函数依赖和规范化在关系和 XML 间的传播*

谈子敬+， 施伯乐

(复旦大学 计算机与信息技术系,上海 200433)

Propagating Functional Dependency and Normalization Between Relations and XML

TAN Zi-Jing+, SHI Bai-Le

(Department of Computing and Information Technology, Fudan University, Shanghai 200433, China)

+ Corresponding author: Phn: +86-21-55664603, E-mail: zjtan@fudan.edu.cn, http://www.cit.fudan.edu.cn

Received 2004-07-09; Accepted 2004-09-08

Tan ZJ, Shi BL. Propagating functional dependency and normalization between relations and XML. *Journal of Software*, 2005,16(4):533−539. DOI: 10.1360/jos160533

Abstract: To combine XML with relations is a hotspot in research field. This paper studies the functional dependency and normalization propagation between relations and XML. First the paper gives the definition of functional dependencies and keys for XML; based on it, the concepts of redundancy and DTD normalization are defined. The paper then discusses the functional dependency propagation between relations and XML. When using a general mapping from relational schema to DTD, the paper shows that all the relational functional dependencies can be preserved in the DTD; and when applying a commonly used method to mapping DTD to relational schema, each functional dependency on relations has a corresponding one in the original DTD. The significance of functional dependency propagation lies in the normalization propagation. The paper proves that using the methods above, if the original relation is in BCNF, the generated DTD is normalized, and if the original DTD is normalized, the generated relations are in BCNF.

Key words: XML; relation; propagation; functional dependency; normalization

摘 要: XML 和关系的结合是一个重要的研究领域,讨论函数依赖和规范化在关系及 XML 间的传播问题.首先引入 XML 上函数依赖和键的定义,并进一步定义 XML 上的数据冗余和规范化 DTD 的概念.分别讨论在关系和 XML 相互转化的过程中,函数依赖的传播问题.针对一种一般化的关系模式 DTD 表示,证明原有关系中的函数依赖可以在生成的 XML 文档上得到表示.针对一种常见的 XML 关系存储方法,说明最终生成关系上的函数依赖与原有 XML 上函数依赖的对应关系.函数依赖传播的核心意义在于规范化的传播.证明使用上述方法时,若原有的关系是满足 BCNF 的,则发布得到的 DTD 也是规范化的;若原始的 DTD 是规范化的,则得到的关系存储也满足 BCNF 范式.

关键词: 可扩展标记语言;关系;传播;函数依赖;规范化

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2002AA4Z3430 (国家高技术研究发展计划(863))

作者简介: 谈子敬(1975－),男,上海人,博士,讲师,主要研究领域为 XML,数据库理论;施伯乐(1935－),男,教授,博士生导师,主要研究领域为数据库与知识库.

中图法分类号: TP311 文献标识码: A

在 XML 的相关研究中,一个很重要的领域是与传统数据库的交互.一方面,目前绝大多数的数据都存放在关系数据库中.为了将这部分数据在网络上灵活、自由地交换和共享,关系数据的 XML 发布引起了相当多的关注.另一方面,随着 XML 文档数目的增加,对相应的存储、查询、索引等技术提出了更高的要求.人们很自然地希望可以为 XML 生成相应的关系模式,将文档存储到数据库,再进一步地将对 XML 的查询转化为对数据库的查询.在这个双向过程中,已有的工作更多地把重点放在结构的维护和保持上,而对于相关约束的传播则考虑得不够.这主要是由于 XML 作为一种半结构化的形式,其自身所含的数据约束信息是有限的.但是约束的信息,比如键、函数依赖等,作为数据语义的组成部分,对于数据完整性、查询优化和数据集成等都具有非常重要的作用.因而无论是在进行关系数据的 XML 发布,或是 XML 数据的关系存储时,相应约束的传播都是一个很重要的问题.约束,特别是函数依赖的传播,其核心意义在于规范化的传播.本文将进一步讨论规范化在关系及 XML 间的传播问题.

文献[1,2]在 DTD 的基础上,扩展了键的定义能力,可以表示相对键,并讨论了键的逻辑蕴涵问题.文献[3]给出一种非形式化的 XML 上的函数依赖定义,但它不能表示相对约束.文献[4]通过 XML 文档向关系元组的映射讨论 XML 的范式,给出一个将 DTD 转化为符合 XNF 形式的算法.文献[5]提出的函数依赖定义方法,将键表示为它的特例,并进一步地定义了数据冗余和规范化的概念.本文基于文献[5]中的定义来进行函数依赖和规范化的传播研究.

