether the finite implication problem for keys and unary independence atoms enjoys finite or infinite Armstrong relations.

Our second main result shows how to construct finite Armstrong relations for every given set of independence atoms and unary keys. Note that finite and general implication problems coincide in this case.

Theorem 7.2. *The implication problem and finite implication problem for independence atoms and unary keys enjoy finite Armstrong relations.*

Proof. Let R be a relation schema and Σ a set of independence atoms and unary keys over R . As in the proof of Theorem 7.1, we may assume that $V := \{A \in R : \Sigma \models A \perp A\} = \emptyset$. Given $U := \{A \in R : \Sigma \models \mathcal{K}(A)\}$ and let $\Sigma' := \{X \setminus U \perp Y \setminus U : X \perp Y \in \Sigma\}$. Let $r' = \{t_1, \dots, t_n\}$ be a finite Armstrong relation (with respect to IAs only) for Σ' over $R \setminus U$, obtained by Theorem 4 in [42]. We define $r := \{t_{i,j} : i = 1, \dots, n \text{ and } j = 0, 1\}$ where each $t_{i,j}$ agrees with t_i on attributes of $R \setminus U$ and maps all attributes of U to $jn + i$. By the construction, $r \models \mathcal{K}(A) \Leftrightarrow \Sigma \models \mathcal{K}(A)$.

Let $X \perp Y$ be a non-trivial R -independence atom, i.e., atom of the form $X \perp Y$ where X and Y are both non-empty (otherwise clearly $r \models X \perp Y$ and $\Sigma \models X \perp Y$). First note that $XY \cap U = \emptyset$ because otherwise using $\mathcal{R}2, \mathcal{R}4, \mathcal{R}7, \mathcal{R}9$ one contradicts the assumption $V = \emptyset$. Since r' is an Armstrong relation for Σ' , it holds by the construction that $\Sigma' \models X \perp Y \Leftrightarrow r \models X \perp Y$. It remains to show that $\Sigma \models X \perp Y \Leftrightarrow \Sigma' \models X \perp Y$, and since $\Sigma \models \Sigma'$ it suffices to consider only the direction “ \Rightarrow ”. Assume that $\Sigma \models X \perp Y$ and let r_0 be a relation over $R \setminus U$ with $r_0 \models \Sigma'$. Let r_1 be obtained from r_0 by extending its tuples with key values for the attributes of U . Then r_1 satisfies every unary key and

trivial IA that is in Σ . Moreover, if $A \perp B \in \Sigma$ is non-trivial, then $AB \cap U = \emptyset$ by $\mathcal{R}2, \mathcal{R}7, \mathcal{R}9$ and since $V = \emptyset$. Hence $A \perp B \in \Sigma'$ and $r_1 \models A \perp B$ by the construction. Therefore by the assumption $r_1 \models X \perp Y$, and hence $r_0 \models X \perp Y$. This concludes the proof of the claim and the theorem. \square

Theorem 7.2 meets the theoretical and practical motivation for Armstrong relations in the case of independence atoms and unary keys. Further research may investigate how to improve our construction in terms of the number of tuples required in an Armstrong relation. Based on algorithmic implementations of these constructions, it is worth investigating how many more meaningful independence atoms and unary keys can be identified by the inspection of Armstrong relations, in comparison to not having available such relations.

8. On the complexity of the implication problems

In this section we study the complexity of the implication problems axiomatized in Theorems 4.2 and 6.2. It is known that the implication problems of keys and independence atoms in isolation can be decided in polynomial time [42, 65]. Below $|(R, \Sigma \cup \{\phi\})|$ denotes the length of a string encoding R and the formulas in $\Sigma \cup \{\phi\}$ under a fixed string encoding of these objects.

The following lemma shows that an IA is derivable exactly when its constant part and non-constant part are derivable separately.

