存在多值依赖的 XML DTD 规范化研究*)

丘 威[1] 张立臣[2]

(嘉应学院计算机系 广东 梅州 514015)[1] (广东工业大学计算机学院 广州 510090)[2]

摘 要 XML DTD文档中可能包含由非函数依赖引起的数据冗余和操作异常,首先从消除DTD文档内数据冗余的角度出发研究了文档的规范化的问题,讨论了在DTD文档中存在多值依赖的情况下,如何规范XML文档,提出了以DTD为模式的XML文档的多值依赖的概念。然后基于多值依赖的概念,提出了XML文档的一种多值依赖范式MXNF。最后在此基础上提出了把一个XML文档的DTD无损联接地分解成为符合MXNF的规范化算法,来规范存在多值依赖的XML DTD文档,并给出了该算法的分析说明。

关键词 XML DTD,规范化,多值依赖,多值XML范式(MXNF)

Study of Normalization Existing MVD in XML DTD

QIU Wei[1] ZHANG Li-Chen[2]

(Department of Computer Science and Technology, Jiayying University, Guangdong Meizhou 514015)[1]

(Faculty of Computer Science, Guangdong University of Technology, Guangzhou 510090)[2]

Abstract XML DTD documents may contain data redundancies and operation anomlies due to non-functional dependencies. First the normalization problem of XML DTD is studied, which should avoid the occurrence of redundant information in documents, discusses how to narmalize XMLdocument when existing MVD in XML DTD document, The concept of multi-valued dependency for XML documents with DTDs as their schemas is proposed in this paper. Second a XML normal form, Multivalued XML Normal Form(MXNF), is defined based on the concept of multi-valued dependency. Finally, a lossless join decomposition algorithm for transforming an XML document's DTD into MXNF is also given, and to normalizing this XML DTD document.

Keywords XML DTD, Normalization, Multi-valued dependency, Multivaluend XML normal form(MXNF)

1 引言

XML已经成为Internet上的主要数据交换标准之一。在实际应用中,DTD是XML文档使用最多和应用最成熟的模式。由于在XML DTD文档也存在函数依赖(Functional Dependency)和多值依赖(Multivalued Dependency),正如关系数据库一样,如果XML DTD设计不好,在XML文档中同样也有数据冗余和操作异常现象,从而导致更新、删除、插入等操作异常现象。所以,规范化理论也是设计XML模式和半结构化数据库的主要核心组成部分和基础理论。对于XML模式和DTD规范化设计,现在开展的研究还不多,而且才刚起步,主要工作有:Provost提出将关系数据库理论应用于XML模式规范化设计的思想[2],这一思想还没有付诸实施;Arenas和Libkin给出XML函数依赖的概念,定义XML模式的范式XNF,论述了存在函数依赖FD的XML文档中的规范化问题,并提出将一个任意的DTD转化为一个符合XNF的DTD的算法,这一算法是通过移动属性和建立新的元素类型,以实现DTD的规范化设计[1];Ling和Lee等人定义半结构化模式的范式S3-NF,NF-SS和ORA-SS,给出将一个半结构化模式转化为一个满足某个范式的XML模式的算法,这一算法是通过规范化规则对半结构化模式进行重新构造,以实现XML模式的规范化设计[3]。本文在文[1]的基础上提出了存在多值依赖MVD的XML DTD的规范化问题。研究了XML DTD文档中由多值依赖引起的数据冗余和操作异常现象,首先给出了DTD文档的多值依赖的概念。然后,为了消除DTD文档中存在的多值依赖,从而消除由多值依赖所引起的数据冗余,提出了DTD文档的一种范式MXNF,并且给出了如何把一个给定的DTD无损联接地分解成符合MXNF的规范化的算法。

