第35卷 第19期 计 算 机 工 程 2009年10月
Vol.35 No.19 Computer Engineering October 2009
·软件技术与数据库· 文章编号：1000—3428(2009)19—0097—02 文献标识码：A 中图分类号：TP391

无环 XML 文档的研究

王桂兰，王振旗，罗贤缙
(华北电力大学信息与网管中心，保定 071003)

摘 要：无环数据库有许多优良特性，因此，无环成为判断数据库模式优劣的重要特性。针对以上情况，以关系数据库中无 γ 环数据库模式为参考，给出 XML 文档中无 XML_γ 环的相关定义、表示、特征及无 XML_γ 环的 XML 数据模式的设计方法，理论分析证明，所设计的规则是有效的。

关键词：规范化；二义性；XML_γ 环

Research of Acyclic XML Document

WANG Gui-lan, WANG Zhen-qi, LUO Xian-jin
(Information and Network Management Center, North China Electric Power University, Baoding 071003)

【Abstract】 Acyclic database has many good characteristics, so acyclic is an important characteristic for database schema. Based on γ acyclic database schema, this paper presents the definitions of none XML_γ cycle, the characteristics of none XML_γ cycle in XML documents and design method of XML schema for none XML_γ cycle. Theory analysis proves the effectiveness of the rules designed.

【Key words】 normalization; ambiguity; XML_γ cycle

在关系数据库中，无环数据库有许多优良特性，因此，无环成为判断数据库模式优劣的另一重要特性。在对无环的研究过程中，发现数据库模式存在不同级别的无环性，文献[1]描述了 3 种重要的无环类型：无α环，无β环，无 γ 环，且无 γ 环⇒无β环，无β环⇒无α环。其中，无 γ 环数据库模式消除了查询二义性，而且具有许多优良特性。网络技术的发展使 XML 成为事实上的网络传输标准。随着大量关系数据向 XML 数据转化，关系数据库中存在的问题，比如查询二义性，不可避免地出现在了 XML 数据中。XML 数据可以表示为树型结构，XML 数据的查询以路径为基础，如果从某一节点出发，通过不同的路径到达了相同的节点，就会产生查询二义性。

1 相关定义

定义 1 DTD 的简图 $G=(N,L)$，其中，N 是图中的节点集合；L 是图中的有向边集合[2]。

定义 2 DTD 简图节点 $N=(E|A|*)$，其中，E 是 DTD 中出现的元素(element)；A 是 DTD 中出现的属性(attribute)；$*$ 为操作符，表现为 0 次或多于 0 次；每个元素和属性在 DTD 简图中只能出现一次。

定义 3 DTD 简图的边是从 N_1 到 N_2 的有向边。其中，N_1，$N_2 \in N$，N_2 是 N_1 的子元素或属性，如果 DTD 简化后子元素 N_2 带有操作符 $*$，则除去 N_1 到 N_2 的边，增加 N_1 到 $*$ 和 $*$ 到 N_2 的 2 条边。

当 DTD 图中包含有环时，从图中的某个节点出发到达相同节点就有 2 条不同的路径，当 2 条路径分别表示不同的语义时，就产生了模糊的结果，即查询二义性。

数据依赖是数据库理论中最主要的组成部分，是数据库模式设计的理论基础。数据依赖表示数据间存在的一种限制或制约关系。常见的数据依赖有函数依赖、多值依赖等。XML 文档是面向数据的，数据之间必然有各种依赖关系。XML 文档中函数依赖的定义 FD_{XML}[3]如下：

定义 4 XML 函数依赖 FD_{XML} 的表达方式为

$$\left(Q,\left[P_{x1},P_{x2},\cdots,P_{xn}\rightarrow P_y\right]\right)$$

其中：(1) Q 是 FD_{XML} 的头路径并且是一条完整路径(即从文档的根节点出发)。它说明了此 FD_{XML} 的定义范围。(2)每一个 P_{xi} 表示一个 LHS(Left-Hand-Side)实体类型。一个 LHS 实体类型由一个在 XML 文档中的元素名和可选择的主键属性组成。LHS 实体类型的实例被称为 LHS 实体，可被主键属性独一无二地标示。(3) P_y 表示 RHS(Right-Hand-Side)实体类型。每个 RHS 实体类型由一个在 XML 文档中的元素名和可选择的主键属性组成。RHS 实体类型的实例被称为 RHS 实体。

对于任何 2 个在 FD_{XML} 头路径 Q 下的子树，假如所有的 LHS 实体有相同的值，那么所有的 RHS 实体也有相同的值。

每条边只与 2 个节点相关联的图称为线图，每条边可以关联多个节点的图称为超图。线图是图论中应用最广的一种结构。但是，由于线图中限定每条边的关联节点为 2 个，因此限制了线图的表达能力。在现实世界中广泛存在着各种多元联系，难以用线图直观地表达，这就促使人们对超图进行研究。

