A Survey Study on XML Functional Dependencies

Teng Lv[1,2] and Ping Yan[1]
[1]College of Mathematics and System Science, Xinjiang University, Urumqi 830046, China
[2]Teaching and research Section of Computer, Artillery Academy, Hefei 230031, China
E-mails: Lt0410@163.com, yaning@xju.edu.cn

Abstract

There are two major kinds of XML functional dependency (FD) definitions. The first kind of XML FD includes Tree-tuple-based XML FD (tFD) and Path-based XML FD (pFD), and the second kind of XML FD includes Extended-path-based XML FD (epFD), Sub-graph-based XML FD (gFD), and Generalized-tree-tuple-based XML FD (gtFD). The relationships and differences in semantic power between these functional dependencies are analyzed, and some results are obtained as follows: (1) tFD and pFD of the first kind of XML FD have the same expressive power of XML FD. (2) epFD, gFD, and gtFD of the second kind of XML FDs have the same expressive power of XML FD, too. (3) The second kind of XML FD can deal with set elements in XML FD and have more general expressive power than the first kind of XML FD.

1. Introduction

XML [1] has become one of primary standards for data exchange over Web. XML functional dependencies (FD) are very important semantic expressions in XML data, which are the foundations of other related research, such as normalizing XML documents, query optimization, and schema design [2]. XML FDs are a relatively new research topic comparing with the counterpart in relational database (RDB) world [3].

There are two major kinds of XML FDs. The first kind can capture FDs similar to relational FDs in RDB. For example, Tree-tuple-based XML FD (tFD) [4] is proposed similar to the concept of tuple of relational database, and Path-based XML FD (pFD) [5,6] uses XML path to define FDs between elements, attributes, and values of elements and attributes. The second kind can capture more semantic information than the first one, especially the FDs involving set elements of XML, i.e. an element can occur multiple times. This kind of XML FDs include Extended-path-based XML FD (epFD) [7] which extends the pFD and can deal with FDs involving set elements, Sub-graph-based XML FD (gFD) [8] which uses XML sub-graphs to define XML FDs, and Generalized-tree-tuple-based XML FD (gtFD)[9] which extends the concept of tree tuple to define XML FDs.

The major goal of this paper is to compare the above approaches to XML FDs, analyses the relationship between them, and proposes the characteristics of a good XML FD definition.

2. The first kind of XML FD: tFD and pFD

Definition 1 (Tree-tuple-based XML FDs (tFD))[4]. Given a DTD D and two non-empty sub-sets S_1 and S_2 over the path set $paths(D)$. A tFD over D has the form $S_1 \rightarrow S_2$. For an XML tree $T \models D$, for tree tuples t_1 and t_2 in the set of tree tuples $tuples(T)$, if $t_1.S_1 = t_2.S_1$ and $t_1.S_1 \neq null$ implies $t_1.S_2 = t_2.S_2$, then XML tree T satisfies tFD $S_1 \rightarrow S_2$.

Example 1. The following DTD D_1 describes the information of course, student, and teacher:

```
<!ELEMENT courses (course*)>
<!ELEMENT course (title,takenby)>
   <!ATTLIST course cno CDATA #REQUIRED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT takenby (student*)>
<!ELEMENT student (sname, teacher)>
   <!ATTLIST student sno CDATA #REQUIRED>
<!ELEMENT sname (#PCDATA)>
<!ELEMENT teacher (tname)>
   <!ATTLIST teacher tno CDATA #REQUIRED>
<!ELEMENT tname (#PCDATA)>
```

Suppose an FD such that a student's number (@sno) can uniquely determines a student node within the sub-tree rooted on a course node, which can be expressed by a tFD as: {*courses.course,ourses.course.takenby.student.@sno*} → {*courses.course.takenby.student*}.

Definition 2 (Path-based XML FD (pFD)) [6]. A pFD has the form $\{P_{l1}, P_{l2}, \ldots, P_{ln}\} \rightarrow P_{lr}$, where LHS (Left Hand Side) path set $\{P_{l1}, P_{l2}, \ldots, P_{ln}\}$ specifies the condition elements, RHS (Right Hand Side) path P_{lr} specifies the implication element.