Lemma 8.1. *Assume that R is a relation schema and that Σ is a finite set of R -keys and R -independence atoms. Denote by $Z \subseteq R$ the maximal set of attributes such that $\Sigma \vdash_{\exists} Z \perp Z$. Then $\Sigma \vdash_{\exists} X \perp Y$ if and only if the following conditions hold:*

1. $\Sigma^* \vdash_{\exists} (X \setminus Z) \perp (Y \setminus Z)$, where $\Sigma^* := \{(U \setminus Z) \perp (V \setminus Z) : U \perp V \in \Sigma\}$
2. $X \cap Y \subseteq Z$ (i.e., $\Sigma \vdash_{\exists} (X \cap Y) \perp (X \cap Y)$).

Furthermore, $\Sigma \vdash_{\exists} \mathcal{K}(X)$ if and only if $Y \setminus Z \subseteq X$ for some $\mathcal{K}(Y) \in \Sigma$.

Proof. We consider the first claim only. The conditions 1 and 2 together are clearly sufficient. We show using induction on the length of derivation that $\Sigma \vdash_{\exists} X \perp Y$ implies conditions 1 and 2. We consider the rule $\mathcal{R}5$ only. Assume that an atom $X \perp YU$ has been derived using rule $\mathcal{R}5$ applied to atoms $X \perp Y$ and $XY \perp U$. By the induction hypothesis, it holds that

- i) $\Sigma^* \vdash_{\exists} (X \setminus Z) \perp (Y \setminus Z)$ and $\Sigma^* \vdash_{\exists} (XY \setminus Z) \perp (U \setminus Z)$,
- ii) $\Sigma^* \vdash_{\exists} (X \cap Y) \perp (X \cap Y)$ and $\Sigma^* \vdash_{\exists} (XY \cap U) \perp (XY \cap U)$.

Note first that since

$$X \cap YU = (X \cap Y) \cup (X \cap U)$$

$\Sigma^* \vdash_{\exists} (X \cap YU) \perp (X \cap YU)$ can be easily showed using ii). Analogously, by noting that $XY \setminus Z = (X \setminus Z)(Y \setminus Z)$ one application of the rule $\mathcal{R}5$ to the atoms in i) allows us to show that $\Sigma^* \vdash_{\exists} (X \setminus Z) \perp (YU \setminus Z)$. \square

Theorem 8.2. *The general implication problems of keys and unary independence atoms, and of unary keys and general independence atoms can be decided in polynomial time.*

Proof. We define an algorithm which for a given finite input consisting of a relation schema R and a finite set $\Sigma \cup \{\phi\}$ of R -keys and R -independence atoms decides whether $\Sigma \models \phi$ holds in time polynomial in $|(R, \Sigma \cup \{\phi\})|$. By Theorems 4.2 and 6.2 it suffices to define a polynomial time algorithm for deciding whether $\Sigma \vdash_{\mathcal{J}} \phi$. Let us first consider the case where ϕ is of the form $X \perp Y$. It suffices to check whether conditions 1 and 2 of Lemma 8.1 hold. Note that both conditions 1 and 2 can be checked in polynomial time assuming the set Z can be computed in polynomial time. For condition 1 this follows by Theorem 9 of [22]. Therefore, in the remainder of the proof it suffices to show that the set Z can indeed be computed in polynomial time.

Denote by $Z_0 \subseteq Z$ the maximal set of attributes such that $\Sigma^- \vdash_{\mathcal{J}} Z_0 \perp Z_0$, where $\Sigma^- := \{U \perp V : U \perp V \in \Sigma\}$. Note that the set Z_0 can be computed easily since it holds that

$$Z_0 = \cup\{X \mid XY \perp XZ \in \Sigma\}.$$

A polynomial time algorithm for computing the set Z

Repeat the following two steps until no new attributes are added to Z .
First initialize Z to Z_0 .

- Step 1 Extend Σ by $\mathcal{K}(X \setminus Z)$ for every $\mathcal{K}(X) \in \Sigma$,
Step 2 Extend Σ by $W \perp W$ and add the attributes of W to Z , for each subset $\{V \perp W, \mathcal{K}(V')\} \subseteq \Sigma$ such that $V' \subseteq V$.

