

Efficient Reasoning About XFDs with Pre-image Semantics

Sven Hartmann, Sebastian Link, and Thu Trinh

Information Science Research Centre, Department of Information Systems
Massey University, Palmerston North, New Zealand
{s.hartmann, s.link, t.trinh}@massey.ac.nz

Abstract. The study of integrity constraints has been identified as one of the major challenges in XML database research. The main difficulty is finding a balance between the expressiveness and the existence of automated reasoning tools. We investigate a previous proposal for functional dependencies in XML (XFDs) that is based on homomorphisms between data trees and schema trees. We demonstrate that reasoning about our XFDs is well-founded. We provide a finite axiomatisation and show that their implication is equivalent to the logical implication of propositional Horn clauses and thus decidable in time linear in the size of the constraints. Hence, our XFDs do not only capture valuable semantic information but also permit efficient automated reasoning support.

1 Introduction

The importance of XML integrity constraints is due to a wide range of applications ranging from schema design, query optimisation, efficient storing and updating, data exchange and integration, to data cleaning [3]. Several classes of integrity constraints have been defined for XML including functional dependencies [1,4,5,6,8,9,10,11]. While there is a well-accepted single concept for the notion of functional dependency in relational databases the complex nature of XML data has resulted in various proposals for XFDs that deviate in their expressiveness but are all justified as they naturally occur in XML data.

For an example consider the XML data tree in Figure 1 that stores simple purchase profiles showing customers, the items they bought (an item is a pair consisting of an article and its price) and the discount received for the purchase. In the data tree the same articles have the same price. This observation is likely to be called a functional dependency between the article and its price. In Figure 2, this functional dependency is no longer valid. Still, the data stored in this tree is not independent from one another: whenever two customers have purchased all the same items then they both receive the same discount. That is, the set of items purchased functionally determines the discount. This dependency does not occur just accidentally but captures important semantic information that should be satisfied by every legal data tree of this form. *Lisa* might have received a discount of 0.5\$ since *Kiwis* for the price of 2\$ were on special.

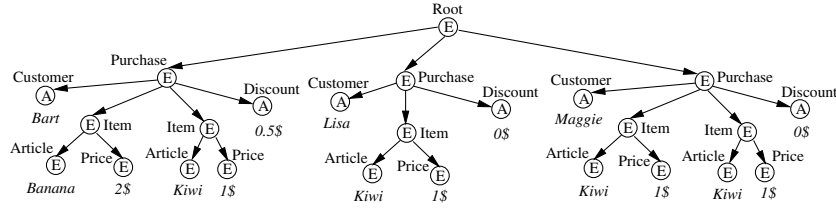


Fig. 1. XML data tree exhibiting some functional dependency

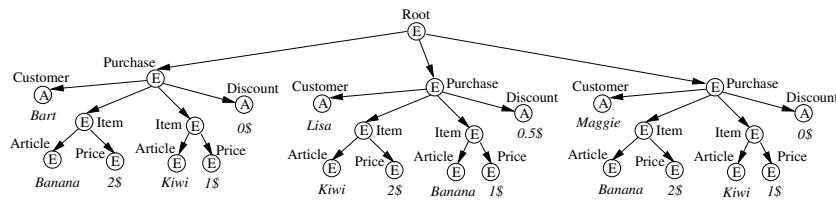


Fig. 2. Another XML data tree exhibiting another kind of functional dependency

The majority of proposals has considered the first kind of XFDs [1,6,9] which is reminiscent of earlier research on path-based dependencies in semantic and object-oriented data models, while this paper studies the second kind [4,10]. We use the simple XML graph model from [4,5]. An XML tree is a rooted tree T with node set V_T , arc set A_T , root r_T , and mappings $name : V_G \rightarrow Names$ and $kind : V_G \rightarrow \{E, A\}$. The symbols E and A indicate elements and attributes. A data tree is an XML tree T' with string values assigned to its leaves. Two data trees T' and T are value-equal if there is a value-preserving isomorphism between them. A schema tree is an XML tree T where no two siblings have the same name and kind, and with frequencies assigned to its arcs.

A v -walk of an XML tree T is a directed path from a fixed node v to some leaf of T . A v -subgraph of T is the union of v -walks of T . Clearly, a v -subgraph is an XML tree again. The empty v -subgraph is denoted by $\emptyset_{T,v}$. The total v -subgraph $T(v)$ is the union of all v -walks of T . Consider two XML trees T' and T with a homomorphism between them. An $r_{T'}$ -subgraph U' of T' is a subcopy of T if

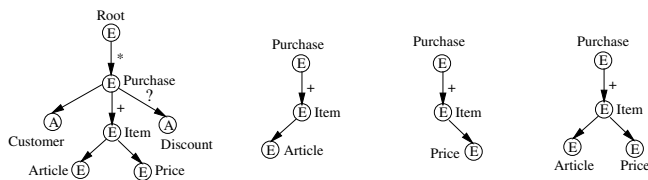


Fig. 3. A schema tree with arc labels for the frequencies, and three of its $v_{Purchase}$ -subgraphs: the $v_{Purchase}$ -walks $[[Article]]$ and $[[Price]]$, and their union $[[Article, Price]]$ (for convenience, we use an example where walks can be identified by their leaf names)

U' is isomorphic to some r_T -subgraph U of T , and an *almost-copy* of T if it is maximal with this property. Given an r_T -subgraph U of T , the *projection* of T' to U is the union of all subcopies of U in T' , and denoted by $T'|_U$.