Example 2. The above FD can be expressed by pFD as: {*courses.course , courses.course.takenby.student.@sno*} → *courses.course.takenby.student*.

Relationship between tFD and pFD.

0-7695-3016-8/07 $25.00 © 2007 IEEE
DOI 10.1109/ISDPE.2007.32

For a tFD $S_1 \rightarrow S_2$, it can be expressed as a set of n pFDs: $\{P_{l1}, P_{l2}, \ldots, P_{ln}\} \rightarrow P_{lri}$, where $S_1 = \{P_{l1}, P_{l2}, \ldots, P_{ln}\}$, $S_2 = \{P_{lri}\} (i=[1,n])$. It is obvious that two tree tuples t_1 and t_2, if $t_1.S_1 = t_2.S_2$ and $t_1.S_1 \neq null$ implies $t_1.S_2 = t_2.S_2$, then $t_1.\{P_{l1}, P_{l2}, \ldots, P_{ln}\} = t_2.\{P_{l1}, P_{l2}, \ldots, P_{ln}\}$ and $t_1.\{P_{l1}, P_{l2}, \ldots, P_{ln}\} \neq null$ implies $t_1.\{P_{lri}\} = t_2.\{P_{lri}\}$. Moreover, t_1 and t_2 can be converted to the instance sets of the pFD. Similarly, a pFD can also be expressed as a tFD. So we have:

Proposition 1. tFD and pFD have the same expressive power of XML FD.

3. The second kind of XML FD: epFD, gFD, and gtFD

Definition 3 (Extended-path-based XML FDs (epFD))[7]. Given a *DTD D*, a epFD *f* over *D* has the form $\{S_h, [S_{x1},\ldots,S_{xn}] \rightarrow [S_{y1},\ldots,S_{ym}]\}$, where (1) $S_h \in paths(D)$ is called the header path of *f* which defines the scope of *f* over *D*. And the last symbol of path S_h is an element name, i.e., $last(S_h) \in E$. If $S_h \neq$null and $S_h \neq r$, then *f* is called a local FD which means that the scope of *f* is the sub-tree rooted on $last(S_h)$; otherwise, *f* is called a global FD which means the scope of *f* is the overall *D*. (2) $[S_{x1},\ldots,S_{xn}]$ is called LHS path set of *f*. For $i=1,\ldots,n$, it is the case that $S_{xi} \in paths(D)$, $S_{xi} \supseteq_{Path} S_h$ (S_h is a prefix of S_{xi}, but not necessarily a proper prefix), $S_{xi} \neq$null, and $last(S_{xi}) \in E \cup A \cup S$. (3) $[S_{y1},\ldots,S_{ym}]$ is called RHS path set of *f*. For $j=1,\ldots,m$, it is the case that $S_{yj} \in paths(D)$, $S_{yj} \supseteq_{Path} S_h$, $S_{yj} \neq$null, and $last(S_{yj}) \in E \cup A \cup S$.

For an XML tree $T \models D$, we call *T* satisfies epFD *f* (denoted as $T \models f$) iff for any nodes $H \in [[S_h]]$ (let $H = root$ if $S_h = null$) and $X_1, X_2 \in H[[S_{x1} \cap \ldots \cap S_{xn}]]$ in *T*, if there exist nodes $X_1[[S_{x1}]] =_v X_2[[S_{x1}]], \ldots, X_1[[S_{xn}]] =_v X_2[[S_{xn}]]$, and it is the case that for any nodes $Y_1, Y_2 \in H[[S_{y1} \cap \ldots \cap S_{ym}]]$ and $H(p(X_1) \cap p(Y_1)), H(p(X_2) \cap p(Y_2)) \in H[[S_{x1} \cap \ldots \cap S_{xn} \cap S_{y1} \cap \ldots \cap S_{yn}]]$ such that $Y_1[[S_{y1}]] =_v Y_2[[S_{y1}]], \ldots, Y_1[[S_{ym}]] =_v Y_2[[S_{ym}]]$.