对关系数据的 XML 发布典型研究如文献[6]等,它们考虑了关系的结构信息,而没有涉及函数依赖等语义的传播.使用关系数据库来存储 XML 文档的研究包括文献[7,8],它们关注了文档的结构信息,而没有考虑到文档所具有的语义信息.其中,文献[7]给出一个常见和质朴的 XML 关系存储方法,它在只考虑结构信息的情况下将 DTD 映射到关系模式,由于其相当直观和一般化,因而被作为很多其他相关讨论(如文献[9,10]等)的基础. 文献[9]在文献[7]的基础上,利用 XML 的键定义来辅助关系生成,其键定义方法出自文献[1].文献[10]提出一种在进行 XML 文档的关系存储时,同时映射函数依赖的方法,其函数依赖的定义源自文献[3].如前所述,这种函数依赖定义方法不能表示 XML 上常见的相对约束.文献[11]提出一种基于约束、可以减少冗余的 XML 文档关系存储方法,其最终生成的关系模式满足 3NF.以上这些研究的重点是在给定 DTD 定义及相关约束后,对文献[7]的方法作修改,以得到保持约束,且形式较好的关系表示(最多是 3NF).而本文直接使用文献[7]中给出的 DTD 向关系模式映射的质朴方法,考察在这种常见的映射下函数依赖和规范化的传播问题.具体来说,给定 DTD 和 DTD 上的函数依赖集,使用文献[7]的方法得到相应的关系模式集 $G(G_1,G_2,...,G_n)$,我们要考察 G_i 上函数依赖的成立情况,同时要研究 G_i 的规范化问题.本文讨论了 G_i 上的函数依赖与原始 DTD 上函数依赖的对应情况;进一步地,如果原始的 DTD 是规范化的,则证明直接使用该方法得到的关系模式集 G 中的每个 G_i 都是满足 BCNF 的.

文献[12]讨论了 XML 关系存储时键传播的问题,其中键的定义方法出自文献[1].这里的键只是本文所讨论函数依赖的一个特例,同时,由于单纯键的信息不足以定义数据冗余和规范化,文献[12]没有对规范化的传播进行讨论.

1 符号说明

1.1 DTD和XML文档的树模型

定义 **1**. DTD 是一个五元组(E,A,M,N,r)[1]:

(1) E 是有限的元素类型集合;(2) A 是有限的属性类型集合,属性和元素不同名;存在一个特殊的属性 $id\in A$;(3) 对 $\forall e\in E$,$M(e)$称为 e 的元素类型定义.$M(e)=S$(字符串),或者是一个正规表达式 $\alpha ::= \varepsilon \mid e' \mid \alpha , \alpha \mid \alpha^*$.其中 ε 表示空,$e'\in E$,“,”和“*”分别代表连接和闭包;(4) 对 $\forall e\in E$,$N(e)$称为 e 的属性类型定义,$N(e)\subseteq A$.对于 $\forall e\in E$,$id\in N(e)$;(5) $r\in E$,称为根元素类型.若 $e,e'\in E$,e'属于 $M(e)$所定义的字母表,且在 $M(e)$中不包含 e'的自连接,则称 e'相对于 e 是唯一的.

定义 **2**. 符合给定 DTD(E,A,M,N,r)的 XML 文档的树模型.

(1) V 表示有限的节点集,S 表示字符串;(2) 对每一个节点 v,定义函数 $name(v)\in E\cup A$.根据 $name(v)$,进一步定义两个 V 的子集:$V_e=\{v|v\in V,name(v)\in E\}$;$V_a=\{v|v\in V,name(v)\in A\}$;(3) 对 V_e 中的节点,定义函数 $subelem(v)$ 是列表$[v_1,v_2,\ldots,v_n](v_i\in V_e)$,若 $name(v)=e,M(e)=\alpha$,则 $name(v_1),name(v_2),\ldots,name(v_n)$属于 α 定义的正规集;(4) 对于 V_e 中的节点,定义函数 $attr(v)$是集合$\{v_1,v_2,\ldots,v_m\}(v_i\in V_a)$.对任意 $a\in N(name(v))$,存在唯一的 $i\in[1,m]$, $name(v_i)=a$;(5) 对每一个节点 v,定义函数 $\mathrm{value}(v)\in\{S\}$.若 $name(v_1)=name(v_2)=id$,则 $value(v_1)\neq\mathrm{value}(v_2)$;(6) 有且仅有一个特殊的节点,记为根节点 $root$,$name(root)=r$.

定义 **3**. 节点值相等和节点相等(重合).

对节点 v 和 v',当 $name(v)=name(v'),value(v)=value(v')$时,称节点 v 和 v'值相等,记为 $v\equiv v'$.