Note that the extensions of Σ in the above algorithm correspond to derivations by the rules $\mathcal{R}8$ and $\mathcal{R}9$. In the derivations corresponding to Step 2 the upward closure of keys (rule $\mathcal{R}7$) is also needed. On the other hand, using Lemma 8.1 it is easy to see that the algorithm described above generates the maximal set Z such that $\Sigma \vdash_{\mathcal{J}} Z \perp Z$.

It is easy to see that the algorithm halts in at most $|R \setminus Z_0|$ rounds, and that in each round only polynomially many new atoms are added to Σ . The running time of the algorithm is clearly polynomial in $|(R, \Sigma \cup \{\phi\})|$.

Let us then consider the case where ϕ is of the form $\mathcal{K}(X)$. By Lemma 8.1, $\Sigma \vdash_{\mathcal{J}} \mathcal{K}(X)$ if and only if $Y \setminus Z \subseteq X$ for some $\mathcal{K}(Y) \in \Sigma$. Since Z can be constructed in polynomial time, it follows that $\Sigma \vdash_{\mathcal{J}} \mathcal{K}(X)$ can be checked in polynomial time. □

The next example illustrates the execution of the above algorithm.

Example 8.3. Let $R = \{A_1, \dots, A_5\}$ and

$$\Sigma := \{\mathcal{K}(A_1), \mathcal{K}(A_2A_3), \mathcal{K}(A_4A_5), A_1 \perp A_2, A_3 \perp A_4\}.$$

It is now straightforward to verify that three iterations of the algorithm are needed to show that $\Sigma \vdash_{\mathcal{J}} \mathcal{K}(A_5)$. The first iteration of the algorithm generates the atom $A_2 \perp A_2$. Then, in the first and the second step of the second iteration, key $\mathcal{K}(A_3)$ and independence atom $A_4 \perp A_4$ are generated, respectively. Finally, the target key $\mathcal{K}(A_5)$ is generated during the first step of the third iteration.

It is worth noting that the algorithm of Theorem 8.2 does not completely solve the implication problem of keys and non-unary independence atoms. In Example 4.3 it was noted that

$$\{A \perp BC, \mathcal{K}(AB), B \perp D\} \models BC \perp D,$$

but the algorithm fails to identify the atom $BC \perp D$ as a logical consequence of $\{A \perp BC, \mathcal{K}(AB), B \perp D\}$.

9. Conditions for no interaction between keys and independence atoms

In this section we formulate conditions guaranteeing no interaction between keys and independence atoms. The next definition formalizes this notion.

Definition 9.1. Let R be a relation schema and Σ_K and Σ_I be sets of R -keys and R -independence atoms, respectively. We say that Σ_I and Σ_K have *no interaction* if the following holds:

1. for any R -key ϕ , if $\Sigma_I \cup \Sigma_K \models \phi$, then $\Sigma_K \models \phi$.
2. for any R -independence atom ϕ , if $\Sigma_I \cup \Sigma_K \models \phi$, then $\Sigma_I \models \phi$.

In general, the decidability of the problem of determining whether given finite sets Σ_I and Σ_K have no interaction is open. However, as an application of Theorem 4.2 the following result can be obtained. The analogous result obviously holds also for unary keys and arbitrary independence atoms.

Theorem 9.2. *Let R be a relation schema and let Σ_K and Σ_I be sets of R -keys and unary R -independence atoms such that the following conditions hold:*

1. *for all $\mathcal{K}(X) \in \Sigma_K$, and $A \in R \setminus X$, the atom $X \perp A$ cannot be derived from Σ_I using rules $\mathcal{R}1 - \mathcal{R}5$,*
2. *for all $\mathcal{K}(X) \in \Sigma_K$, $Z_0 \cap X = \emptyset$, where Z_0 consists of the attributes Y such that $Y \perp Y$ can be derived from Σ_I using rule $\mathcal{R}4$.*

Then Σ_I and Σ_K have no interaction. Furthermore, the question whether Σ_I and Σ_K satisfy conditions 1 and 2 can be decided in polynomial time.

Proof. Conditions 1 and 2 imply that Rules $\mathcal{R}8$ and $\mathcal{R}9$ cannot be instantiated using the atoms of $\Sigma_I \cup \Sigma_K$. By Theorem 4.2, the claim follows. \square

The next example illustrate the use of Theorem 9.2.

