Embedded Cardinality Constraints

Ziheng Wei and Sebastian Link(✉)

Department of Computer Science, University of Auckland, Auckland, New Zealand
{zwei891,s.link}@aucklanduni.ac.nz

Abstract. Cardinality constraints express bounds on the number of data patterns that occur in application domains. They improve the consistency dimension of data quality by enforcing these bounds within database systems. Much research has examined the semantics of integrity constraints over incomplete relations in which null markers can occur. Unfortunately, relying on some fixed interpretation of null markers leads frequently to doubtful results. We introduce the class of embedded cardinality constraints which hold on incomplete relations independently of how null marker occurrences are interpreted. Two major technical contributions are made as well. Firstly, we establish an axiomatic and an algorithmic characterization of the implication problem associated with embedded cardinality constraints. This enables humans and computers to reason efficiently about such business rules. Secondly, we exemplify the occurrence of embedded cardinality constraints in real-world benchmark data sets both qualitatively and quantitatively. That is, we show how frequently they occur, and exemplify their semantics.

Keywords: Cardinality constraint · Data and knowledge intelligence
Data quality · Decision support · Missing information

1 Introduction

Background. Cardinality constraints enforce bounds on the number of data patterns that occur in application domains. Cardinality constraints were introduced in Chen's seminal ER paper [3], and have attracted interest and tool support ever since. A cardinality constraint $card(X) \leq b$ stipulates for an attribute set X and a positive integer b that a relation must not contain more than b different tuples with matching values on all the attributes in X. For example, a social worker may not handle more than five cases at any time. This expressiveness makes cardinality constraints invaluable in applications such as data integration, modeling, and processing [16].

Motivation. Most applications require the efficient handling of missing information. This is particularly true in the big data era where large quantities of data (volume) are integrated from heterogenous sources (variety) with different granularity of completeness (veracity). As such, a major challenge in accommodating missing information in the semantics of classical integrity constraints is

J. Krogstie and H. A. Reijers (Eds.): CAiSE 2018, LNCS 10816, pp. 523–538, 2018.
https://doi.org/10.1007/978-3-319-91563-0_32

Table 1. Snippet of the NCVoter data set

id	*v_id*	*f_name*	*l_name*	*gender*	*address*	*city*	*phone*	*register_date*
t_0	480	doris	thompson	f	hwy 119	mebane	5783747	11/05/1940
t_1	612	odessa	teer	f	hwy 119	mebane	$\perp$	5/09/1940
t_2	622	dallie	boswell	f	hwy 119	mebane	$\perp$	5/09/1940
t_3	972	john	smith	m	hwy 119	mebane	$\perp$	10/26/1940
t_4	433	louise	buckner	f	231 s marshall st	graham	2269183	5/04/1940
t_5	577	ruth	albright	f	231 s marshall st	graham	2266060	5/08/1940

the interpretation of null marker occurrences. Indeed, null markers are frequently introduced to integrate data from heterogenous structures in a relation. Previous research has addressed the extension of integrity constraints to incomplete data by uniformly applying one of many possible interpretations to all occurrences of null markers in a relation. As not all null marker occurrences can be interpreted uniformly, in particular in integrated data sets, the results that are derived from such research have only limited applicability. Here, we take a new approach in which the semantics of a cardinality constraint is only dependent on complete fragments that are embedded in a given incomplete relation. Since the definition of our constraints is independent of the interpretation of null markers, they provide a robust approach to describing the semantics of big data, which is fundamental to how such data is processed. We call this new class *embedded cardinality constraints* (eCCs). They consist of a set E of attributes and an ordinary cardinality constraint $card(X) \leq b$ with $X \subseteq E$. The set E specifies the scope r^E of an input relation r on which $card(X) \leq b$ must hold. As such, $card(X) \leq b$ is embedded in r^E. In examples we commonly write $E - X$ instead of E to emphasize on which additional attributes tuples of the scope must be complete. Embedded cardinality constraints provide users with the ability to separate their requirements for the completeness and uniqueness dimensions of data quality. Users specify the set E to declare their completeness requirements, and the cardinality constraint $card(X) \leq b$ with $X \subseteq E$ to declare their uniqueness requirements. As with any constraint, the main target is to improve the consistency of data by enforcing business rules within the database system.

Examples. As an illustration of embedded cardinality constraints we look at the data snippet in Table 1, which is taken from the real-world data set *ncvoter*[1].

An interesting question concerns the number of voters that can live at the same location (address and city). The snippet, and in fact *ncvoter* as a whole, satisfies the embedded cardinality constraint $(\emptyset, card(\{address, city\}) \leq 4)$, since there are at most four different voters that live at the same location and for which address and city have no missing information. However, for marketing campaigns we may only be interested in how many voters can live at the same location that we can reach by telephone, so we are interested in the smallest bound b such that

[1] https://hpi.de/fileadmin/user_upload/fachgebiete/naumann/projekte/repeatability/DataProfiling/FD_Datasets/ncvoter_1001r_19c.csv.

$(\{phone\}, card(\{address, city\}) \leq b)$ holds. For the actual *ncvoter* data set, and therefore for the snippet, this bound is 2, as witnessed by $\{t_4, t_5\}$. Actually, the same tuple block is a showcase that $(\{phone\}, card(\{address, city, gender\}) \leq 2)$ holds. That is, there are up to two voters of the same gender who live at the same location and have a phone number. In contrast, the ordinary cardinality constraint $(\emptyset, card(\{address, city, gender\}) \leq 3)$ says that up to three voters of the same gender live at the same location, for which $\{t_0, t_1, t_2\}$ is a witness.