当节点 v 和 v'表示的是树上的同一个节点时,称 v 和 v'相等,记为 $v=v'$.

1.2 路径表达式

定义 **4**. D 上的路径表达式 P 为$\rho_1/\rho_2/\ldots/\rho_n$ 的形式.其中$\rho_1,\ldots,\rho_{n-1}\in E,\rho_n\in E\cup A$.对$\forall i\in[2,n-1],\rho_i\in M(\rho_{i-1})$的字母表;$\rho_n\in M(\rho_{n-1})$的字母表$\cup N(\rho_{n-1})$.作为一种特例,使用符号 ε 来表示空路径表达式.

定义 **5**. 使用 $first(P)$和 $last(P)$分别表示路径表达式 P 的第 1 个和最后一个元素或属性类型.若 $first(P)\in M(r)$的字母表$\cup N(r)$,称 P 为根路径表达式.

定义 **6**. P,Q 都是 D 上的路径表达式,P 为$\rho_1/\rho_2/\ldots/\rho_n$,$Q$ 为$\rho_1'/\rho_2'/\ldots/\rho_m'$,则当$\rho_1/\in M(\rho_n)$的字母表$\cup N(\rho_n)$时,$\rho_1/\ldots/\rho_n/\rho_1'/\ldots/\rho_m'$也是路径表达式,称为 P 和 Q 的连接,记为 P/Q.若 $R=P/Q$,则称 R 包含 P,P 为 R 的前缀.对路径表达式 P 和 Q,若 R 既是 P 的前缀,也是 Q 的前缀,称 R 为 P 和 Q 的公共前缀.若对于任意 P 和 Q 的公共前缀 R',有 R'是 R 的前缀,则称 R 是 P 和 Q 的最大公共前缀.若 P 和 Q 的最大公共前缀为 ε,则称 P 和 Q 没有公共前缀.

定义 **7**. 路径的节点集.设有 D 上的路径表达式 $P=\rho_1/\rho_2/\ldots/\rho_n$ 和符合 D 的 XML 树上的元素节点 v,当$\rho_1\in M(name(v))$的字母表$\cup N(name(v))$时,使用记号$\langle v\{P\}\rangle$来表示路径的节点集.其意义为:若树上有节点序列$(v_0,v_1,\ldots,v_n)$,(v_i 是 v_{i-1} 的子节点,$v_0=v$),且 $\mathrm{name}(v_i)=\rho_i(i\in[1,n])$,则 $v_n\in\langle v\{P\}\rangle$.特别地,当$\langle v\{P\}\rangle$只包含单个节点时,使用 $v\{P\}$来表示这一节点.将$\langle root\{P\}\rangle$简写为$\langle P\rangle$,并规定$\langle v\{\varepsilon\}\rangle=\{v\}$,$\langle\varepsilon\rangle=\{root\}$.

定义 **8**. 设 $R=P/Q$,其中 $P=\rho_1/\rho_2\ldots/\rho_n,Q=\rho_1'/\rho_2'/\ldots/\rho_m'$.当$\rho_i'$相对于$\rho'_{i-1}$是唯一的$(i\in[2,m])$,且$\rho'_1$相对于$\rho_n$也唯一时,称 P 唯一决定 Q.

2 XML 上的函数依赖和规范化

文献[5]中给出了 XML 上函数依赖、键和规范化等的定义,本节引入这些概念,这是后面进行函数依赖和规范化传播讨论的基础.

2.1 XML上的函数依赖和键

定义 **9**. XML 上的函数依赖.

给定 $D=(E,A,M,N,r)$,D 上的函数依赖 σ 形为 $R_1,R_2(Q_1,Q_2,\ldots,Q_n\to P_1,P_2,\ldots,P_k)$,其中 R_1 为根路径表达式或 $R_1=\varepsilon$,$R_1/R_2/Q_1,\ldots,R_1/R_2/Q_n,R_1/R_2/P_1,\ldots,R_1/R_2/P_k$ 都是 D 上的根路径表达式.若 $R_1=\varepsilon$,则称该函数依赖为绝对函数依赖,否则称其为相对函数依赖.设 S 为 $Q_1,\ldots,Q_n,P_1,\ldots,P_k$ 的最大公共前缀,令 $Q_i=S/Q_i'$,$P_j=S/P_j'$,则 S 唯一决定 Q_i'和 P_j'.若 $S=\varepsilon$,则 R_2 唯一决定 Q_i 和 $P_j(i\in[1,n],j\in[1,k])$.