Example 9.3. Let $R = \{A_1, \dots, A_5\}$ and

$$\Sigma := \{\mathcal{K}(A_2A_3), \mathcal{K}(A_4A_5), A_1 \perp A_2, A_3 \perp A_4\}.$$

It is now easy to check that Σ satisfies conditions 1 and 2 of Theorem 9.2. Define $\Sigma^* := \Sigma \cup \{\mathcal{K}(A_1)\}$. Note that Σ^* does not satisfy condition 1 of Theorem 9.2. Furthermore, by Example 8.3, the atoms of Σ^* have non-trivial interaction since $\Sigma^* \models \mathcal{K}(A_5)$ but clearly $\{\mathcal{K}(A_1), \mathcal{K}(A_2A_3), \mathcal{K}(A_4A_5)\} \not\models \mathcal{K}(A_5)$.

Next we turn to keys and arbitrary independence atoms. Recall that in this case the decidability of the general implication problem is open.

Theorem 9.4. *Let R be a relation schema and let Σ_K and Σ_I sets of R -keys and R -independence atoms such that the following conditions hold:*

1. *for all keys $\mathcal{K}(X) \in \Sigma_K$ and independence atoms $Y \perp Z \in \Sigma_I$: $X \cap (Y \cup Z) = \emptyset$.*

Then Σ_I and Σ_K have no interaction.

Proof. Let us first consider the case $\phi := X \perp Y$. Assume that $\Sigma_I \not\models \phi$. We need to show

$$\Sigma_I \cup \Sigma_K \not\models \phi. \quad (5)$$

Let r be a finite relation over the attributes $R_I \subseteq R$ of Σ_I such that $r \models \Sigma_I$ but $r \not\models \phi$. Since all keys $\mathcal{K}(U) \in \Sigma_K$ satisfy $U \subseteq R \setminus R_0$, it follows that r can be trivially extended to a relation r' over R satisfying all keys over $R \setminus R_0$. Now r' witnesses (5).

The case $\phi := \mathcal{K}(X)$ is proved analogously. \square

We end this section with two examples illustrating the use of Theorem 9.4.

Example 9.5. Let $R = \{A, B, C, D\}$ and define $\Sigma_I := \{A \perp B, AB \perp D\}$ and $\Sigma_K := \{\mathcal{K}(C)\}$. Now the sets Σ_I and Σ_K have no interaction by Theorem 9.4.

Example 9.6. Let $R = \{A, B, C, D\}$ and define $\Sigma_I := \{A \perp BC, B \perp D\}$ and $\Sigma_K := \{\mathcal{K}(AB)\}$. Now Σ_I and Σ_K do not satisfy the condition of Theorem 9.4. Furthermore, as pointed out in Example 4.3, $\Sigma_I \cup \Sigma_K \models BC \perp D$ but the relation $r := \{t_0, t_1\}$, where

- $t_0(X) = 0$ for $X \in R$
- $t_1(A) = t_1(B) = 0$, and $t_1(C) = t_1(D) = 1$

witnesses that $\Sigma_I \not\models BC \perp D$. Therefore, the sets Σ_I and Σ_K indeed have non-trivial interaction.

We conclude this section by noting that Theorems 9.2 and 9.4 cover many cases of constraint sets that occur in practice. Each theorem establishes conditions of no interaction that are more liberal than simply saying that a given constraint set does not imply any constancy atom. The latter condition, however, seems to be very realistic as any constraint set that occurs in practice is unlikely to imply any constancy atom. If it did, then why store a column in which at most one value can occur.