2 问题的提出

这里通过一个例子来说明DTD中存在的多值依赖,以及多值依赖在XML文档中引起的数据冗余,并指出如何通过合适的DTD变换消除这种数据冗余。

例1 $DTD\ D_1(E_1,A_1,P_1,R_1,r_1)$描述的是一所院系的信息,表示系(department)、学生(student)和教师(teacher)实体之间的关系:

```
〈! ELEMENT department (teacher * )〉
    〈! ATTLIST department dname CDATA # REQUIRED〉
〈! ELEMENT student EMPTY〉
    〈! ATTLIST student sname CDATA # REQUIRED〉
〈! ELEMENT teacher (student * )〉
```

*)本文受国家自然科学基金(No. 60474072)、广东省自然科学基金(No. 04009465)、广东省高校自然科学研究项目(No. Z03024)基金资助。

丘 威 讲师,硕士,研究方向为数据库技术和软件工程;张立臣 教授,主要研究方向:实时系统和软件工程。

〈! ATTLIST *teacher tname* CDATA # REQUIRED〉

例 1 中的 DTD D1 对应的一个 XML 文档树如图 1 所示。

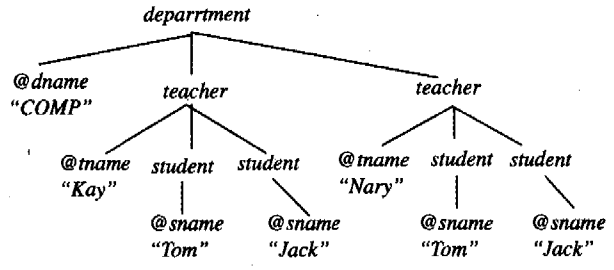

图 1 符合 D1 的一个 XML DTD 文档

该 XML DTD 表示要存储一个系(department)里有哪些教师(teacher)和哪些学生(student),该 student 的元素类型是元素类型 teacher 的子元素类型,并不表示该学生选修了该教师所担任的课程。属性 dname、tname 和 sname 分别表示该系的名称、教师姓名和学生的姓名。图 1 表示 COMP 系里有两个教师 Kay 和 Nary,有两个学生 Tom 和 Jack。显然,在 DTD 中要为每一个教师重复存储每个学生的信息而存在数据冗余,否则,如同关系数据库一样就会出现数据的不完整。另外,该 DTD 也必然存在数据的操作异常现象。例如,若要更新学生 Tom 的信息,则必须同时更新 XML 文档中在每一名老师下的所有的有关 Tom 的信息,否则,就会造成系学生信息的不一致,这引起更新异常;若要在该系中存入一名新生的信息,就必须在每一名老师下都插入该学生的信息,否则,就会造成系学生信息的不一致,这就引起插入异常;若要在该系中删除一名学生的信息,就必须在每一名老师下都删除该学生的信息,否则,同样就会造成系学生信息的不一致,这又会引起删除异常。这些异常现象正是由于 DTD D_1 在设计上存在问题。为了避免这种异常现象,需要对 DTD 模式进行规范化处理。

3 MVD 相关定义

因为在例 1 中有两个多值依赖 MVD1:department. @dname→→department. teacher. @tname 和 MVD2:department. @dname→→department. teacher. student. @sname。也正是这两个多值依赖的存在,才使上面的 DTD 文档存在诸如数据冗余等问题。该文档用文[1]中提出的算法来规范化是不可以的。因为在该文档中存在文[1]不能处理的情况,那就是这里存在 MVD。我们看到在例 1 中,一门课程可以有多个老师担任、可供多个学生选修,而老师和学生之间没有必然的关系。这就与我们在关系数据库中遇到的 MVD 相似。以此为基础,将给出在 DTD 文档中多值依赖的表达式。参考在文[3]中提出的在 XML 文档中的函数依赖的表达方式:$(Q,[Px1,\cdots,Pxn\rightarrow Py])$,其中 Q 是指函数依赖的头路径,$Px1,\cdots,Pxn$ 表示函数依赖的 Left-Hand-Side 实体类型,Py 表示函数依赖的 Right-Hand-Side 实体类型。与此定义相似,我们提出在 DTD 中的 MVD 的表示方法。DTD 文档中的 MVD 可以表示为 $(Q,[Px\rightarrow\rightarrow Py])$,其中 Q、Px、Py 的意义分别为 MVD 的头路径,MVD 的 Left-Hand-Side 和 MVD 的 Right-Hand-Side。但是,在这里 Px、Py 都是元素节点或属性节点的集合。