若符合 D 的 XML 树 X 满足下面条件:

$\forall v\in\langle R_1\rangle,\forall v_1,v_2\in\langle v\{R_2\}\rangle,\forall u_1\in\langle v_1\{S\}\rangle,\forall u_2\in\langle v_2\{S\}\rangle$,如果有 $u_1\{Q_i'\}\equiv u_2\{Q_i'\}$对于 $i\in[1,n]$都成立,则必有 $u_1\{P_j'\}\equiv u_2\{P_j'\}$ 对于 $j\in[1,k]$ 都成立,就称 $R_1,R_2(Q_1,Q_2,\ldots,Q_n\to P_1,P_2,\ldots,P_k)$ 在 X 上成立,记为 $X\models R_1,R_2(Q_1,Q_2,\ldots,Q_n\to P_1,P_2,\ldots,P_k)$.若 σ 在任意符合 D 的 XML 文档上都成立,则称 σ 在 D 上成立.

定义 **10**. 逻辑蕴涵.

给定 $D=(E,A,M,N,r)$和 D 上的函数依赖集Σ,若任意使Σ成立的符合 D 的 XML 文档 X,必然使函数依赖 σ 成立,则称Σ逻辑蕴涵 σ,记为$\Sigma\Rightarrow\sigma$.

定义 11. 被Σ逻辑蕴涵的函数依赖全体构成的集合,称为Σ的闭包,记为Σ^+.$\Sigma^+=\{\sigma \mid \Sigma \Rightarrow \sigma\}$.

定义 12. XML 上的键.

给定 D 和 D 上成立的函数依赖集Σ,若 $R_1,R_2(Q_1,Q_2,\ldots,Q_n \rightarrow id) \in \Sigma^+$,则称 $R_1,R_2(Q_1,Q_2,\ldots,Q_n)$是 D 上的键.若 $R_1=\varepsilon$,则称为绝对键;否则称其为相对键.

定理 1. 以下的一些结论成立:

1) 设 S 为 $Q_1,\ldots,Q_n,P_1,\ldots,P_k$ 的最大公共前缀,若 R_1 唯一决定 R_2/S,则 $R_1,R_2(Q_1,\ldots,Q_n \rightarrow P_1,\ldots,P_k) \in \Sigma^+$.特别地,若 $S=\varepsilon$,且 R_1 唯一决定 $R_2,R_1,R_2(Q_1,\ldots,Q_n \rightarrow P_1,\ldots,P_k) \in \Sigma^+$.

2) 若 $R_1,R_2(Q_1,\ldots,Q_n \rightarrow P_1,\ldots,P_L) \in \Sigma^+$,且 $R_1,R_2(P_1,\ldots,P_L \rightarrow t_1,\ldots,t_m) \in \Sigma^+$,则 $R_1,R_2(Q_1,\ldots,Q_n \rightarrow t_1,\ldots,t_m) \in \Sigma^+$.

3) 若 $R_1,R_2(Q_1,\ldots,Q_n \rightarrow P_1,\ldots,P_k) \in \Sigma^+$,且 $R_2=R_2'/R'$,则有 $R_1,R_2'(R'/Q_1,\ldots,R'/Q_n \rightarrow R'/P_1,\ldots,R'/P_k) \in \Sigma^+$.

4) 若 $R_1,R_2'(R'/Q_1,\ldots,R'/Q_n \rightarrow R'/P_1,\ldots,R'/P_k) \in \Sigma^+$.设 $R_2=R_2'/R'$,则有 $R_1,R_2(Q_1,\ldots,Q_n \rightarrow P_1,\ldots,P_k) \in \Sigma^+$.

5) $R_1,R_2(Q/id \rightarrow P_1,\ldots,P_k) \in \Sigma^+$.

定义 13. 平凡的函数依赖.

给定 D 和 D 上的函数依赖 $R_1,R_2(Q_1,Q_2,\ldots,Q_n \rightarrow P_1,P_2,\ldots,P_k)$,若对 $\forall j \in [1,k]$,都 $\exists i \in [1,n]$,使 $P_j=Q_i$,则称该函数依赖是平凡的.

2.2 冗余和规范化

定义 14. XML 文档中的数据冗余.