定义 5 一个超图 H 是一个有序二元组 $H=<V,E>$，其中，V 是一个有限集，V 中的元素称为 H 的节点；E 是一条超边(简称为边)的集合，E 中每一条超边都是 V 的一个非空子集，并使 V 中每个节点至少属于 E 中的一条超边。若一个数据库模

基金项目：华北电力大学青年教师基金资助项目(200611021)
作者简介：王桂兰(1979－)，女，讲师、硕士，主研方向：XML 数据库，OLAP；王振旗，教授、硕士；罗贤瑨，讲师、硕士
收稿日期：2009-01-10 **E-mail：**yu_bing_2000@163.com

式 $R=\{R_1,R_2,\cdots,R_p\}$的对应超图为 H_R，则 $H_R=<V, E>$，其中，V 是节点的集合；E 是 H 中边的集合。R 中的每一个属性 A_i 对应于 H_R 中的一个节点 v_i；R 中的每一个关系模式 R_j 对应于 H_R 中的一条超边 e_j，即 $v_i \in e_j$，当且仅当 A_i 在关系模式 R_j 中。如果 H 中不存在任一超边完全包含在另一超边中，称 H 为化简超图，记为 $RED(H)$。

定义 6 DTD 图是一个有向图：$G=(V_e, V_a, V_o, E_e, E_o, f_e, f_o)$，且 $V=V=V_e \cup V_a \cup V_o$，其中：(1)$V_e$ 是 DTD 中元素节点的集合；(2)V_a 是 DTD 中属性节点的集合；(3)V_o 是 DTD 中操作符节点的集合，$V_o=\{*, ?, +\}$；(4)E_e 是元素与属性或元素与元素之间边的集合；(5)E_o 是元素或属性与操作符之间边的集合；(6)f_e 是 E_e 到 $V\times V$ 上的一个映射(函数)；(7)f_o 是 E_o 到 $V\times V$ 上的一个映射(函数)。

定义 7 XML 关系模式(R_{XML})是根据文档 DTD 中的 FD_{XML} 关系将在同一 FD_{XML} 中的元素组合成一个关系，所有 FD_{XML} 的组合就形成了 XML 关系模式。

DTD 和 XML Schema 都可以看作 XML 文档的模式。这里定义的 XML 关系模式关注的是隐藏在 XML 文档中的数据关系，不太考虑文档中元素之间的顺序和元素之间的层次关系。这也比较适合讨论面向数据的 XML 文档。

定义 8 设 XML 文档 DTD 图中的元素节点集 U 及 U 上的 XML 关系模式 R_{XML}，如果把 R_{XML} 中的元素作为超图 H_{XML} 中的节点，R_{XML} 中同一关系模式中的元素用一条超边表示，则称 H_{XML} 为 XML 关系模式 R_{XML} 的超图。

定义 9 如果在关系 XML 超图 H_{XML} 中存在这样一个边和点的序列 $S_1,v_1,S_2,v_2,\cdots,S_n,v_n,S_{n+1}$，且满足：(1) $v_1,v_2,\cdots,v_n$ 是 H_{XML} 中的不同节点；(2) $S_1,S_2,\cdots,S_n$ 是 H_{XML} 中的不同边，且有 $S_1=S_{n+1}$；(3) $n\geq 3$，即序列中至少有 3 条不同的边；(4) $v_i \in S_i \cap S_{i+1}, 1\leq i\leq n$；(5)对于所有的 $1\leq (i,j)\leq n, i\neq j, j\neq i+1, j\neq i-1$，$v_i \notin S_j$，则称这样的序列为一个 XML_$\gamma$ 环。

定义 10 在一个 XML 超图 H_{XML} 中，若不存在任何 XML_γ 环，则称该图是无 XML_γ 环图，否则，称为有 XML_γ 环图。如果一个 XML 文档对应的超图 H_{XML} 是一个无 XML_γ 环图，则称该文档为无 XML_γ 环文档，否则，称为有 XML_γ 环文档。

2 XML 超图的生成算法

H_{XML} 是判断和构造无环 XML 文档的基础，生成准确合理的 H_{XML} 对下一步的工作有重要的意义，首先给出生成 H_{XML} 的算法。

算法 构造 XML 超图(H_{XML})