Example 3. The following DTD D_2 describes the information of *course*, a *pair* taking the *course*, and the *rating* of a *course* for a *pair*. Suppose for each *course*, the set of *pair*s can determine the *rating*, it can be expressed by epFD as: {*courses.course*, [*courses.course.pair*] $\rightarrow$ [*courses.course.rating*]}.

```
<!ELEMENT courses (course*)>
<!ELEMENT course (pair*,rating)>
<!ELEMENT pair (male, female)>
<!ELEMENT male (#PCDATA)>
<!ELEMENT female (#PCDATA)>
<!ELEMENT rating (#PCDATA)>
```

Definition 4(Sub-graph-based XML FDs (gFD))[8]. An XML gFD has the form $\{v: X \rightarrow Y\}$, where *v* is a node of XML tree *T*, *X* and *Y* are *v-subgraphs of T* (rooted on node *v*, denoted as $T(v)$). *T* satisfies $\{v: X \rightarrow Y\}$ iff for any two pre-images W_1 and W_2 of $T(v)$, if their projections on *X* are equal, then their projections on *Y* are equal, too.

Example 4. The FD in Example 3 can also be expressed by gFD as: $\{v_{course}: X \rightarrow Y\}$, where *X* is the v_{course}-subgraph with leave elements *male* and *female*, *Y* is the v_{course}-subgraph with leave element *rating*.

Definition 5 (Generalized-tree-tuple-based XML functional dependencies (gtFD))[9]. An XML gtFD is a triple $<C_p, LHS, RHS>$, often written as $\{P_{l1}, P_{l2}, \ldots, P_{ln}\} \rightarrow P_r$ *w.r.t.* C_p, where C_p denotes a tuple class, *LHS* is a set of paths (P_{li}, $i=[1,n]$) relative to path *p*, and *RHS* is a single path P_r relative to *p*. An XML gtFD holds on a XML tree T iff for any two *generalized tree tuples* $t_1, t_2 \in C_p$: (1) $\exists i \in [1,n]$, $t_1.P_{li} = \perp$ or $t_2.P_{li} = \perp$, or (2) if $\forall i \in [1,n]$, $t_1.P_{li} =_{pv} t_2.P_{li}$, then $t_1.P_r \neq \perp$, $t_2.P_r \neq \perp$, and $t_1.P_r =_{pv} t_2.P_r$.

Example 5. The FD in Example 3 can also be expressed by gtFD as:
{*courses.course.pair*} $\rightarrow$ *courses.course.rating w.r.t.* C_{course}.

Relationship between epFD, gFD, and gtFD.

epFD uses the extended paths to express FD involveing set elements. In fact, the tuple class in gtFD is implied in the header path S_h of epFD, so they have the same expressive power.

gtFD uses the concept of *generalized tree tuple* to define XML FDs, which can capture the set elements of XML tree. In fact, a *generalized tree tuple* is a sub-tree (or a sub-graph) of XML tree. For example, C_{course} in DTD D_2 is the set of sub-trees rooted on the 3 *course* nodes, which coincides with the *sub-graph* in gFD definitions. More specifically, for a gtFD $\{P_{l1}, P_{l2}, \ldots, P_{ln}\} \rightarrow P_r$ *w.r.t.* C_p, it can be expressed by a gFD $\{v: X \rightarrow Y\}$, where *v* is the root of path *p*, *X* is a *v-subgraph* including paths $\{P_{l1}, P_{l2}, \ldots, P_{ln}\}$, and *Y* is a *v-subgraph* including path P_r. On the other hand, a gFD $\{v: X \rightarrow Y\}$ can be expressed by a gtFD $\{P_{l1}, P_{l2}, \ldots, P_{ln}\} \rightarrow P_r$ *w.r.t.* C_p, where p is the path rooted on node *v*, the set $\{P_{l1}, P_{l2}, \ldots, P_{ln}\}$ constitutes the *v-subgraph X*, and P_r constitutes the *v-subgraph Y*.

From the above discussion, we have:

Proposition 2. epFD, gFD and gtFD have the same

expressive FD power.