给定 D,D 上成立的函数依赖集Σ,符合 D 的文档 X 和非平凡的函数依赖 $R_1,R_2(Q_1,Q_2,\ldots,Q_n \rightarrow P_1,P_2,\ldots,P_k) \in \Sigma^+$.设 S 为 $Q_1,\ldots,Q_n,P_1,\ldots,P_k$ 的最大公共前缀,令 $Q_i=S/Q_i'$,$P_j=S/P_j'$$(i \in [1,n], j \in [1,k])$.若$\exists v \in \langle R_1 \rangle$,$\exists v_1,v_2 \in \langle v\{R_2\}\rangle$,$\exists u_1 \in \langle v_1\{S\}\rangle$,$\exists u_2 \in \langle v_2\{S\}\rangle (u_1 \neq u_2)$,有 $u_1\{Q_i'\} \equiv u_2\{Q_i'\}$对于 $i \in [1,n]$都成立,则根据函数依赖的定义,此时必有 $u_1\{P_j'\} \equiv u_2\{P_j'\}$对于$j \in [1,k]$都成立,这就是文档 X 中的冗余.若非平凡函数依赖 σ 在文档中导致了数据冗余,则称其为异常函数依赖.

定义 15. 给定 D 和 D 上成立的函数依赖集Σ,称 D 是规范化的,当且仅当:

对任给的非平凡函数依赖 $R_1,R_2(Q_1,Q_2,\ldots,Q_n \rightarrow P_1,P_2,\ldots,P_k) \in \Sigma^+$,设 S 为 $Q_1,\ldots,Q_n,P_1,\ldots,P_k$ 的最大公共前缀,令 $Q_i=S/Q_i'$,$P_j=S/P_j'$$(i \in [1,n], j \in [1,k])$,则 $R_1,R_2/S(Q_1',Q_2',\ldots,Q_n')$是 D 上的键.

定理 2. 规范化的 DTD 消除了文档中的冗余.

3 关系的 XML 发布

在定义了 XML 上的函数依赖和规范化的 DTD 后,本节我们讨论在进行关系数据的 XML 发布时,函数依赖以及规范化的传播问题.基于一种一般化的从关系模式到 DTD 的映射方法,我们证明每个原有关系上的函数依赖都可以用 XML 上的函数依赖来表示.同时,如果原有的关系模式是满足 BCNF 范式的,则得到的 XML 文档也是消除冗余的,即保持了规范化的性质.

关系模式和相应的函数依赖可以很容易地被映射到 XML 文档上,已知关系模式 $G(A_1,\ldots,A_n)$和 G 上的函数依赖集 F,则将其映射为 DTD D 和 D 上的函数依赖集Σ.

$D=(E,A,M,N,r)$的定义如下:

(1) $E=\{db,G\}$;(2) $A=\{A_1,\ldots,A_n,id\}$;(3) $M(db)=G^*$,$M(G)=\varepsilon$;(4) $N(db)=\{id\}$,$N(G)=\{A_1,\ldots,A_n,id\}$;(5) $r=db$.

Σ为:(1) 对每个 $A_{i1},\ldots,A_{im} \rightarrow A_j \in F$,有$\varepsilon,G(A_{i1},\ldots,A_{im} \rightarrow A_j) \in \Sigma$; (2) $\varepsilon,G(A_1,\ldots,A_n \rightarrow id) \in \Sigma$.

定理 3. 按照上面的构造方法,$A_{i1},\ldots,A_{im} \rightarrow A_j \in F^+$当且仅当 $\varepsilon,G(A_{i1},\ldots,A_{im} \rightarrow A_j) \in \Sigma^+$.

定理 4. G 满足 BCNF,当且仅当 D 是规范化的.

4 XML 的关系存储

4.1 XML的关系存储

我们使用文献[7]给出的 DTD 向关系模式的映射作为关系存储的基础,这是一种常见和质朴的 XML 关系

存储方法.这种方法仅保持原有的结构信息,不考虑 XML 文档上存在的其他语义.这里,DTD 被看作是一个森林,可以有多个根,但是通过引入虚根的方法,就可以将其转化为定义 1 的形式,这不会影响讨论的结果.另外,文献[7]允许共享节点的存在,DTD 图中的节点共享总是可以被消除的.将 DTD 图中的共享部分分割开,可能会需要生成更多的关系(如果共享节点和多个父节点之间都是*连接的),但是不再要为共享节点所对应的关系增加 *parent_elm* 字段来区分父节点,并可能会产生更多的内联.这样会减少查询时连接的次数,对查询是有利的;此外,消除共享节点会令文献[7]中的方法变得更为清晰.

算法 **1**. XML 的关系存储.

1) 使用 DTD 图来表示 DTD 的结构.图中的节点包括相应 DTD 中的元素、属性和操作符,并根据需要进行引入虚根和消除共享的处理

2) 在 DTD 图中,识别出需要对应独立关系的顶节点.这样的节点满足以下任一条件:(a) 根元素类型 *r*;(b) 操作符“*”的直接子节点;(c) 相互递归的两个节点中的任意一个(若其中一个为*的子节点,则选择它)

3) 从上面识别出的顶节点 *T* 出发,内联所有从 *T* 可以到达的元素和属性,直至遇到其他顶节点

4) 为每个关系增加一个 *XID* 字段作为键,记录其对应顶节点的 *id* 值

5) 如果某个关系对应的节点有父节点,则为其增加字段 *parent_ID* 作为外键来记录父元素 *X* 的键值.如果 *X* 被内联到另一个元素 *Y*,则在字段 *parent_ID* 中记录 *Y* 的键值.