Contributions. Our contributions can be summarized as follows:

Modeling: We introduce embedded cardinality constraints which hold independently of the interpretation of null marker occurrences in incomplete relations. They subsume previously studied classes of cardinality constraints.

Reasoning: We show that reasoning about embedded cardinality constraints can be done efficiently. Indeed, we characterise the associated implication problem by a finite ground axiomatization and by a linear-time algorithm. Consequently, the gain in expressivity over other classes of constraints does not sacrifice good computational properties. We illustrate how efficient reasoning helps minimise the costs for processing updates, speeds up query evaluation, and prunes the search space when computing the constraints that hold on an incomplete relation.

Case Studies: We illustrate the occurrence of embedded cardinality constraints in actual data sets, previously used as benchmarks for data profiling algorithms. Qualitatively, we present showcases for what embedded cardinality constraints can express in the real world and also provide insight into the lattice structures that these constraints exhibit. Quantitatively, we have implemented a simple heuristics to discover embedded cardinality constraints from an incomplete relation. The heuristics is sound but not complete, so it does not find all embedded cardinality constraints that hold, but those it does find do hold and are minimal. Details of the heuristics are out of scope, but its main purpose is to illustrate that embedded cardinality constraints occur frequently, and that the separation of completeness requirements from uniqueness requirements generates substantial additional patterns that are exhibited by data sets.

Organization. We discuss related work in Sect. 2. Embedded cardinality constraints are introduced in Sect. 3. Some real-world examples and their underlying structure are presented in Sect. 4. Computational problems and their applications are characterized in Sects. 5 and 6. A quantitative analysis of embedded cardinality constraints is presented in Sect. 7. We conclude and mention future work in Sect. 8.

2 Related Work

We demonstrate in this section how embedded cardinality constraints are different from previous work.

Cardinality constraints are an influential contribution of data modeling [16]. They were already present in Chen's seminal paper [3], and are now part of

all major languages for data modeling, including UML, EER, ORM, XSD, and OWL. Cardinality constraints have been extensively studied [2,4,6–13,15]. Since the primary goal of cardinality constraints is to improve consistency, they need to be enforced on actual data sets. Real-world data often exhibits incompleteness in the form of null marker occurrences. This has necessitated the study of (cardinality) constraints over incomplete relations. According to our best knowledge, we only know of the approach that ignores tuples with any null marker occurrence on any column over which an integer bound is specified [8]. This approach, however, is covered by embedded cardinality constraints $(E, card(X) \leq b)$ as the special case where $E = X$. Hence, embedded cardinality constraints handle completeness requirements by specifying E, and they handle uniqueness requirements by stipulating $card(X) \leq b$ where $X \subseteq E$. The previous approach, that is when $E = X$, can only handle both requirements at the same time. Another special case occurs when $X = \emptyset$: here, b stipulates how many tuples in the given relation have no null marker occurrences on any of the columns in E. Furthermore, embedded cardinality constraints also subsume the recently introduced class of contextual keys [17] as the special case where $b = 1$. Embedded cardinality constraints are therefore considerably more expressive than previously studied classes of cardinality constraints. We also exemplify in this article to which degree they occur more frequently in the real-world data than contextual keys. Despite the gain in expressivity, we show that axiomatic and algorithmic reasoning about embedded cardinality constraints is not much more involved than that for contextual keys. More precisely, we can establish a finite ground axiomatization and a linear time algorithm to decide the implication problem for embedded cardinality constraints. These subsume those established recently for contextual keys as the special case where $b = 1$ for each given embedded cardinality constraint. Recently, cardinality constraints have also been investigated for uncertain data models, including possibilistic models [5] and probabilistic models [14]. These are orthogonal directions of research about cardinality constraints, but possibilistic and probabilistic embedded cardinality constraints can be investigated in future work.

3 Embedded Cardinality Constraints

We fix concepts from relational databases and introduce the central notion of embedded cardinality constraints.

A *relation schema* is a finite set R of attributes A. Each attribute A is associated with a domain $dom(A)$ of values. Based on the demand in traditional and modern applications, data models need to accommodate missing information. In order to represent the standard approach adapted by relational database technology, we assume that the domain $dom(A)$ of every attribute A contains the distinguished symbol $\perp$, representing the null marker. We stress that the null marker is not a domain value, and is only included in the domain of attributes for convenience and ease of discussion. A tuple t over R is a function that assigns to each attribute A of R an element $t(A)$ from the domain $dom(A)$. For an attribute

set X, a tuple t is said to be X-total whenever $t(A) \neq \perp$ for all $A \in X$. A *relation* over R is a finite set of tuples over R. An expression $card(X) \leq b$ with some subset $X \subseteq R$ and a positive integer $b \in \mathbb{N}$ is called a *cardinality constraint over* R. A cardinality constraint $card(X) \leq b$ over R is said to *hold* in a relation r of R, denoted by $r \models card(X) \leq b$, if and only if there are not $b+1$ different tuples $t_1, \cdots, t_{b+1} \in r$ such that for all $1 \leq i < j \leq b+1$, $t_i \neq t_j$ and for all $A \in X$, $t_i(A) = t_j(A) \neq \perp$.