In relational databases, a functional dependency $X \rightarrow Y$ is satisfied if and only if any two tuples that agree on their projections to X also agree on their projections to Y . Surprisingly, it is not that obvious how to translate the concept of functional dependency to XML. Most importantly, one has to decide what the tuples in an XML data tree should be. Arenas/Libkin [1] suggested to consider almost-copies of a schema tree T in a T -compatible data tree T' as tree-tuples. Almost-copies are of interest as T' does not necessarily contain copies of T . XFDs of this kind have been studied in detail, e.g., in [1,5,8,9,11].

2 Deciding Implication of XFDs Based on Pre-images

The homomorphism between a T -compatible data tree T' and a schema tree T induces a mapping of the total subgraphs of T' to the total subgraphs of T . For a fixed node v of T , the pre-images of the total v -subgraph $T(v)$ are just the total subgraphs rooted at the pre-images of the node v in T' . In [4] we suggested to consider pre-images as tree-tuples and gave natural examples for such XFDs. Given T and v , a *functional dependency* (XFD , v - XFD) is an expression $v : \mathcal{X} \rightarrow \mathcal{Y}$ where \mathcal{X} and \mathcal{Y} are non-empty sets of v -subgraphs of T . T' satisfies $v : \mathcal{X} \rightarrow \mathcal{Y}$ if and only if for any two pre-images W_1 and W_2 of $T(v)$ in T' the projections $W_1|_Y$ and $W_2|_Y$ are value-equal for all $Y \in \mathcal{Y}$ whenever the projections $W_1|_X$ and $W_2|_X$ are value-equal for all $X \in \mathcal{X}$.

For example, the data tree in Figure 1 satisfies $v_{Purchase} : \llbracket Article, Price \rrbracket \rightarrow \llbracket Discount \rrbracket$. This can be checked by inspecting the three tree-tuples, that is, the total subgraphs rooted at the three pre-images of $v_{Purchase}$. The data tree in Figure 2 satisfies the same XFD. This is noteworthy as the latter data tree does not satisfy the XFD $\llbracket Article, Price \rrbracket \rightarrow \llbracket Discount \rrbracket$ when based on almost-copies as tree-tuples. Note that the definition of v -XFDs should not be simplified to expressions $v : X \rightarrow Y$ with single v -subgraphs X and Y as this causes a loss of expressiveness, e.g., the XFDs $v_{Purchase} : \llbracket Article, Price \rrbracket \rightarrow \llbracket Discount \rrbracket$ and $v_{Purchase} : \llbracket Article \rrbracket, \llbracket Price \rrbracket \rightarrow \llbracket Discount \rrbracket$ are different from one another. In fact, the data tree in Figure 2 satisfies the former XFD, but not the latter one.

Theorem 1. *The inference rules below form a sound and complete set of inference rules for the implication of v -XFDs:*

$$\begin{array}{ccc}
 \frac{}{v : \emptyset_{T,v} \rightarrow T(v)} \text{ } v \text{ simple} & \frac{}{v : X \rightarrow Y} \text{ } Y \text{ v-subgraph of } X & \frac{}{v : X, Y \rightarrow X \sqcup Y} \text{ } X, Y \text{ reconcilable} \\
 \text{(uniqueness)} & \text{(subgraph)} & \text{(join)} \\
 \\
 \frac{}{v : \mathcal{X} \rightarrow \mathcal{Y}} \text{ } \mathcal{Y} \subseteq \mathcal{X} & \frac{}{v : \mathcal{X} \rightarrow \mathcal{Y}} & \frac{}{v : \mathcal{X} \rightarrow \mathcal{Y}, v : \mathcal{Y} \rightarrow \mathcal{Z}} \\
 \text{(reflexivity)} & \text{(extension)} & \text{(transitivity)}
 \end{array}$$

The uniqueness axiom states that if the path from the root to the node v in the schema tree T is *simple*, i.e., does only contain arcs of frequency ? or 1,

then a T -compatible data tree T' has at most one pre-image of $T(v)$. The join axiom gives a sufficient (and also necessary) condition when the projections of a pre-image W of $T(v)$ on two v -subgraphs X and Y uniquely determine the projection on their union $X \sqcup Y$. Two v -subgraphs X, Y are called *reconcilable* if whenever X and Y share some arc (u, w) of frequency other than ? or 1, then X contains the total w -subtree of Y or Y contains the total w -subtree of X .