定义 1 给定 DTD D,$X\cup Y\in paths(tree(X))$,$Z=paths(tree(X))-X\cup Y$,$x,y,z$ 分别表示元组在路径 X,Y,Z 上的值[4]。对于任意的 XML 树 $T\models D$,如果在 T 中只要存在元组 $t1$ 和 $t2$,并且满足如下的关系:$t1(X)=t2(X)=x$,$t1(Y)=y1$,$t1(Z)=z1$,$t2(Y)=y2$ 和 $t2(Z)=z2$,那么在 T 中也存在元组 $t3$ 和 $t4$,并且满足如下的关系:$t3(X)=t4(X)=x$,$t3(Y)=y1$,$t3(Z)=z2$,$t4(Y)=y2$ 和 $t4(Z)=z1$,那么就称多值依赖(简记为 MVD)$X\rightarrow\rightarrow Y$ 在 D 上成立。D 上所有的多值依赖的集合表示成 $MVD(D)$。假定 $\Sigma\subseteq MVD(D)$,$T\models D$ 表示如果对每一个 $MVD\varphi\in\Sigma$,都有 $T\models\varphi$。$T\models(D,\Sigma)$ 当且仅当 $T\models\Sigma$ 和 $T\models D$。$(D,\Sigma)\vdash\varphi$ 表示 (D,Σ) 隐含 $MVD\varphi$。由 (D,Σ) 隐含的所有 MVD 的集合表示为 $(D,\Sigma)^+$。

定义 2 给定 DTD D 和 $paths(tree(X))$ 上成立的一个多值依赖 $X\rightarrow\rightarrow Y$,其中 $X\cup Y\subseteq paths(tree(X))$ 如果 $Y\subseteq X$ 或者 $X\cup Y=paths(tree(X))$,则称多值依赖 $X\rightarrow\rightarrow Y$ 为平凡的多值依赖。实际上,所有的平凡的多值依赖都可以根据定义 1 直接推出。这里只考虑非平凡的多值依赖,如果无特别说明,这里所说的多值依赖均指非平凡的多值依赖[5]。

定义 3 对于给定的 XML 树 $T=(V,lab,ele,att,val,root)$,令 $V1=ele(lat(root))$ 是根结点的子元素的集合,称为第一层元素结点,Ve 是叶子结点的集合,也就是如果 $v\in Ve$,那么 $ele(v)=\Phi$ 或者 v 是属性。给定非递归的 DTD D 和 XML 树 $T\models D$,如果对每一个 $v\in Ve$,v 的值都是不可分的原子值,那么称 D 是规范化的 XML 模式。

例 1 中的 DTD 都是规范化的 XML 模式,这种类型的 DTD 也是实际中使用最多的一种 XML 的模式,这是对 DTD 进行规范化的最基本的要求,因此本文只讨论这种类型的 DTD。但是仅仅符合这种条件的 DTD 往往存在着数据冗余和操作异常,这一点正如例 1 所指出的那样,因此有必要像关系数据库那样,引入一种规范 XML DTD 的范式,以避免这种数据冗余和操作异常。给定 DTD D 和 D 上成立的函数依赖的集合 $FD(D)$,如果对在 $FD(D)$ 中的每一个非平凡的 FD $X\rightarrow Y$,都满足 X 可以惟一决定 $paths(tree(X))$,那么 D 是 XML 范式(简记为 XNF)[6]。