Note that this simple model is already expressive enough to address at least three dimensions of big data: volume is represented by the numbers of columns and rows in a relation, veracity is represented by null marker occurrences in relations, and variety is represented by the ability to integrate information from different sources and of different structure by putting domain values into columns where they are known for and null marker occurrences where they are not.

A critical issue in extending integrity constraints to data models with incomplete information is the way in which null marker occurrences are handled. One popular approach is to assign a particular semantics to the occurrences, such as 'value exists but is currently unknown' or 'value does not exist' or 'no information'. In practice, that is in SQL, there is no room to associate different interpretations with different occurrences: Only one universal interpretation is assigned to every occurrence. With this limitation it is difficult to obtain meaningful results when the data is used. This is particularly true for applications that employ integrated data where some occurrences of null markers are bound to have different interpretation. Nevertheless, a plethora of research has been conducted in this area, resulting in different notions of constraints. In contrast, this article follows a complementary approach in which constraints are evaluated independently of any null marker occurrences. That is, the satisfaction of the cardinality constraints is only dependent on the complete fragments in incomplete relations. This has the strong advantage that the results obtained from any use of the constraints are robust under varying interpretations of null marker occurrences. With this motivation in mind, we will now introduce the central notion of embedded cardinality constraints.

Definition 1. *An* embedded cardinality constraint *(*eCC*) over relation schema R is an expression $(E, card(X) \leq b)$ where $X \subseteq E \subseteq R$ and $b \in \mathbb{N}$. We call E the* extension *and $card(X) \leq b$ the cardinality constraint associated with the eCC. For a relation r over R, the extension E defines the* scope r^E *of the eCC as $r^E = \{t \in r \mid t \text{ is } E - total\}$. The eCC $(E, card(X) \leq b)$ over R is* satisfied by*, or said to* hold in*, the relation r over R if and only if the scope r^E satisfies the cardinality constraint $card(X) \leq b$ associated with the eCC.*

We sometimes simply write $(E - X, card(X) \leq b)$ instead of $(E, card(X) \leq b)$ in order to save space or emphasize the (non-)existence of additional attributes in the extension. The introduction has already presented several real-world examples of embedded cardinality constraints. The following section provides further insight into the expressivity of this new class of constraints.

4 Real-World Examples with Embedded Lattice View

We illustrate the business rules that embedded cardinality constraints can express, and illustrate the inherent structure that these constraints exhibit. The latter can be exploited as a navigational aid that assists users in understanding, representing and browsing cardinality profiles of their data. As an interesting special case, we present completeness cubes as a navigational aid that users can employ as a model of how many tuples are complete on a given set of attributes. We use the public data set *bridges* as a running example.

The Pittsburgh bridge data set, *bridges*, is a popular reference data set on the UCI machine learning repository[2]. It provides some basic information about 108 different bridges in Pittsburgh. Table 2 shows 14 tuples from the full data set where some columns were removed to focus on the attributes that matter for the embedded cardinality constraints we would like to discuss.

For example, we may want to know the maximum number of bridges that lead over the same river, were built for the same purpose, and are of the same type. Indeed, 14 is the answer, which is the smallest upper bound b with which the eCC $(\emptyset, card(\{river, purpose, type\}) \leq b)$ is satisfied by *bridges*. We may wonder for how many of those the length and the number of lanes are both known. The relevant eCC would be $(\{length, lanes\}, card(\{river, purpose, type\}) \leq 8)$.

For ordinary cardinality constraints the integer bounds are non-increasing with an increasing number of attributes, as illustrated on the left of Fig. 1. For eCCs, additional attributes in the extension E generate an embedded lattice

Table 2. Snippet of the bridges data set

id	*river*	*loccation*	*erected*	*purpose*	*length*	*lanes*	*rel-l*	*type*
E17	M	4	1863	RR	1000	2	⊥	SIMPLE-T
E21	M	16	1874	RR	⊥	2	⊥	SIMPLE-T
E25	M	10	1882	RR	⊥	2	⊥	SIMPLE-T
E26	M	12	1883	RR	1150	2	S	SIMPLE-T
E31	M	8	1887	RR	1161	2	S	SIMPLE-T
E37	M	18	1891	RR	1350	2	S	SIMPLE-T
E45	M	14	1897	RR	2264	⊥	F	SIMPLE-T
E47	M	15	1898	RR	2000	2	S	SIMPLE-T
E94	M	13	1901	RR	⊥	2	F	SIMPLE-T
E95	M	16	1903	RR	1300	2	S	SIMPLE-T
E51	M	6	1903	RR	1417	2	F	SIMPLE-T
E50	M	21	1903	RR	1154	⊥	F	SIMPLE-T
E89	M	4	1904	RR	1200	2	S-F	SIMPLE-T
E92	M	10	1914	RR	2210	⊥	F	SIMPLE-T

[2] https://archive.ics.uci.edu/ml/index.php.