10. Conclusion and Future Work

We have initiated research on the interaction between independence atoms and keys. We showed that the finite and general implication problems for the combined class of independence atoms and keys differ from one another. For the combined class of keys and unary independence atoms we established a finite axiomatization \mathcal{J} for the general implication problem, and showed that the finite implication problem for this class has no finite axiomatization. The non-axiomatizability result holds also in the case where the arity of independence atoms is not restricted to one. For the combined class of independence atoms and unary keys we showed that the finite and general implication problems coincide, and that \mathcal{J} also forms a finite axiomatization for this class. As an application of our axiomatization we further showed that i) the general implication problem for the combined class of keys and unary independence atoms, and ii) the finite and general implication problems for the combined class of independence atoms and unary keys can all be decided in polynomial time. We also showed how to construct i) infinite Armstrong relations for the general implication of keys and unary independence atoms, and ii) finite Armstrong relations for the finite and general implication of independence atoms and unary keys. As a final application we established two conditions which guarantee that there is no interaction between keys and independence atoms. Either of the conditions can be verified in polynomial time, which means that instances of the combined implication problem that satisfy one of our conditions can be solved efficiently by using tools to decide implication for either keys or independence atoms alone.

Despite our negative results there is still hope for a general practical solution as the general implication problem for arbitrary independence atoms and keys might enjoy a finite axiomatization, or the finite implication problem may be decidable. In fact, there are showcases in the literature for positive and negative results. On the one hand, join dependencies are not axiomatizable by a finite set of Horn rules [59], but efficiently decidable by the Chase [53]. On the other hand, functional and inclusion dependencies are not axiomatizable by a finite set of Horn rules and also undecidable [9, 54]. Also, the lack of a finite axiomatization may not apply if one permits other intermediate results in a derivation. This approach of finding axioms by allowing the use of an extended language is traditionally taken after the non-axiomatizability of the non-extended language has been established [54]. The same applies to undecidability results. As an alternative it is also interesting to consider inference systems that are complete, but not necessarily sound. An important recent and very related example is the work by Niepert, Gyssens, and Van Gucht [56]. In that paper, the authors present a system that is complete for the implication problem of probabilistic conditional independence statements. One may use this result in the contrapositive: if a given statement φ cannot be inferred from the given set Σ of statements in the complete system, then φ is not implied by Σ , either. This type of result can be useful in practice. Yet another alternative pathway for future work is to explore different sub-classes for the combined class of independence atoms and keys, and possibly other constraints. As pointed out before,

an important class are inclusion dependencies [8, 39]. While the implication problem is already undecidable for keys and foreign keys [19] it is interesting to consider fragments of keys and independence atoms together with unary inclusion dependencies [10]. Our research should thus be seen as a driver for future investigations on the interaction of keys and independence atoms, similar to what has been done for other classes of dependencies, such as functional and inclusion dependencies. Modern real-world requirements often demand more flexible data formats, for which more sophisticated notions of keys and independence atoms must be developed. As far as we know there are various proposals for keys in SQL [40], possibilistic [27, 38] or probabilistic data models [5, 61], RDF [44] and XML [31, 33], for example, but independence atoms have not been studied yet in advanced data formats.

Acknowledgements

The authors would like to thank Henning Köhler for a number of useful corrections and suggestions, especially for pointing out simpler proofs for Theorem 4.2 and Lemma 5.5. We would like to thank the anonymous referees for helping us improve the final presentation of our results.

- [1] M. Arenas. Normalization theory for XML. *SIGMOD Record*, 35(4):57–64, 2006.
- [2] M. Arenas, C. Gutierrez, D. P. Miranker, J. Pérez, and J. Sequeda. Querying semantic data on the web? *SIGMOD Record*, 41(4):6–17, 2012.
- [3] C. Beeri, R. Fagin, and J. H. Howard. A complete axiomatization for functional and multivalued dependencies in database relations. In D. C. P. Smith, editor, *Proceedings of the 1977 ACM SIGMOD International Conference on Management of Data, Toronto, Canada, August 3-5, 1977.*, pages 47–61, 1977.
- [4] R. R. Bouckaert, R. Hemmecke, S. Lindner, and M. Studený. Efficient algorithms for conditional independence inference. *Journal of Machine Learning Research*, 11:3453–3479, 2010.
- [5] P. Brown and S. Link. Probabilistic keys for data quality management. In J. Zdravkovic, editor, *Advanced Information Systems Engineering - 27th European Conference, CAiSE 2015, Stockholm, Sweden, June 8-12, 2015. Proceedings*, volume 9097 of *Lecture Notes in Computer Science*, pages 118–132. Springer, 2015.
- [6] A. Cali, D. Calvanese, G. De Giacomo, and M. Lenzerini. Data integration under integrity constraints. *Inf. Syst.*, 29(2):147–163, 2004.
- [7] A. Cali, D. Lembo, and R. Rosati. On the decidability and complexity of query answering over inconsistent and incomplete databases. In *PODS*, pages 260–271. ACM, 2003.