例 1:学院开设有一组课程,每个课程有其名称和开设这门课程的多个教师.在每个教师下,记录其唯一标识的教师号、姓名、课程的助教和学分.

⟨!ELEMENT college (course*)⟩

⟨!ELEMENT course (teacher*)⟩

⟨!ATTLIST course cname CDATA #REQUIRED⟩

⟨!ELEMENT teacher (tname,TA,credits)⟩

⟨!ATTLIST teacher tno CDATA #REQUIRED⟩

采用算法 1 为其生成关系.

1) 使用 DTD 图来表示 DTD 的结构,如图 1 所示.

2) 在图中识别出需要对应独立关系的顶节点,包括节点 course 和 teacher.

3) 内联节点,cname 被内联到关系 course;tno,tname,TA 和 credits 被内联到关系 teacher.

4) 为关系 course 和 teacher 增加字段 CID 和 TID 作为主键.

5) 为关系 teacher 增加字段 parent_ID 记录其父节点标识.

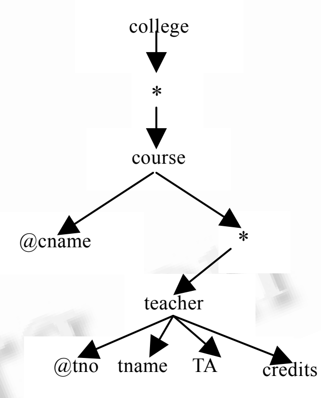

Fig.1 DTD graph

图 1 DTD 图

最后生成的关系为(关系中的键使用下划线标出):

Course(<u>CID</u>, cname);Teacher(parent_ID,<u>TID</u>,tno,tname,TA,credits).

4.2 函数依赖和规范化的传播

本文所研究的函数依赖传播问题是指,给定 DTD *D* 和 *D* 上的函数依赖集 Σ,假设使用算法 1 得到相应的关系模式集 $G(G_1,G_2,\dots,G_n)$.此时对给定 G_i 上的某个函数依赖 *F*,判定 *F* 是否在 G_i 上成立.通过分析算法 1 可知,G_i 的一般化的形式为$(pID,ID,A_1,\dots,A_n)$,其中 *ID* 对应关系所对应顶节点 v_x 的 *id* 属性,且为关系的键.*pID* 对应 v_x 父(祖先)节点 v_y 的 *id* 属性,当 v_x 没有父节点时,*pID* 不出现在关系中.$A_1,\dots,A_n$ 对应内联的节点 $v_1,\dots,v_n$ 的值,这些节点都是 v_x 的子节点(直接或间接).设在 XML 文档中,根节点 *root* 到节点 v_y 的路径为 R_1,v_y 到 v_x 的路径为 R_2,v_x 到 $v_1,\dots,v_n$ 的路径分别为 $P_1,\dots,P_n$.不失一般性,在下面的讨论中,假设 $P_1,\dots,P_n$ 没有公共前缀,参考定理 1,易知有公共前缀的情况是类似的.下面我们针对 *F* 的不同形式分别展开讨论,首先从最基本的开始,限于篇幅略去了证明.

1) $ID\to A_{i1},\dots,A_{im}$,$ID\to pID$,$pID,ID\to A_{i1},\dots,A_{im}$ 类似形式函数依赖的成立是显然的;

2) $A_{i1},\dots,A_{im}\to A_j$,它在 G_i 上成立,当且仅当 $\varepsilon,R_1/R_2(P_{i1},\dots,P_{im}\to P_j)$在 *D* 上成立;

3) $pID,A_{i1},...,A_{im} \to A_j$,它在 G_i 上成立当且仅当函数依赖 $R_1,R_2(P_{i1},...,P_{im} \to P_j)$在 D 上成立;

4) $A_{i1},...,A_{im} \to ID$ 该函数依赖成立当且仅当 XML 上 ε ,$R_1/R_2(P_{i1},...,P_{im} \to id)$成立;

5) $pID,A_{i1},...,A_{im} \to ID$ 该函数依赖成立当且仅当 XML 上 $R_1,R_2(P_{i1},...,P_{im} \to id)$成立;

6) $pID \to A_{i1},...,A_{im}$,它在 G_i 上成立,当且仅当 $R_1,R_2(empty \to P_{i1},...,P_{im})$在 D 上成立.