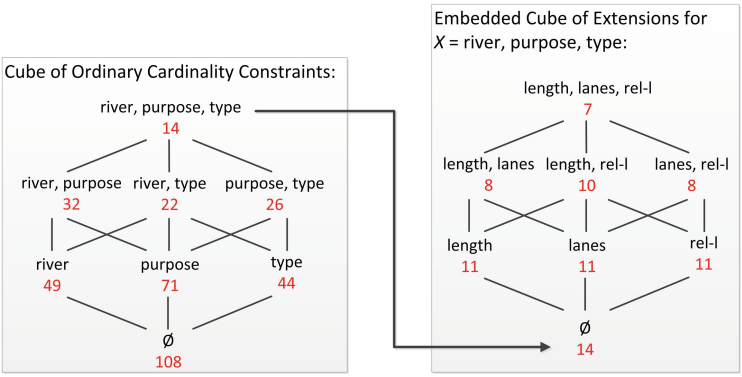

Fig. 1. Embedded lattice of extensions for each ordinary cardinality constraint

structure, for each fixed set X of attributes. Indeed, if E increases, the corresponding bounds cannot increase. This is illustrated on the right of Fig. 1. Table 2 contains those tuples that generate all the integer bounds marked red in the right of Fig. 1.

An interesting special case of these lattices are given by eCCs of the type $(E, card(\emptyset) \leq b)$. These stipulate upper bounds on the numbers of tuples that are E-total. Figure 2 shows these bounds for the data set *bridges*, based on all combinations of the four attributes *length*, *lanes*, *span*, and *rel-l*. For example, there are 90 tuples that are total on *span* and *rel-l*.

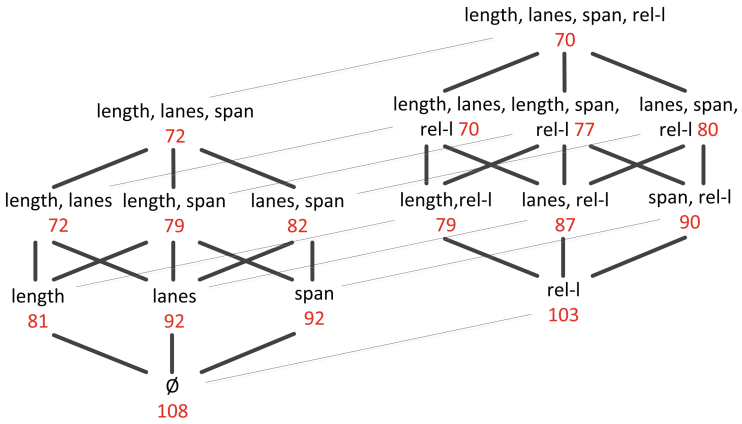

Fig. 2. Lattice of cardinality constraints for completeness dimensions

5 Axiomatic Characterization of the Implication Problem

We establish a finite ground axiomatization for the implication problem of embedded cardinality constraints. This will enable us to effectively enumerate all implied eCCs, that is, to determine the semantic closure $\Sigma^* = \{\sigma \mid \Sigma \models \sigma\}$ of any given eCC set Σ. A finite axiomatization facilitates human understanding of the interaction of the given constraints, and ensures all opportunities for the use of these constraints in applications can be exploited. We comment on a couple of direct application areas for the axiomatization.

In using an axiomatization we determine the semantic closure by applying *inference rules* of the form $\frac{\text{premise}}{\text{conclusion}}$. Since no conditions are stipulated for the application of these inference rules, the resulting axiomatization is called a ground axiomatization. For a set $\mathfrak{R}$ of inference rules let $\Sigma \vdash_{\mathfrak{R}} \varphi$ denote the *inference* of φ from Σ by $\mathfrak{R}$. That is, there is some sequence $\sigma_1, \ldots, \sigma_n$ such that $\sigma_n = \varphi$ and every σ_i is an element of Σ or is the conclusion that results from an application of an inference rule in $\mathfrak{R}$ to some premises in $\{\sigma_1, \ldots, \sigma_{i-1}\}$. Let $\Sigma^+_{\mathfrak{R}} = \{\varphi \mid \Sigma \vdash_{\mathfrak{R}} \varphi\}$ be the *syntactic closure* of Σ under inferences by $\mathfrak{R}$. $\mathfrak{R}$ is *sound* (*complete*) if for every set Σ over every R we have $\Sigma^+_{\mathfrak{R}} \subseteq \Sigma^*$ ($\Sigma^* \subseteq \Sigma^+_{\mathfrak{R}}$). The (finite) set $\mathfrak{R}$ is a (finite) *axiomatization* if $\mathfrak{R}$ is both sound and complete. Table 3 shows the axiomatization $\mathfrak{C}$ for eCCs that we will prove to be sound and complete. Here, R denotes an arbitrarily given underlying relation schema, $E, E', X, X' \subseteq R$, and b, b' are positive integers.

Theorem 1. *The set $\mathfrak{C} = \{\mathcal{B}, \mathcal{E}, \mathcal{S}, \mathcal{T}\}$ is sound and complete for the implication problem of embedded cardinality constraints.*

We note that the rules

$$\frac{}{card(R) \leq 1} \qquad \frac{card(X) \leq b}{card(XX') \leq b} \qquad \frac{card(X) \leq b}{card(X) \leq b'}$$

are sound and complete for the implication of ordinary cardinality constraints [7,8], and are embedded in our inference rules $\mathcal{T}$, $\mathcal{S}$, and $\mathcal{B}$.