如果在给出的 MVD 集合 M 中存在这样的两个 MVD $X\rightarrow\rightarrow Y$ 和 $Z\rightarrow\rightarrow W$,$Z$ 相应于 $X\rightarrow\rightarrow Y$ 是可分解的(即 $X\subset Z$,$Z\cap Y\neq\emptyset$,$Z-X-Y\neq\emptyset$),并且 $W\subset Y$,则相应于 $Z\rightarrow\rightarrow W$ 的分解会产生两个包含 Z 的简单类型元素和属性集合,而后相应于 $X\rightarrow\rightarrow Y$ 进行分解,在产生的集合族中会有一个集合是另一个集合的子集。在给出的 MVD 集合中,存在这样的两个 MVD:$X\rightarrow\rightarrow Y$ 和 $Z\rightarrow\rightarrow W$,$Z$ 相应于 $X\rightarrow\rightarrow Y$ 是可分解的,并且 $W\subset Y$。

如图 1 所示,在 DTD $D1$ 树中由两个 MVD 可表示为:$\Sigma=\{$(department. @dname→→department. teacher. @tname);(department. @dname→→department. teacher. student. @sname)$\}$,虽然 DTD $D1$ 存在数据冗余,但是它在一定程度上反映了数据之间(MVD)的依赖关系。对于在 DTD $D1$ 的任一 MVD$(Q,[Px\rightarrow\rightarrow Py])$,如果存在另外一 MVD$(R,[Pz\rightarrow\rightarrow Pw])$,且有 $Px\cap Pz$,$Px\cap Pw$,$Py\cap Pz$,$Py\cap Pw$ 均为空,则说这两个 MVD 没有冲突;否则,就称这两个 MVD 存在冲突。我们在这里没有详细定义各种不同的冲突关系。根据上面的定义,可以很容易地得出该 $D1$ 树$[U,\Sigma]$的无冲突的依赖集合。首先任选一 MVD$(Q,[Px\rightarrow\rightarrow Py])$加入 Σc,然后将所有与此 MVD 有冲突的 MVD 去掉;

再从与此 MVD 无冲突的 MVD 中选择一个加入$\sum c$，然后再将剩下的 MVD，且与此 MVD 有冲突的 MVD 去掉；依次类推，直到结束。这样生成的$\sum c$就是 $D1$ 树的一个无冲突依赖集。修改后的 XML DTD 有效地消除了第一个模式存在的问题。在关系数据库中提出了第四范式(4NF)的概念，与在关系数据库中的 4NF 相适应，在 XML DTD 中提出了 MXNF (Multivalued XML Normal Form)。

定义 4 给定非递归的规范化的 DTD D、一棵 XML 树 $T \vdash D$ 及其上成立的 MVD 的集合 MVD(D)，如果对在 MVD(D)中的每一个非平凡的 MVD $X \rightarrow\rightarrow Y$，都满足元组在 X 上的值可以唯一决定元组在 $paths(tree(X))$ 上的值(即元组在其中每一条路径上的值不重复存储)，那么 D 是 MXML 范式(记为 MXNF)[7]。

对于给定的 XML DTD，可以根据文[1]的方法，对其规范化使其首先满足 XNF 定义。当文档中存在 FD 时，满足 XNF 的文档已最大程度消除了数据冗余，但当文档中存在 MVD 时，满足 XNF 的文档仍然可能存在数据冗余。如对于图 1，无法用文[1]提出的算法来规范，尽管很容易地发现在文档中存在数据冗余。这里给出如何把一个 DTD 无损联接地分解成符合 MXNF。在该算法中用到一个变换规则：

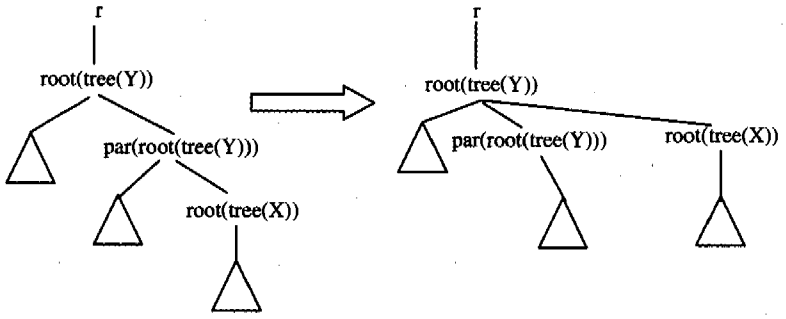

图 2 变换规则示意图

对于非平凡的 MVD $\varphi: X \rightarrow\rightarrow Y$，如果 $root(tree(X)) \neq par(root(tree(Y))$，如图 2 所示，图中的三角形表示 XML 树中的其他部分。