- [8] M. A. Casanova, R. Fagin, and C. H. Papadimitriou. Inclusion dependencies and their interaction with functional dependencies. *J. Comput. Syst. Sci.*, 28(1):29–59, 1984.
- [9] A. K. Chandra and M. Y. Vardi. The implication problem for functional and inclusion dependencies is undecidable. *SIAM Journal on Computing*, 14(3):671–677, 1985.
- [10] S. S. Cosmadakis, P. C. Kanellakis, and M. Y. Vardi. Polynomial-time implication problems for unary inclusion dependencies. *J. ACM*, 37(1):15–46, 1990.
- [11] A. P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(1):1–31, 1979.
- [12] C. Delobel. Normalization and hierarchical dependencies in the relational data model. *ACM Trans. Database Syst.*, 3(3):201–222, 1978.
- [13] J. Demetrovics. On the number of candidate keys. *Inf. Process. Lett.*, 7(6):266–269, 1978.
- [14] J. Demetrovics, G. O. H. Katona, D. Miklós, O. Seleznev, and B. Thalheim. Asymptotic properties of keys and functional dependencies in random databases. *Theor. Comput. Sci.*, 190(2):151–166, 1998.
- [15] G. V. den Broeck and M. Niepert. Lifted probabilistic inference for asymmetric graphical models. In B. Bonet and S. Koenig, editors, *AAAI*, pages 3599–3605. AAAI Press, 2015.
- [16] R. Fagin. Multivalued dependencies and a new normal form for relational databases. *ACM Trans. Database Syst.*, 2(3):262–278, 1977.
- [17] R. Fagin. A normal form for relational databases that is based on domains and keys. *ACM Trans. Database Syst.*, 6(3):387–415, 1981.
- [18] R. Fagin. Horn clauses and database dependencies. *J. ACM*, 29(4):952–985, 1982.
- [19] W. Fan and L. Libkin. On XML integrity constraints in the presence of dtds. *J. ACM*, 49(3):368–406, 2002.
- [20] Z. Galil. An almost linear-time algorithm for computing a dependency basis in a relational database. *J. ACM*, 29(1):96–102, 1982.
- [21] A. Gates, O. Natkovich, S. Chopra, P. Kamath, S. Narayanam, C. Olston, B. Reed, S. Srinivasan, and U. Srivastava. Building a highlevel dataflow system on top of mapreduce: The pig experience. *PVLDB*, 2(2):1414–1425, 2009.