第 6)是一种比较特殊的情况.当 ID 相对于 pID 不唯一时,这个函数依赖实质表示的是一种非规范化,即节点 $v_{i1},...,v_{im}$ 应该作为 v_y 的子节点,而不是 v_x 的子节点.注意到在规范化的 DTD 上,$R_1,R_2(empty \to P_{i1},...,P_{im})$是不会成立的(若 R_1 不能唯一决定 R_2),也就是说,如果对规范化的 DTD 使用算法 1 来进行关系存储,仅当 pID 唯一决定 ID 时,类似于 $pID \to A_{i1},...,A_{im}$ 的函数依赖才会在关系上成立.当采用算法 1 进行 XML 文档的关系数据库存储时,每个关系上的函数依赖或者其成立与否可直接判定,或者与一个 XML 上的函数依赖相对应,这解决了函数依赖从 XML 到关系的传播问题.另一方面,注意到例 1 中的 DTD 不是规范化的,而得到的关系 Teacher(parent_ID,<u>TID</u>,tno,tname,TA,credits)也不满足 BCNF 范式.这是因为函数依赖 tno $\to$ tname 在关系上成立,而 tno 不是关系 Teacher 的键.由此出发,我们进一步考察规范化从 XML 到关系的传播.

定理 5. 使用算法 1 得到相应的关系模式集 $G(G_1,G_2,...,G_n)$.若 D 是规范化的,则 G_i 满足 BCNF.

证明:对 G_i 上的每个非平凡函数依赖 F,若 F 成立,要证明它的左边属性集能够构成关系的键.

1) $ID \to A_{i1},...,A_{im}, ID \to pID, pID, ID \to A_{i1},...,A_{im}$,显然,左边属性集是关系的键;

2) $A_{i1},...,A_{im} \to A_j, pID, A_{i1},...,A_{im} \to A_j$,若它们在 G_i 成立,由前面的讨论可知 $\varepsilon,R_1/R_2(P_{i1},...,P_{im} \to P_j)$和 $R_1,R_2(P_{i1},...,P_{im} \to P_j)$分别在 D 上成立.注意到在 D 上对应的函数依赖是非平凡的,且 D 是规范化的,则 $\varepsilon,R_1/R_2(P_{i1},...,P_{im} \to id)$和 $R_1,R_2(P_{i1},...,P_{im} \to id)$分别在 D 上成立.再由前面的讨论可知,此时有 $A_{i1},...,A_{im} \to ID$ 和 $pID,A_{i1},...,A_{im} \to ID$ 分别在 G_i 上成立,则左边属性集构成关系的键;

3) $A_{i1},...,A_{im} \to ID, pID,A_{i1},...,A_{im} \to ID$ 显然此时分别有 $A_{i1},...,A_{im}$ 或 $pID,A_{i1},...,A_{im}$ 是关系的键;

4) $pID \to A_{i1},...,A_{im}$,若该函数依赖在 G_i 成立,当且仅当 $R_1,R_2(empty \to P_{i1},...,P_{im})$在 D 上成立.因为 D 是规范化的,可知 $R_1,R_2(empty \to id)$也在 D 上成立,此时必然有 R_1 惟一决定 R_2.映射到关系上也就是 pID 唯一决定 ID,即 $pID \to ID$,则 pID 是关系的键.

综上可知,G_i 满足 BCNF.

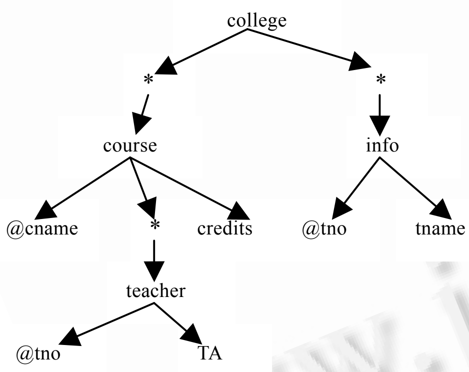

Fig.2 Normalized DTD
图 2 规范化的 DTD

例 2:针对例 1 中的 DTD,通过变换,我们给出相应的规范化 DTD 表示(如图 2 所示)和其上成立的函数依赖,使用算法 1 进行关系存储,通过函数依赖的传播,可以看到此时每个关系都满足 BCNF.

1) 课程的名称是唯一的.

ε ,course(cname $\to$ id).

2) 教师的教师号可以决定姓名.

ε ,info(tno $\to$ id).

3) 在一门课程下,教师号是唯一的.

course,teacher(tno $\to$ id).