Table 3. The finite ground axiomatization $\mathfrak{C} = \{\mathcal{B}, \mathcal{E}, \mathcal{S}, \mathcal{T}\}$ of eCCs

$\frac{}{(R, card(R) \leq 1)}$ (trivially embedded keys, $\mathcal{T}$)	$\frac{(E, card(X) \leq b)}{(EE', card(X) \leq b)}$ (super extension, $\mathcal{E}$)
$\frac{(E, card(X) \leq b)}{(E, card(XX') \leq b)}$ (super set, $\mathcal{S}$)	$\frac{(E, card(X) \leq b)}{(E, card(X) \leq b + b')}$ (weaker bound, $\mathcal{B}$)

Proofs of Soundness and Completeness. Let $\Sigma \cup \{\varphi\}$ denote a set of eCCs over a given relation schema R.

Soundness. We need to show that every eCC φ that can be inferred from a given eCC set Σ by $\mathfrak{C}$ is also implied by Σ. Let r denote a relation of the given relation schema R. It suffices to show the following for every inference rule in $\mathfrak{C}$: If the premise of the rule is satisfied by r, then the conclusion of the rule is also satisfied by r.

For the soundness of $\mathcal{T}$ we observe that the scope r^R contains all tuples of r that are complete on all attributes of R. Since r^R is a set, r^R is also a set and, consequently, there cannot be two different tuples that have matching values on all the attributes of R. Hence, r^R satisfies $card(R) \leq 1$.

For the soundness of $\mathcal{E}$ we assume that $r \models (E, card(X) \leq b)$. By definition, $r^E \models card(X) \leq b$. Consequently, there cannot be $b+1$ different tuples in r^E that all have matching values on all the attributes in X. For every subset $E' \subseteq R$, $r^{EE'}$ is a subset of r^E. Consequently, there cannot be $b+1$ different tuples in $r^{EE'}$ that all have matching values on all the attributes in X. It follows that $r^{EE'}$ satisfies $card(X) \leq b$. By definition, r satisfies $(EE', card(X) \leq b)$.

For the soundness of $\mathcal{S}$ we assume that r satisfies $(E, card(X) \leq b)$. By definition, r^E satisfies $card(X) \leq b$. Consequently, there cannot be $b+1$ different tuples in r^E that all have matching values on all the attributes in X. For every subset $X' \subseteq R$, X is a subset of XX'. Consequently, there cannot be $b+1$ different tuples in r^E that all have matching values on all the attributes in XX'. Indeed, otherwise there would have to be $b+1$ different tuples in r^E that all have matching values on all the attributes in X. It follows that r^E satisfies $card(XX') \leq b$. By definition, r satisfies $(E, card(XX') \leq b)$.

For the soundness of $\mathcal{B}$ we assume that r satisfies $(E, card(X) \leq b)$. By definition, r^E satisfies $card(X) \leq b$. Consequently, there cannot be $b+1$ different tuples in r^E that all have matching values on all the attributes in X. For every non-negative integer b', b is at most $b+b'$. Consequently, there cannot be $(bb')+1$ different tuples in r^E that all have matching values on all the attributes in X. Indeed, otherwise there would already be $b+1$ different tuples in r^E that all have matching values on all the attributes in X. It follows that r^E satisfies $card(X) \leq b + b'$. By definition, r satisfies $(E, card(X) \leq b + b')$.

Completeness. We need to show that every φ that is implied by Σ can also be inferred from Σ by the use of inference rules in $\mathfrak{C}$ only. We show the contraposition, that is, we assume that φ cannot be inferred from Σ by $\mathfrak{C}$ and construct a relation over R that satisfies Σ but violates φ. Let $\varphi = (E, card(X) \leq b)$ such that $\Sigma \vdash_{\mathfrak{C}} \varphi$ does not hold. We define a relation r over R that consists of $b+1$ different tuples as follows: For $R - E = \{A_0, \ldots, A_{n-1}\}$ and $j = 0, \ldots, b$, tuple $t_j(A) := \begin{cases} 0, & \text{if } A \in X \\ j, & \text{if } A \in E - X \\ j, & \text{if } A = A_j \\ \bot, & \text{if } A \in R - (E \cup \{A_j\}) \end{cases}$. The relation r may look as follows:

$E-X$	X	$R-E$
$0\cdots0$	$0\cdots0$	$0\ \bot\quad\bot\quad\bot\cdots\quad\bot\ \bot$
$1\cdots1$	$0\cdots0$	$\bot\ 1\quad\bot\quad\bot\cdots\quad\bot\ \bot$
$\vdots$	$\vdots$	$\vdots$
$b\cdots b$	$0\cdots0$	$\bot\cdots\quad\bot\quad b\ \bot\ \cdots\ \bot$

The relation is well-defined, that is, contains $b+1$ different tuples for the following reason. If $R-E=\emptyset$ and $E-X=\emptyset$, then $\varphi=(R, card(R)\leq b)$. However, $(R, card(R)\leq 1)\in\Sigma^+_{\mathfrak{C}}$ by application of $\mathcal{T}$, and this would lead to $\varphi\in\Sigma^+_{\mathfrak{C}}$ by application of $\mathcal{B}$. Hence, $R-E\neq\emptyset$ or $E-X\neq\emptyset$, and $|r|=b+1$.

The relation does not satisfy φ since $r^E=r$ and r contains $b+1$ different tuples with matching values on all the attributes in X.