令 $e_{move} = root(tree(Y))$，上移以 e_{move} 为根的子树，使其成为 $root(tree(X))$的子节点。也就是令 $D' = (E, A, P', R, r)$，其中 $P'(par(e_{move})) = P(par(e_{move})) \setminus e_{move}$，$P'(root(tree(X))) = P(root(tree(X)))$，$e * _{move}$。

通过该规则的变换，使此 XML DTD 满足了 MXNL，并很好地表达了其中存在 MVD。基于这种变换规则，提出了一个通用算法来转化任一满足 XNF 到 MXNF 的 DTD 无损联接分解算法。

4 存在 MVD 的 DTD 规范化算法

DTD 的无损联接分解可以用它相应的关系表示形式的无损联接分解来加以简化，也就是认为 DTD 相应的关系表示形式的无损联接分解和它本身的无损联接分解是等价的。有如下定义：

定义 5 给定 DTD $D=(E, A, P, R, r)$及其相应的关系表示形式 R_D。$D'=(E', A', P', R', r)$是从 D 构造得到的，它相应的关系表示形式是$(R1, \cdots, Rn)$。称 D'是对 D 的无损联接分解，则$(R1, \cdots, Rn)$是对 R_D的无损联接分解[8]。

通过上面的有关 MVD 的定义和存在 MVD 的 XML DTD 变换规则，这里提出 DTD 无损联接地分解成 MXNF 的算法：

输入：DTD $D=(E, A, P, R, r)$和$\sum \subseteq MVD(D)$，D 是 XNF。

输出：对于 D 的无损联接分解的 DTD D，满足 D'是 MXNF。

方法：重复运用变换规则进行分解，直到变换后的 D'是 MXNF 为止。

步骤 1(初始化)：$D'=D$；

步骤 2(判断是否终止)：如果 D 是 MXNF，那么转到步骤 5；

步骤 3(运用变换规则)：肯定存在非平凡的 MVD $\varphi: X \rightarrow\rightarrow Y$。如果 $root(tree(X)) \neq par(root(tree(Y))$，那么运用变换规则，令 $e_{move} = root(tree(Y))$，上移以 e_{move} 为根的子树，使其成为 $root(tree(X))$的子节点；否则，转步骤 2；

步骤 4：(整理变换结果)：删除冗余元素；

步骤 5：(算法结束)：输出 D'。

算法分析：

(1) 算法在有限的时间内终止。该算法重复运用变换规对 D 进行变换，直到 D 为 MXNF 为止。由于这一变换每次都会减少 D 中 MVD 的个数，而且 MVD 的数目是有限的，因此，算法可以在有限的时间内终止。

(2)算法的输出 D'是对输入 D 的无损联接分解，且符合 MXNF。假定存在 D 的相应关系表示形式 $R(last(X), last(Y), Z)$，其中 Z 表示其它属性的集合。由算法中的变换规则可知，最后 D'的相应关系表示形式肯定包括 $R1(last(X), last(Y))$和 $R2(last(X), last(Y), Z)$。易知，$R1$ 和 $R2$ 是对 R 的无损联接分解，并且 D'是符合 MXNF。因此算法的输出 D'是对输入 D 的无损联接分解，且符合 MXNF。