- [22] D. Geiger, A. Paz, and J. Pearl. Axioms and algorithms for inferences involving probabilistic independence. *Information and Computation*, 91(1):128–141, 1991.
- [23] D. Geiger and J. Pearl. Logical and algorithmic properties of conditional independence and graphical models. *The Annals of Statistics*, 21(4):2001–2021, 12 1993.
- [24] E. Grädel and J. Väänänen. Dependence and independence. *Studia Logica*, 101(2):399–410, 2013.
- [25] J. Grant and J. Minker. Inferences for numerical dependencies. *Theor. Comput. Sci.*, 41:271–287, 1985.
- [26] M. Gyssens, M. Niepert, and D. V. Gucht. On the completeness of the semigraphoid axioms for deriving arbitrary from saturated conditional independence statements. *Inf. Process. Lett.*, 114(11):628–633, 2014.
- [27] N. Hall, H. Koehler, S. Link, H. Prade, and X. Zhou. Cardinality constraints on qualitatively uncertain data. *Data & Knowledge Engineering*, 99:126–150, 2015.
- [28] M. Hannula and J. Kontinen. A finite axiomatization of conditional independence and inclusion dependencies. In C. Beierle and C. Meghini, editors, *Foundations of Information and Knowledge Systems - 8th International Symposium, FoIKS 2014, Bordeaux, France, March 3-7, 2014. Proceedings*, volume 8367 of *Lecture Notes in Computer Science*, pages 211–229. Springer, 2014.
- [29] M. Hannula, J. Kontinen, and S. Link. On independence atoms and keys. In J. Li, X. S. Wang, M. N. Garofalakis, I. Soboroff, T. Suel, and M. Wang, editors, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 1229–1238. ACM, 2014.
- [30] S. Hartmann, U. Leck, and S. Link. On Codd families of keys over incomplete relations. *Comput. J.*, 54(7):1166–1180, 2011.
- [31] S. Hartmann and S. Link. Efficient reasoning about a robust XML key fragment. *ACM Trans. Database Syst.*, 34(2), 2009.
- [32] S. Hartmann and S. Link. The implication problem of data dependencies over SQL table definitions: Axiomatic, algorithmic and logical characterizations. *ACM Trans. Database Syst.*, 37(2):13, 2012.
- [33] S. Hartmann, S. Link, and T. Trinh. Solving the implication problem for XML functional dependencies with properties. In A. Dawar and R. J. G. B. de Queiroz, editors, *Logic, Language, Information and Computation, 17th International Workshop, WoLLIC 2010, Brasilia, Brazil, July 6-9, 2010. Proceedings*, volume 6188 of *Lecture Notes in Computer Science*, pages 161–175. Springer, 2010.

- [34] A. Heise, J. Quiané-Ruiz, Z. Abedjan, A. Jentzsch, and F. Naumann. Scalable discovery of unique column combinations. *PVLDB*, 7(4):301–312, 2013.
- [35] C. Herrmann. On the undecidability of implications between embedded multivalued database dependencies. *Information and Computation*, 122(2):221 – 235, 1995.
- [36] C. Herrmann. Corrigendum to ”on the undecidability of implications between embedded multivalued database dependencies” [inform. and comput. 122(1995) 221-235]. *Inf. Comput.*, 204(12):1847–1851, 2006.
- [37] D. S. Johnson and A. C. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. *J. Comput. Syst. Sci.*, 28(1):167–189, 1984.
- [38] H. Köhler, U. Leck, S. Link, and H. Prade. Logical foundations of possibilistic keys. In E. Fermé and J. Leite, editors, *Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings*, volume 8761 of *Lecture Notes in Computer Science*, pages 181–195. Springer, 2014.
- [39] H. Köhler and S. Link. Inclusion dependencies reloaded. In *CIKM*, pages 1361–1370. ACM, 2015.
- [40] H. Köhler, S. Link, and X. Zhou. Possible and certain SQL keys. *PVLDB*, 8(11), 2015.
- [41] S. Kolahi. Dependency-preserving normalization of relational and XML data. *J. Comput. Syst. Sci.*, 73(4):636–647, 2007.
- [42] J. Kontinen, S. Link, and J. A. Väänänen. Independence in database relations. In L. Libkin, U. Kohlenbach, and R. J. G. B. de Queiroz, editors, *WoLLIC*, volume 8071 of *Lecture Notes in Computer Science*, pages 179–193. Springer, 2013.
- [43] W. Langeveldt and S. Link. Empirical evidence for the usefulness of Armstrong relations in the acquisition of meaningful functional dependencies. *Inf. Syst.*, 35(3):352–374, 2010.
- [44] G. Lausen. Relational databases in RDF: Keys and foreign keys. In V. Christophides, M. Collard, and C. Gutierrez, editors, *Semantic Web, Ontologies and Databases, VLDB Workshop, SWDB-ODBS 2007, Vienna, Austria, September 24, 2007, Revised Selected Papers*, volume 5005 of *Lecture Notes in Computer Science*, pages 43–56. Springer, 2008.
- [45] V. Le, S. Link, and F. Ferrarotti. Empirical evidence for the usefulness of Armstrong tables in the acquisition of semantically meaningful SQL constraints. *Data & Knowledge Engineering*, 98:74–103, 2015.