It remains to show that r satisfies every $\sigma=(E', card(X')\leq b')\in\Sigma$. Assume that r violates σ. Then $E'\subseteq E$ (as otherwise $|r^{E'}|\leq 1$ and r would satisfy $(E', card(X')\leq 1)$ and by soundness of $\mathfrak{B}$ also σ), $X'\subseteq X$ (as otherwise all tuples would have different projections on X, so r would satisfy $(E', card(X')\leq 1)$ and by soundness of $\mathfrak{B}$ also σ), and $b'\leq b$ (as otherwise there couldn't be $b'+1$ tuples to violate σ). Consequently, we can apply $\mathcal{E}$, $\mathcal{S}$, and $\mathcal{B}$ to $\sigma=(E', card(X')\leq b')\in\Sigma$ to obtain $(E, card(X)\leq b)\in\Sigma^+_{\mathfrak{C}}$, a contradiction. Consequently, our assumption that r violates σ must have been wrong. Since σ was chosen arbitrarily we have just shown that r satisfies all elements of Σ and violates φ. We conclude that φ is not implied by Σ.

Applications. While axiomatizations facilitate human understanding of how to reason, there are also a number of more tangible applications. This is not surprising, as axiomatizations are usually taken as the first step towards developing automated reasoning tools. Indeed, axiomatizations are commonly employed to develop algorithms that can decide associated implication problems. This, in turn, has numerous applications. The next section deals directly with the development of algorithmic characterizations of the implication problem for eCCs.

Algorithms can decide instances of implication problems efficiently, but they simply return either true or false. People often wonder how an algorithm derived at a particular decision. Here, axiomatizations can provide additional insight. If the answer yes, then a derivation of the candidate constraint from the given constraint set must exist. More intriguingly, if the answer is no, our completeness proof is guaranteed to provide users with an example relation that shows why the candidate constraint is not implied by the given constraint set. The completeness proof can be converted into a tool that constructs such an example relation automatically, whenever the decision algorithm returns false.

As a second application we mention the discovery problem (aka dependency mining or data profiling), in which an algorithm ought to return all those constraints from a given class that hold in a given relation [1]. Quite frequently, the search and solution spaces are massive, which makes it necessary to derive effective pruning strategies that decrease the search space and allow solutions to the discovery problem to be returned efficiently. Here, sound inference rules can directly be translated into pruning strategies. In fact, if an eCC $(E, card(X)\leq b)$

Algorithm 1. Inference

Require: $R, \Sigma, (E, \mathit{card}(X))$ with a set Σ of embedded cardinality constraints
Ensure: $\min\{b : \Sigma \models (E, \mathit{card}(X) \leq b)\}$
1: **if** $E = R$ and $X = R$ **then**
2: **return** 1;
3: **else**
4: $b \leftarrow \infty$;
5: **for all** $(E', \mathit{card}(X') \leq b') \in \Sigma$ **do**
6: **if** $E' \subseteq E$ and $X' \subseteq X$ and $b' < b$ **then**
7: $b \leftarrow b'$;
8: **return** b;

has been validated to hold on the input relation, then every check of any eCC $(E', \mathit{card}(X') \leq b')$ where $E \subseteq E'$, $X \subseteq X'$, and $b \leq b'$ hold is redundant and should be omitted. Having a complete axiomatization ensures that all pruning strategies are known.

6 Algorithmic Characterization

In this section we develop algorithmic tools that decide the implication problem for embedded cardinality constraints in linear time in the input. As outlined before, this complements our axiomatization established in the last section.

Indeed, computing Σ^* and checking whether $\varphi \in \Sigma^*$ is not an efficient approach towards deciding the implication problem. The following theorem allows us to decide the implication problem for embedded cardinality constraints with a single scan of the input. Note that the proof employs the construction from the completeness proof of our axiomatization.

Theorem 2. *Let $\Sigma \cup \{(E, \mathit{card}(X) \leq b)\}$ denote a set of eCCs over R. Then Σ implies $(E, \mathit{card}(X) \leq b)$ iff (i) $E = R$ and $X = E$ or (ii) there is some $(E', \mathit{card}(X') \leq b') \in \Sigma$ such that $E' \subseteq E$, $X' \subseteq X$, and $b' \leq b$ hold.*

Proof. If (i) or (ii) hold, then $(E, \mathit{card}(X) \leq b)$ can be inferred from Σ by $\mathfrak{C}$. Consequently, the soundness of $\mathfrak{C}$ ensures that $(E, \mathit{card}(X) \leq b)$ is implied by Σ.

Vice versa, assume that neither (i) nor (ii) hold. Invalidity of (i) ensures that $R - E \neq \emptyset$ or $E - X \neq \emptyset$ holds. This guarantees that the relation r from the completeness proof has $b + 1$ different tuples. Invalidity of (ii) ensures that r satisfies Σ. Since the relation also violates $(E, \mathit{card}(X) \leq b)$ by construction, it follows that Σ does not imply $(E, \mathit{card}(X) \leq b)$.

Instead of translating Theorem 2 directly into a decision algorithm, we prefer to establish a linear-time algorithm for the more general computational problem that computes for a given set Σ of eCCs over a given relation schema R, and a given attribute set pair (E, X) with $X \subseteq E \subseteq R$ the minimum positive integer b (or $b = \infty$ if no integer exists) such that $(E, \mathit{card}(X) \leq b)$ is still implied by Σ.