(3)算法的第三步，检测条件 $root(tree(X)) \neq par(root(tree(Y))$是否成立。因为只有满足该条件，才有必要把以 $root(tree(Y))$为根的子树上移，并且把节点 $root(tree(Y))$作为 $root(tree(X))$的子节点。如在例 1 中 D2 是 MXNF 并且是对 D1 的无损联接分解。D1 中存在两个非平凡的 MVD：MVD1：department. @ dname →→ department. teacher. @ tname 和 MVD2：department. @dname→→department. teacher. sudent. @sname。运用变换规则，首先考虑 MVD1。此时元素类型 teacher 已经是元素类型 department 的子元素类型，因此，没有必要上移以 teacher 为根的子树。在考虑 MVD2，此时元素类型 student 不是元素类型 department 的子元素类型。因此，要上移以 student 为根的子树，使得 student 成为元素类型 department 的子元素类型。最后的变换结果为 DTD D2，如图 3 所示。由该算法的分析可知这一变换正是对 D1 的无损联接分解。

(下转第 185 页)

25 裴柄镇,陈晓明,胡褶,等.一种建立中文概念分类关系的新算法.计算机工程与应用,2004,36:18~21
26 Wang B B, Mckay R I(Bob), Abbass H A, et al. Learning text classifier using the dom-ain concept hierarchy. IEEE, 2002. 1230~1234
27 Zheng De-Quan, Zhao Tie-Jun, Yu Fe-ng, et al. Machine learning for automatic acquisition of Chinese lingu-istic ontology knowledge. IEEE, 2005. 3728~3733
28 Shauntrelle D D, Tia B W. Engi-neering knowledge. In: Proceedings of the 42nd Annual Southeast Regional Co-nference, Huntsville, Alabama, 2004. 406~407
29 Suryanto H, Compton P. Learning classification taxonomies from a classi-fication knowledge based system. In: Proceedings of the ECAI 2000 Works-hop on Ontology Learning (OL'2000), 2000
30 Bisson G, Nedellec C, Canamero D. Designing clustering methods for ontology building: The Mo'K workbe-nch. In: Proceedings of the ECAI 2000 Workshop on Ontology Learning (OL'2000), 2000
31 Lenat G. Building large knowledgebased system: representation and infer-ence in the CYC project. 1st edition. Boston: Addison Wesley Press, 1989
32 Snasel V, Moravec P, Pokorny J. WordNet ontology based model forweb retrieval. IEEE, 2005. 220~225
33 Gruninger M, Fox M. Methodologyfor the design and evaluation of ontol-ogies. In: Proceedings of the IJCAI 95 Workshop on Basic Ontological Iss-ues in Knowledge Sharing, 1995
34 Emde W, Wettschereck D. Relatio-nal instance based learning. In: Proc-eedings of 13th International Confere-nceon Machine Learning (ICML'96), 1996. 122~130
35 Yamaguchi T. Acquiring conceptual relations from domain-specific texts. In: Proceedings of the IJCAI 2001 W-orkshop on Ontology Learning, 2001
36 Maedche A, Staab S. Ontology le-arning. In: Proceedings of 14th Euro-pean Conference on Artificial Intellig-ence, 2000
37 Craven M, DiPasquo D, Freitag D, et al. Learning to constru-ct knowledge bases from the World Wide Web. Artificial Intelligence, 2000. 69~113
38 Maedche A, Volz R. The Text-To-Onto ontology extraction and mainten-ance environment. In: Proceedings of the ICDM Workshop on Integrating Data Mining and Knowledge Manage-ment, California, 2001
39 Wu S H, Hsu W L. SOAT: A semi-automatic domain ontology acqui-sition tool from Chinese corpus. In: the 19th International Conference on Computational Linguistics, Howard In-ternational House and Academia Sinic-a, Taipei, Taiwan, 2002
40 Chaelandar G, Grau B. SVETLAN'a system to classify words in context. In: Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI'00, Berlin, Germany, 2000

(上接第 151 页)

(4) 算法的第四步是删除冗余的元素。这是因为在运用变换规则的时候,有可能使得某些元素既没有属性也没有任何元素,而且也不可以取字符串值。此时,可以根据实际应用的需求删除这些冗余的元素。

例 2 为了避免 DTD *D*1 中存在多值依赖的数据冗余,可以把 *D*1 通过变换规则和 MVD 无损联接分解算法,成为如下 DTD *D*2:

〈! ELEMENTdepartment (teacher * ,student *)〉
〈! ATTLIST department dname CDATA # REQUIRED〉
〈! ELEMENT student EMPTY〉
〈! ATTLIST student sname CDATA # REQUIRED〉
〈! ELEMENT teacher EMPTY 〉
〈! ATTLIST teacher tname CDATA # REQUIRED〉

例 2 中的 DTD *D*2 对应的一个 XML 文档树,如图 3 所示。

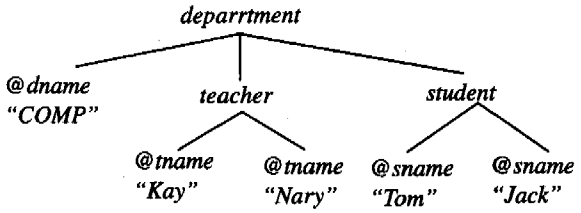

图 3 符合 D2 的一个 XML DTD 文档

通过这种变换,把直接联系的实体放在一起形成嵌套关系,在此处是把教师和学生都作为系的子元素类型,从而消除了 *D*1 中存在的数据冗余和操作异常。

通过此算法,我们可以获得满足 MXNF 的 DTD 文档。这样的文档很好地消除了当存在 MVD 时的数据冗余,而且文档有较好的结构。但是,通过这一算法得到的 DTD 并不一定保持依赖,这与关系数据库中的 4NF 的分解很相似,这里不再对此详细实证[9]。

结论与进一步的工作 本文研究了 XML 文档中的多值依赖问题,讨论了当 XML DTD 中存在 MVD 时的规范化问题,分析了 XML 文档中由多值依赖引起的数据冗余和各种操作异常现象。提出了 DTD 的多值依赖的概念,定义了 XML DTD 的一种范式 MXNF,给出了把 DTD 无损联接地分解成符合 MXNF 的算法,它不仅消除了 XML DTD 文档中的数据冗余和各种异常,而且更好表达了现实世界中实体的语义关系。此外,对于 XML DTD 相关的多值依赖的变换规则以及如何充分发挥 XML DTD 的特性来更好地描述数据,提出 MXNF 规范化算法。这将对未来的 XML 函数依赖保持、XML 完整性约束、推理规则、XML 多值依赖以及 XML 模式的进一步规范化研究奠定理论基础[10]。

参 考 文 献

1 Arenas M, Libkin L. A Normal Form for XML Documents. In: Symposium on Principles of Database Systems (PODS'02), Madison, Wisconsin, U. S. A. ACM Press, 2002. 85~96
2 Provost W. Normalizing XML. http:// www. xml. com/pub/a/2002/11/13/normalizing. html
3 Lee Mong Li, Ling Tok Wang, Low Wai Lup. Designing Functional Dependencies for XML. In: VIII Conference on Extending Database Technology (EDBT), Prague, March 2002
4 谈子敬,施伯乐. DTD 的规范化. 计算机研究与发展,2004,41(4):594~601
5 Vianu V. A Web Odyssey: from Codd to XML. In: Proceedings of ACM PODS, Santa Barbara CA USA, 2001. 148~160
6 张忠平,王超,朱扬勇. 基于约束的 XML 文档规范化算法. 计算机研究与发展,2005,42(5):755~764
7 Tan Zi-Jing, Shi Bo-Le. Propagating Functional Dependency and Normalization Between Relations and XML. Journal of Software, 2005,16(4):533~539
8 吕腾,顾宁,闫萍等. XML 文档的范式. 小型微型计算机系统[J], 2004,10(25):1836~1840
9 Fan W, Libkin L. On XML Integerity constraints in the presence of DTDs. In: Proceedings of ACM Symposium on Principles of Database Systems (PODS), Santa Barbara, California, May 2001. 114~125
10 Fan W, Simen J. Integerity constraints for XML. In: Proceedings of ACM Symposium on Principles of Database Systems(PODS), Dallas, Texas, May 2000. 23~34