使用单下划线标出算法 1 所生成的键,使用双下划线标出由函数依赖传播而得的键,最终得到的关系为:

Course(<u>CID</u>, <u>cname</u>, credits)

Info(<u>IID</u>, <u>tno</u>, tname)

Teacher(<u>parent_ID</u>, <u>TID</u>, <u>tno</u>, TA)

5 小 结

本文讨论了函数依赖和规范化在关系及 XML 间的传播问题.针对一种一般化的关系模式的 DTD 表示,证明原有关系中的函数依赖可以在生成的 XML 文档上表示;若原有的关系是满足 BCNF 的,则得到的 DTD 也是

规范化的.针对一种常见的 XML 的关系存储方法,说明最终生成关系上的函数依赖与原有 XML 上函数依赖的对应情况;若原始的 DTD 是规范化的,则得到的关系存储也满足 BCNF 范式.函数依赖和规范化在关系及 XML 间的传播问题有助于寻找更优的关系存储策略,并进一步地研究 XML 和 XML 的相互转化中函数依赖的传播.

References:

[1] Buneman P, Davidson SB, Fan WF, Hara CS, Tan W-C. Keys for XML. In: Proc. of the 10th Int'l World Wide Web Conf. Hong Kong: ACM Press, 2001. 201−210.

[2] Buneman P, Davidson SB, Fan WF, Hara CS, Tan W-C. Reasoning about keys for XML. In: Ghelli G, Grahne G, eds. Proc. of the 8th Int'l Workshop. Frascati: Springer-Verlag, 2001. 133−148.

[3] Mong LL, Tok WL, Wai LL. Designing functional dependencies for XML. In: Christian S, Keith G, eds. Proc. of the 8th Int'l Conf. on Extending Database Technology. Springer-Verlag, 2002. 124−141.

[4] Arenas M, Libkin L. A normal form for XML documents. In: Lucian P, ed. Proc. of ACM Symp. on Principles of Database Systems (PODS). Madison: ACM Press, 2002. 85−96.

[5] Tan ZJ, Shi BL. Normalization for DTD. Journal of Computer Research and Development, 2004,41(4):594−601 (in Chinese with English abstract).

[6] Mary FF, Atsuyuki M, Dan S, Wang CT. Publishing relational data in XML: The SilkRoute approach. IEEE Data Engineering Bulletin, 2001,24(2):12−19.

[7] Shanmugasundaram J, Gang H, Tufte K, Zhang C, Dewitt D. Relational databases for querying XML documents: Limitations and opportunities. In: Atkinson MP, Orlowska ME, Valduriez P, Zdonik SB, Brodie ML, eds. Proc. of the 25th VLDB Conf. Edinburgh, Scotland: Morgan Kaufmann Publishers, 1999. 302−314.

[8] Deutsh A, Fernandez M. Storing semistructured data with Stored. In: Delis A, Faloutsos C, Ghandeharizadeh S, eds. ACM SIGMOD Int'l Conf. on Management of Data. Philadelphia: ACM Press, 1999. 431−442.

[9] Chen Y, Davidson SB, Zheng YF. Constraints preserving schema mapping from XML to relations. In: Fernandez MF, Papakonstantinou Y, eds. Proc. of the 5th Int'l Workshop on the Web and Databases, WebDB 2002. Madison: ACM Press, 2002. 7−12.

[10] Wang Q, Zhou JM, Wu HW, Xiao JC, Zhou AY. Mapping XML documents to relations in the presence of functional dependencies. Journal of Software, 2003,14(7):1275−1281 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/1275.htm

[11] Chen Y, Davidson SB, Hara CS, Zheng YF. RRXF: Redundancy reducing XML storage in relations. In: Freytag JC, Lockemann PC, Abiteboul S, Carey MJ, Selinger PG, Heuer A, eds. Proc. of the 29th Int'l Conf. on Very Large Data Bases (VLDB). Berlin: Morgan Kaufmann Publishers, 2003. 189−200.

[12] Davidson SB, Fan WF, Hara CS, Qin J. Propagating XML constraints to relations. In: Dayal U, Ramamritham K, Vijayaraman TM, eds. Proc. of the 19th Int'l Conf. on Data Engineering (ICDE). Bangalore: IEEE Computer Society, 2003. 543−556.

附中文参考文献:

[5] 谈子敬,施伯乐.DTD 的规范化.计算机研究与发展,2004,41(4):594−601.

[10] 王庆,周俊梅,吴红伟,萧建昌,周傲英.XML 文档及其函数依赖到关系的映射.软件学报,2003,14(7):1275−1281. http://www.jos.org.cn/1000-9825/14/1275.htm