Algorithm 1 computes this supremum b as follows: if $E = X = R$, then $b = 1$ is returned according to axiom $\mathcal{T}$ of our axiomatization $\mathfrak{C}$. Otherwise, all input eCCs from Σ are scanned and the current supremum b is revised to b' whenever an eCC is found whose extension E', attribute set X', and bound b' satisfy $E' \subseteq E$, $X' \subseteq X$, and $b' < b$. This is valid due to the remaining inference rules in $\mathfrak{C}$. If no appropriate input eCC is found, then ∞ is returned. The total number of attributes that occur in Σ and R are denoted by $|\Sigma|$ and $|R|$, respectively.

Theorem 3. *On input* $(R, \Sigma, (E, card(X)))$*, Algorithm 1 returns in* $\mathcal{O}(|\Sigma|+|R|)$ *time the supremum* b *with which* $(E, card(X) \leq b)$ *is implied by* Σ*.*

Proof. The correctness of Algorithm 1 follows directly from Theorem 2. For the time complexity, we only require one scan over all input attributes in Σ plus the input attributes in (E, X). The latter could be provided in the format $(E-X, X)$ ensuring that every attribute in R only occurs once.

Algorithm 1 can directly be used to decide the implication problem of eCCs. Indeed, given an eCC set $\Sigma \cup \{(E, card(X) \leq b)\}$ over relation schema R, such a decision algorithm will return yes if and only if $b \geq b'$ where b' is returned by Algorithm 1 on input $(R, \Sigma, (E, card(X)))$.

Corollary 1. *The implication problem of embedded cardinality constraints can be decided in time linear in the input.* □

Applications. Our algorithm has direct applications in saving update and query costs. When updating a data set, we need to ensure that the resulting data set satisfies all the eCCs that have been established as meaningful business rules of the underlying application domain. Validating the satisfaction of any business rule that is implied by the remaining rules is a waste of time. Being able to detect implied rules enables us to minimal set of business rules in which none is implied by the rest, thereby ensuring a minimal overhead in maintaining the consistency of data sets under updates. For example, if we have already validated that a data set satisfies $\sigma = (\{phone\}, card(address, city) \leq 2)$, then there is no need to validate that it satisfies $\sigma' = (\{phone, register\_date\}, card(address, city, gender) \leq 3)$. When querying *ncvoter* one may ask to return the voter-id of all voters who live at locations where no more than 3 voters of the same gender live for whom phone numbers and registration dates are known. Being aware that *ncvoter* satisfies the eCC σ and deciding that σ implies σ', the original query can automatically be optimized to the query that returns the voter-id of voters.

7 Quantitative Analysis of Our Real-World Data Sets

This section provides some quantitative insight into the occurrence of embedded and ordinary constraints in five real-world data sets from the UCI machine learning repository. These data sets are frequently used to test the performance of data dependency discovery algorithms [1]. We have implemented a heuristic to discover embedded cardinality constraints from incomplete relations. The

heuristic is sound as the eCCs it finds are guaranteed to hold on the given data set and also minimal. The heuristic is not complete, so it is not guaranteed to find all eCCs that hold on the given data set. The point of the heuristic is to show that eCCs occur frequently in real-world data, which we illustrate by the sheer number of their occurrences and also by comparing that to the number of occurrences of ordinary cardinality constraints (oCCs), that is eCCs with an empty extension, embedded uniqueness constraints (eUCs), that is eCCs where $b = 1$, and ordinary uniqueness constraints (oUCs), that is eCCs with an empty extension and where $b = 1$.

Table 4. Characteristics of data sets and numbers of oCCs, pCCs, oUCs, pUCs

	$\#r$	$\#c$	$\#\perp$	$\#ir$	$\#ic$	$\#oCCs$	$\#pCCs$	$\#oUCs$	$\#pUCs$
breast	691	11	16	16	1	557	259	1	1
bridges	108	13	77	38	9	301	1877	0	3
echo	132	13	132	71	12	135	1668	18	27
hepatitis	155	20	167	75	15	312	1262	344	102
ncvoter	1000	19	2863	1000	5	438	976	78	69

7.1 Occurrences of Ordinary and Embedded Constraints

Table 4 shows some characteristics of the five data sets[3] we analyzed: the number of rows ($\#r$), columns ($\#c$), null marker occurrences ($\#\perp$), incomplete rows ($\#ir$), and incomplete columns ($\#ic$).

Our heuristic revealed the number of oCCs ($\#oCCs$), which are eCCs where $E = \emptyset$, and the number of pure $eCCs$ ($\#pCCs$), which are eCCs where $E \neq \emptyset$. In previous work we had developed algorithms that determine the total number of oUCs ($\#oUCs$), which are eCCs where $E = \emptyset$ and $b = 1$, and the number of pure eUCs ($\#pUCs$), which are eCCs where $E \neq \emptyset$ and $b = 1$. While $\#pCCs$ and $\#oCCs$ are lower bounds based on our heuristic, $\#pUCs$ and $\#oUCs$ are actual numbers of a sound and complete algorithm from previous work.

With the exception of *breast* our heuristic has found many more pure eCCs than oCCs. Even though the numbers of (ordinary and pure) eCCs are just lower bounds and that of (ordinary and pure) eUCs are exact, the analysis gives an indication of how many more business rules can be expressed by eCCs in comparison to eUCs.