4. Relationship between the first and second kind of XML FD

As the first kind of XML FD, i.e. tFD and pFD have the same expressive FD power, and the second kind of XML FDs, i.e. epFD, gFD, and gtFD have the same expressive FD power, too, so we choose pFD and gFD from the first and second kind of XML FD respectively to discuss the relationship between them.

For gFD $\{v_{course}: X \rightarrow Y\}$ in DTD D_2, where X is the v_{course}-subgraph with leave elements *male* and *female*, Y is the v_{course}-subgraph with leave element *rating*, it can not be consistently expressed by pFD as {*courses.course*, [*courses.course.pair.male*, *courses.course.pair.female*]→ [*courses.course.rating*]}, as the pFD only says that "for a specific course, elements male and female can determine the corresponding rating", which is not the exact meaning of the gFD which says that "for each course, the set of pairs can determines the rating".

As the second kind of XML FD can deal with set elements in XML FDs while the first kind of XML FD can not, so we have the following:

Proposition 3. The second kind of XML FD (epFD, gFD, and gtFD) can deal with set elements in XML FD and have more general expressive power than the first kind of XML FD (tFD and pFD).

Example 6. The tFD in Example 1 can be expressed by epFD as {*courses.course*, [*courses.course.takenby.student.@sno*] →[*courses.course.takenby.student*]}, by gFD as $\{v_{course}: X \rightarrow Y\}$, where X is the v_{course}-subgraph with leave attribute @sno, and Y is the v_{course}-subgraph with leave element *student*, and by gtFD as {*courses.course.takenby.student.@sno*}→ *courses.course.takenby.student w.r.t.* C_{course}.

5. Conclusions

It is very important to research XML FD which is fundamental to other related areas of XML theories and applications. This paper gives a detailed comparison between two major kinds of XML FD definitions. In general, the second kind of definition is a natural extension of the first kind, and can express more general XML FD, especially can deal with FD involving set elements. From the above discussion, we think a good XML FD definition should have the following characteristics: (1) it is a natural extension of existed FD definitions in XML or RDB world; (2) it can deal with complicated situations in XML data; (3) it has relative simple and intuitive meaning in formal definition; (4) it should have the consistency with XML keys.

6. Acknowledgement

This work is supported by Natural Science Foundation of Anhui Province (No.070412057), National Natural Science Foundation of China (No. 60563001), and College Science & Research Plan Project of Xinjiang (No.XJEDU2004S04).

7. References

[1] T. Bray, J. Paoli, C. M. Sperberg-McQueen, et al, "Extensible Markup Language (XML) third edition", http://www.w3.org/TR/REC-xml.

[2] T. Teng, N. Gu, and P. Yan, "Normal forms for XML documents", *Information and Software Technology*, 2004, 46(12):839-846.

[3] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of databases*, Massachusetts: Addison-Wesley, 1995.

[4] M. Arenas and L. Libkin, "A normal form for XML documents", In: Lucian Popa. *Proc. of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'02)*, New York: ACM press, 2002. 85-96.

[5] M. L. Lee, T. W. Ling, and W. L. Low, "Designing functional dependencies for XML", In: Christian S. Jensen, Keith G. Jeffery, Jaroslav Pokorný, et al, *Proc. of 8th International Conference on Extending Database Technology (EDBT'02)*, Germany: Springer, 2002, 124-141.

[6] M. Vincent, J. Liu, and C. Liu, "Strong functional dependencies and their application to normal forms in XML", *ACM Transactions on Database Systems*, 2004, 29(3): 445-462.

[7] T. Lv, P. Yan, and Z. Wang, "Functional dependencies for XML", *Mini-Macro Systems*, 2005, 26(5): pp.864-868.

[8] S. Hartmann and S. Link, "More functional dependencies for XML", *Proc. of ADBIS 2003*, LNCS 2798. Germany: Springer, 2003, pp.355-369.

[9] C. Yu and H. V. Jagadish, "Efficient discovery of XML data redundancies", *Proc. of VLDB'06*, ACM press, 2006, pp.103-114.