- [46] M. Levene and G. Loizou. How to prevent interaction of functional and inclusion dependencies. *Inf. Process. Lett.*, 71(3-4):115–125, 1999.
- [47] M. Levene and G. Loizou. Guaranteeing no interaction between functional dependencies and tree-like inclusion dependencies. *Theor. Comput. Sci.*, 254(1-2):683–690, 2001.
- [48] M. Levene and M. W. Vincent. Justification for inclusion dependency normal form. *IEEE Trans. Knowl. Data Eng.*, 12(2):281–291, 2000.
- [49] S. Link. Charting the completeness frontier of inference systems for multivalued dependencies. *Acta Inf.*, 45(7-8):565–591, 2008.
- [50] S. Link. Characterisations of multivalued dependency implication over undetermined universes. *J. Comput. Syst. Sci.*, 78(4):1026–1044, 2012.
- [51] S. Link. Frontiers for propositional reasoning about fragments of probabilistic conditional independence and hierarchical database decompositions. *Theor. Comput. Sci.*, 603:111–131, 2015.
- [52] C. L. Lucchesi and S. L. Osborn. Candidate keys for relations. *J. Comput. Syst. Sci.*, 17(2):270–279, 1978.
- [53] D. Maier, A. O. Mendelzon, and Y. Sagiv. Testing implications of data dependencies. *ACM Trans. Database Syst.*, 4:455–469, December 1979.
- [54] J. C. Mitchell. The implication problem for functional and inclusion dependencies. *Information and Control*, 56(3):154–173, 1983.
- [55] M. Niepert, D. V. Gucht, and M. Gyssens. Logical and algorithmic properties of stable conditional independence. *Int. J. Approx. Reasoning*, 51(5):531–543, 2010.
- [56] M. Niepert, M. Gyssens, B. Sayrafi, and D. V. Gucht. On the conditional independence implication problem: A lattice-theoretic approach. *Artif. Intell.*, 202:29–51, 2013.
- [57] G. Paolini and J. Väänänen. Dependence Logic in Pregeometries and ω -Stable Theories. *ArXiv e-prints*, Oct. 2013.
- [58] J. Paredaens. The interaction of integrity constraints in an information system. *J. Comput. Syst. Sci.*, 20(3):310–329, 1980.
- [59] S. V. Petrov. Finite axiomatisation of languages for representation of system properties. *Inf. Sci.*, 47(3):339–372, 1989.
- [60] J. Rissanen. Independent components of relations. *ACM Trans. Database Syst.*, 2(4):317–325, 1977.

- [61] T. Roblot and S. Link. Probabilistic cardinality constraints. In P. Johannesson, M. Lee, S. W. Liddle, A. L. Opdahl, and O. P. López, editors, *Conceptual Modeling - 34th International Conference, ER 2015, Stockholm, Sweden, October 19-22, 2015, Proceedings*, volume 9381 of *Lecture Notes in Computer Science*, pages 214–228. Springer, 2015.
- [62] D. Stott Parker Jr. and K. Parsaye-Ghomi. Inferences involving embedded multivalued dependencies and transitive dependencies. In P. P. Chen and R. C. Sprowls, editors, *Proceedings of the 1980 ACM SIGMOD International Conference on Management of Data, Santa Monica, California, May 14-16, 1980*, pages 52–57. ACM Press, 1980.
- [63] M. Studený. Conditional independence relations have no finite complete characterization. In S. Kubik and J. Visek, editors, *Transactions of the 11th Prague Conference on Information Theory, Statistical Decision Functions and Random Processes*, pages 377–396. Kluwer, 1992.
- [64] M. Studený. *Probabilistic Conditional Independence Structures*. Springer-Verlag, 2005.
- [65] B. Thalheim. *Dependencies in relational databases*. Teubner, 1991.
- [66] J. Väänänen. *Dependence Logic*. Cambridge University Press, 2007.
- [67] M. W. Vincent. Semantic foundations of 4nf in relational database design. *Acta Inf.*, 36(3):173–213, 1999.