7.2 Cardinality Histograms for All Data Sets

For additional insight we have analyzed the distribution of the integer bounds (cardinalities) in the eCCs we were able to discover. The results are visualized in Fig. 3. The distributions are skewed towards lower cardinalities, which is natural since projections with larger cardinalities are typically less frequent. The

[3] https://hpi.de/naumann/projects/repeatability/data-profiling/fds.html#c168191.

Fig. 3. Distribution of cardinalities on data sets

distributions for pure eCCs are less skewed than the distributions for ordinary eCCs, indicating that the independence of the completeness requirements (as expressed by non-trivial extensions) generates substantial additional constraints with diverse cardinalities.

8 Conclusion and Future Work

We have introduced the new class of embedded cardinality constraints. Their most interesting feature is their independence of any interpretation of missing information, which makes their employment for applications robust in the context of integrated big data sets. Despite the ability of embedded cardinality constraints to express previous classes of constraints as special cases, we showed that embedded cardinality constraints enjoy a finite ground axiomatization and their implication problem can be decided in linear time in the input. This makes their application also effective, as all opportunities of employment

can be efficiently checked automatically. In addition, we have exemplified their expressivity on real-world data sets, visualized the interaction they exhibit in the form of embedded lattice structures, and provided quantitative evidence of their frequent occurrence in practice.

There are many more interesting problems associated with embedded cardinality constraints, including their discovery problem and the computation of Armstrong relations. Solutions to these two problems would provide computational support towards the acquisition of embedded cardinality constraints that are meaningful in a given application domain. Other problems include the interaction with other constraints, such as functional dependencies, or the definition of embedded cardinality constraints in models for Web or uncertain data.

References

1. Abedjan, Z., Golab, L., Naumann, F.: Profiling relational data: a survey. VLDB J. **24**(4), 557–581 (2015)
2. Calvanese, D., Lenzerini, M.: On the interaction between ISA and cardinality constraints. In: Proceedings of the Tenth International Conference on Data Engineering, Houston, Texas, USA, 14–18 February 1994, pp. 204–213. IEEE Computer Society (1994)
3. Chen, P.P.: The Entity-Relationship model - toward a unified view of data. ACM Trans. Database Syst. **1**(1), 9–36 (1976)
4. Ferrarotti, F., Hartmann, S., Link, S.: Efficiency frontiers of XML cardinality constraints. Data Knowl. Eng. **87**, 297–319 (2013)
5. Hall, N., Köhler, H., Link, S., Prade, H., Zhou, X.: Cardinality constraints on qualitatively uncertain data. Data Knowl. Eng. **99**, 126–150 (2015)
6. Hartmann, S.: On the implication problem for cardinality constraints and functional dependencies. Ann. Math. Artif. Intell. **33**(2–4), 253–307 (2001)
7. Hartmann, S.: Reasoning about participation constraints and Chen's constraints. In: Schewe, K., Zhou, X. (eds.) Proceedings of the 14th Australasian Database Conference on Database Technologies, ADC 2003, Adelaide, South Australia, February 2003. CRPIT, vol. 17, pp. 105–113. Australian Computer Society (2003)
8. Hartmann, S., Köhler, H., Leck, U., Link, S., Thalheim, B., Wang, J.: Constructing Armstrong tables for general cardinality constraints and not-null constraints. Ann. Math. Artif. Intell. **73**(1–2), 139–165 (2015)
9. Jones, T.H., Song, I.Y.: Analysis of binary/ternary cardinality combinations in Entity-Relationship modeling. Data Knowl. Eng. **19**(1), 39–64 (1996)
10. Liddle, S.W., Embley, D.W., Woodfield, S.N.: Cardinality constraints in semantic data models. Data Knowl. Eng. **11**(3), 235–270 (1993)
11. McAllister, A.J.: Complete rules for n-ary relationship cardinality constraints. Data Knowl. Eng. **27**(3), 255–288 (1998)
12. Queralt, A., Artale, A., Calvanese, D., Teniente, E.: OCL-Lite: finite reasoning on UML/OCL conceptual schemas. Data Knowl. Eng. **73**, 1–22 (2012)
13. Roblot, T.K., Link, S.: Urd: a data summarization tool for the acquisition of meaningful cardinality constraints with probabilistic intervals. In: 33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, 19–22 April 2017, pp. 1379–1380. IEEE Computer Society (2017)

14. Roblot, T.K., Link, S.: Cardinality constraints with probabilistic intervals. In: Mayr, H.C., Guizzardi, G., Ma, H., Pastor, O. (eds.) ER 2017. LNCS, vol. 10650, pp. 251–265. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69904-2_21
15. Thalheim, B.: Fundamentals of cardinality constraints. In: Pernul, G., Tjoa, A.M. (eds.) ER 1992. LNCS, vol. 645, pp. 7–23. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-56023-8_3
16. Thalheim, B.: Integrity constraints in (conceptual) database models. In: Kaschek, R., Delcambre, L. (eds.) The Evolution of Conceptual Modeling. LNCS, vol. 6520, pp. 42–67. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-17505-3_3
17. Wei, Z., Link, S., Liu, J.: Contextual keys. In: Mayr, H.C., Guizzardi, G., Ma, H., Pastor, O. (eds.) ER 2017. LNCS, vol. 10650, pp. 266–279. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69904-2_22