In the sequel we discuss how to decide implication efficiently. Let T be a schema tree, and v a node of T . The set $\mathcal{B}(v)$ of *essential subgraphs* is defined as the smallest set of v -subgraphs of T such that every v -walk of T belongs to $\mathcal{B}(v)$ and if $X, Y \in \mathcal{B}(v)$ are not reconcilable then $X \sqcup Y \in \mathcal{B}(v)$. Note that two pre-images that coincide on the projections to all members of $\mathcal{B}(v)$ must be value-equal, and $\mathcal{B}(v)$ is the smallest set with this property. For a set \mathcal{X} of v -subgraphs of T let $\vartheta(\mathcal{X})$ contain all the essential subgraphs in $\mathcal{B}(v)$ that are subgraphs of some member of \mathcal{X} and are maximal with respect to this property, i.e., $\vartheta(\mathcal{X}) = \max\{Y \in \mathcal{B}(v) : Y \text{ is } v\text{-subgraph of } X \text{ for some } X \in \mathcal{X}\}$. A T -compatible XML data tree T' satisfies the XFD $v : \mathcal{X} \rightarrow \mathcal{Y}$ if and only if T' satisfies the XFD $v : \vartheta(\mathcal{X}) \rightarrow \vartheta(\mathcal{Y})$. We may therefore assume without loss of generality that every XFD $v : \mathcal{X} \rightarrow \mathcal{Y}$ is of the form $\mathcal{X} = \vartheta(\mathcal{X})$ and $\mathcal{Y} = \vartheta(\mathcal{Y})$.

Now we establish a correspondence between the implication of XFDs and the logical implication of propositional Horn clauses. Let $\varphi : \mathcal{B}(v) \rightarrow \mathcal{V}$ be a mapping that assigns propositional variables to the v -subgraphs of T . If σ is an XFD $v : \{X_1, \dots, X_k\} \rightarrow \{Y_1, \dots, Y_n\}$ on T , then let Π_σ be the set of the following n Horn clauses: $\neg\varphi(X_1) \vee \dots \vee \neg\varphi(X_k) \vee \varphi(Y_1), \dots, \neg\varphi(X_1) \vee \dots \vee \neg\varphi(X_k) \vee \varphi(Y_n)$. If Σ is a set of v -XFDs on T , then let Π_Σ be the union of the sets $\Pi_\sigma, \sigma \in \Sigma$. Further, the structure of $\mathcal{B}(v)$ can be encoded by the set $\Pi_T = \{\neg\varphi(U) \vee \varphi(W) : U, W \in \mathcal{B}(v), U \text{ covers } W\}$, where a v -subgraph U is said to *cover* a v -subgraph W if U is the union of W and just one additional v -walk of T .

Theorem 2. *Let $\Sigma \cup \{\sigma\}$ be a set of v -XFDs on T . Σ implies σ if and only if $\Pi_\Sigma \cup \Pi_T$ logically implies Π_σ .*

Corollary 3. *The problem whether Σ implies σ can be decided in time linear in the total number of essential subgraphs in Σ .*

The corollary follows straight from the linear time decidability for the implication of propositional Horn clauses [2]. Thus, XFDs based on pre-images do not only occur naturally in XML data but enjoy well-founded reasoning techniques that can be implemented efficiently for native XML data management. This is in contrast to many other classes of XML constraints [3].

References

1. M. Arenas, L. Libkin. A normal form for XML documents. *ACM ToDS* 29, 2004.
2. W. Dowling, J. H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Logic Programming* 1, 1984.
3. W. Fan. XML constraints. *DEXA Workshops* 2005.

4. S. Hartmann, S. Link. More functional dependencies for XML. *ADBIS 2003*, LNCS 2798.
5. S. Hartmann, T. Trinh. Axiomatising functional dependencies for XML with frequencies. *FoIKS 2006*, LNCS 3861.
6. M. Lee, T. Ling, W. Low. Designing functional dependencies for XML. *EDBT 2002*, LNCS 2287.
7. M. Nicola, B. van den Linden. Native XML support in DB2. *VLDB 2005*.
8. M. Vincent, J. Liu. Completeness and decidability properties for functional dependencies in XML. CoRR cs.DB/0301017, 2003.
9. M. Vincent, J. Liu, C. Liu. Strong functional dependencies and their application to normal forms in XML. *ACM ToDS* 29, 2004.
10. J. Wang. A comparative study of functional dependencies for XML. *APWeb 2005*, LNCS 3399.
11. J. Wang, R. Topor. Removing XML data redundancies using functional and equality-generating dependencies. *ADC 2005*, CRPIT 39.