输入 XML 文档的 DTD

输出 XML 超图 $H_{\mathrm{XML}}=\{N,E\}$

(1)生成文档 DTD 简图。

(2)根据文档 DTD 中的语义关系和层次关系确定 DTD 中的 FD_{XML}，并去除一些无关 FD_{XML}。

(3)根据得到的 FD_{XML} 集合，写出文档的 XML 关系模式 $R_{\mathrm{XML}}=\{R_1,R_2,\cdots,R_k\}$。

(4)由 R_{XML} 生成 XML 超图 $H_{\mathrm{XML}}=\{N,E\}$。

3 无 XML_γ 环的 XML 数据模式判定

对于给定的任一文档的 DTD，要判断该文档中是否存在 XML_γ 环，可以在超图 H_{XML} 中观察是否存在环，但这样仍存在局限性。下面给出一个完成此判断的算法：

算法 判断一个 XML 文档是否存在 XML_γ

输入 XML 模式 $R_{\mathrm{XML}}=\{R_1,R_2,\cdots,R_k\}$

输出 假如是无 XML_γ 环，输出 true；否则，输出 false

(1)构造 R_{XML} 对应的 XML 超图 $H_{\mathrm{XML}}=(N,E)$。

(2)对 H_{XML} 反复施加以下规则，直到不能施加为止：

1)如果 1 个节点仅属于 1 条边，则删除节点；

2)如果 1 条边仅含 1 个节点，则删除这条边；

3)如果 2 条边含有相同的节点，则删除其中 1 条边；

4)如果 2 个节点精确地出现在相同的边集中，则从该边集中的每条边中删除一个节点；

5)如果一条边不含任何节点，则删除该边。

(3)如果 H_{XML} 为空集，输出 true，否则，输出 false。

上述算法中主要是几个规则的应用。根据有关 XML_γ 环的定义，一个 XML_γ 环是一个边和点的序列：$S_1,v_1,S_2,v_2,\cdots,S_n,v_n,S_{n+1}$。在 XML_$\gamma$ 环路中，点 v_i 涉及 2 条边，而规则 1)应用后删除的仅是一个独立节点而非 v_i，不会影响 XML_γ 环。在 XML_γ 环路中，每个 v_i 是超图 H_{XML} 中的不同节点，因此，一条边至少含有 2 个不同的节点，仅含有一个节点的边不会出现在环中，规则 2)的应用同样不会影响 XML_γ 环。在 XML_γ 环路中，边 $S_1,S_2,\cdots,S_n$ 是 H_{XML} 中的不同超边，不会出现完全相同的 2 条边，因此，应用规则 3)删除完全相同的 2 条边中的一条也不会影响 XML_γ 环。若节点 A 和节点 B 同时出现在环的边集 S' 的每条边中，而没有单一的节点 A 或 B 出现在除 S' 之外的环的其他边中，则删除 A, B 中的一个不会影响环的连通。可见，若 H_{XML} 有 XML_γ 环，算法中规则的应用不会破坏 H_{XML} 中的 XML_γ 环，仅减少环中边所包含的节点数。因此，对于一个有 XML_γ 环的超图，算法执行的最后结果不会是空集。

4 无 XML_γ 环的数据模式设计

在设计无 XML_γ 环文档时，可以借鉴关系数据库中的无环数据库模式的设计思路加以实现。无论是在 XML 文档中还是在关系数据库中，数据之间的依赖关系都是组织数据的核心。在关系数据库中，可以根据数据之间的依赖关系对数据模式进行分解，使之满足一定的范式(如 3NF, BCNF)要求，对存在二义性的属性进行重命名以克服二义性，满足无环性的要求。在 XML 文档中，需要根据数据之间的依赖关系确定嵌套关系、属性的所有关系。为了减小数据的冗余，需要根据元素之间的语义关系对文档进行适当的分解。当有标签表示了双重语义时，在没有特定的语义环境中查询特定内容就会产生模糊的语义，造成查询二义性，从而导致 XML_γ 环。

无 XML_γ 环 XML 关系模式 R_{XML} 的设计是一个对应超图的设计序列：$H_1,H_2,\cdots,H_n, H_1=(N_1,E_1)$，$E_1$ 仅有一条超边，$R_{\mathrm{XML}}=H_1$。对 $H_i=(N_i,E_i), H_{i+1}=(N_{i+1},E_{i+1}), 1\leq i\leq n$，$(H_i,H_{i+1})$ 称为一个设计步，超边 $e' \in E_{i+1}$，$N_{i+1}=N_i \cup e', E_{i+1}=E_i \cup e', e' \cap N_i$ 称为连接节点集。

设计规则如下：

(1)对于 $H_i=(N_i,E_i)$，若有 $e\in E_i$，$N'=\{u \mid u\in E_i-e\}$ 使得 $e' \cap N_i \subseteq e$ 且 $e \cap N_i' \subseteq e' \cap N_i$，则 $H_{i+1}=(N_i \cup e', E_i \cup e')$。

(2)对于 $H_i=(N_i,E_i)$，如果 E_i 中每条边 e 使得 $e' \cap N_i \subseteq e$

(下转第 101 页)