

一个基于粗糙集的 XML 函数依赖发现算法

孙 伟 刘大昕

(哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)

摘 要 数据依赖是数据库的一个重要概念。函数依赖是一种常见的数据依赖关系,是数据语义的重要组成部分。随着 XML 文档的大量出现,这一概念被引入到 XML 的领域中。本文在约束限制范围的基础上,给出了 XML 函数依赖的定义。引入粗糙集解决 XML 数据不完整的特点,给出 XML 函数依赖的判定定理。并且提出了一个发现 XML 文档中最小非平凡函数依赖的算法。该算法基于一致集的概念,通过不可分辨关系划分元组集减少求一致集的运算次数,使用逐层求精的算法来计算最小非平凡 XML 函数依赖集的左部。通过该算法得到的 XML 函数依赖的语义信息对数据存储模式设计、查询优化和更新异常检查来说是十分重要的。

关键词 可扩展标记语言,语义,数据依赖,XML 函数依赖(XFD),粗糙集

Rough Set Based Algorithm of Discovering Functional Dependency for XML

SUN Wei LIU Da-Xin

(College of Computer Science and Technology, Harbin Engineering University, Harbin 150001)

Abstract Data dependency is an important concept in database research. Functional Dependency plays a fundamental role in data dependency where it provides semantics for the data. Since XML recently appear largely, XFD is introduced into XML research. On the basis of the range of constraints, XFD is redefined in XML. And we propose one algorithm that discovering minimal XFD, based on the rough set handling non-integrity data. It is based on the concept of agree sets, and the computation of agree sets can be decreased by using stripped partition tuples on indiscernibility relation. A levelwise algorithm is used for computing the left hand sides (lhs) of minimal non-trivial functional dependency. The semantics information is important to the design of XML schema, query optimization and check of update abnormality.

Keywords XML, Semantics, Data dependency, Functional dependency, Rough set

1 前言

1998 年 2 月, W3C 推出了可扩展标记语言 XML。随着 XML 成为 Web 上进行半结构化数据表示和数据交换的标准, XML 数据相关技术的研究成为热点。例如: XML 数据的存储技术与发布技术的研究, XML 数据查询与优化技术等。在这些方面的研究不仅要考虑 XML 文档的内容和结构信息,更需要充分利用 XML 文档的语义信息。而 XML 在提供了灵活语法的同时,提供了极少的语义内容,由此,一些文章重点研究提高 XML 的语义表达。在这些方法中,较有效的是在 XML 中定义完整性约束。这些完整性约束包括,键约束^[1,2],路径约束^[3]和包含约束^[4]。但是,将关系型数据库的完整性约束扩展到 XML 领域的研究较少。

函数依赖和多值依赖是数据库中语义的重要组成部分。Arenas^[5]等人最先给出函数依赖在 XML 中的定义和基于 XML 函数依赖的 XML 模式,但是该定义建立在复杂的树元组概念上,没能将 XML 的函数依赖与关系型数据库中的函数依赖关系对应。Vincent^[6~8]等人将这两个概念引入到 XML 领域,给出了相应的定义以及相关范式的。在 XML 函数依赖的定义中,没有能体现关系型数据库中函数依赖关系对约束保持范围的要求。在国内对 XML 函数依赖的研究主要集中在对 XML 模式规范化的研究^[9,10]。

粗糙集(Rough sets)理论^[11]是由波兰华沙理工大学 Pawlak 教授于 20 世纪 80 年代初提出的一种研究不完整、不

确定知识和数据的表达、学习、归纳的理论方法,近年来得到国际上众多学者的重视。现在该理论被广泛应用于机器学习、知识发现和专家系统等领域。基于传统不分明关系的粗糙集理论是不能处理不完备信息系统的,对传统的粗糙集理论进行了扩充^[12]。

本文的主要工作,给出了 XML 函数依赖的定义。由于 XML 中数据的不完备,利用粗糙集的不可分辨关系,提出一个发现最小非平凡 XML 函数依赖的算法,并且通过仿真实验验证算法的正确性和有效性。

本文第 2 节给出 XML 函数依赖的定义;第 3 节结合粗糙集对不完备数据的处理方法,提出 XML 函数依赖的发现算法;最后是结论和未来工作展望。

2 XML 函数依赖定义

在本节中,我们在给出一些基本定义的基础上,给出 XML 函数依赖的定义。XML 文档被看作树型结构,我们首先给出 XML 树的定义。

定义 1(XML 函数依赖) 给定 DTD $D=(E, A, P, R, r)$,任意的 XML 文档树 T 符合 D , $Paths(D)$ 为 D 中所有路径的集合, $Paths(T)$ 为文档树 T 上的路径集合, D 上函数依赖定义为如下形式: $(H[P \rightarrow Q])$ 。其中:

(1) H 是 XML 函数依赖的头部路径,即从 XML 文档的根节点开始的路径,是一个完全限定路径表达式,定义约束保持的范围;

(2) P 是 XML 函数依赖的左部路径, P 是 $Paths(T)$ 的子集并且对任意 $p \in P$, 有 $Last(p) \in EU \cup \{S\}$;

(3) Q 是 XML 函数依赖的右部路径, Q 是 $Paths(T)$ 的子

集并且对任意 $q \in Q$, 有 $Last(q) \in EU \cup \{S\}$ 。

对于文档树 T 中任意两个树元组 t_1 和 t_2 , 在头部路径 H 约束保持范围内, 若存在 $t_1[P] = t_2[P]$, 一定有 $t_1[Q] = t_2[Q]$ 。

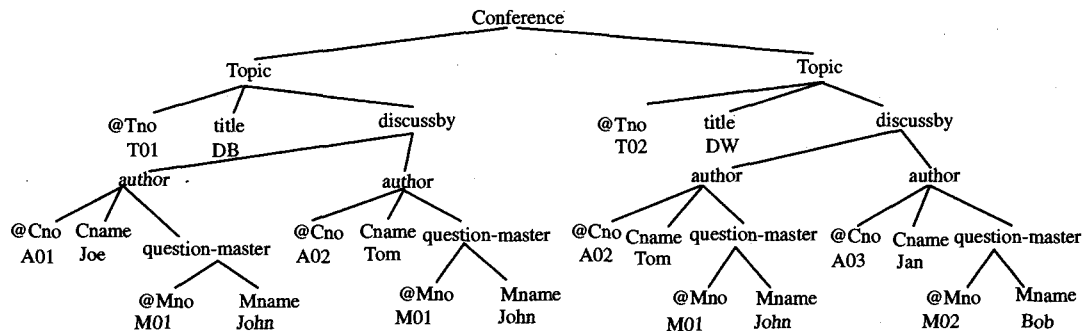


图1 含有 XML 函数依赖关系的 XML 树实例

XML 平凡函数依赖: 给定 DTD $D = (E, A, P, R, r)$, 和 XML 函数依赖 φ , 如果 φ 在任意满足 DTD D 上的文档树 T 都是满足的, 称 φ 是平凡的 XML 函数依赖。

XML 函数依赖集: 给定 DTD $D = (E, A, P, R, r)$, 任意的 XML 文档树 T 符合 D , $Paths(T)$ 为文档树 T 上的路径集合, D 上函数依赖集定义为如下形式: $\Sigma = \{\varphi | \varphi \text{ 是满足 DTD } D \text{ 或 XML 树 } T \text{ 上的函数依赖}\}$, 表示所有 XML 函数依赖的集合。

3 XML 函数依赖发现算法

对于 XML 文档时常会出现数据的不完整。因此, 我们引入粗糙集理论来处理这一问题。对于不完备的理解, 存在两种语意解释: 遗漏语意和缺席语意。这里我们采用遗漏语意, 即认为遗漏值可以与任意值相匹配, 在处理后采用容差关系^[12]。上近似(X^B)和下近似(X_B)分别定义为:

$$X_B = \bigcup \{I_B(x) | x \in \bigcup, I_B(x) \subseteq X\}$$

$$X_B = \bigcup \{I_B(x) | x \in \bigcup, I_B(x) \cap X \neq \emptyset\}$$

在上节给出 XML 函数依赖定义的基础上, 将图 1 所示 XML 树, 在 Conference/Topic/discussby/Author 约束保持范围内, XML 树转化为信息表, 如表 1 所示。

表1 图1的对应信息表

	Ano	Aname	Mno	Mname
1	A01	Joe	M01	John
2	A02	Tom	M01	John
3	A02(*)	Tom	M01	John
4	A03	Jan	M02	Bob

根据信息表, 我们依据不可分辨关系 $IND(B)$ 对信息表划分, 实现对信息表的细化。不可分辨关系是等价关系, 依据不可分辨关系可以得到子属性的等价类。我们有定理 1。

定理 1 在信息表中, 属性集 $S, P, Q \subseteq S$, P 的划分是 $X = \{P_1, \dots, P_n\}$, Q 的划分是 $Y = \{Q_1, \dots, Q_m\}$, 函数依赖 $P \rightarrow Q$ 成立, 当且仅当 $\sigma(P, Q_1) \cup \dots \cup \sigma(P, Q_m) = |P|$, 其中 $\sigma(P, Q_i) (1 \leq i \leq m)$ 为 Q_i 在 P 上的下近似, $|P|$ 表示信息表元组数目。(证明略)

定义 2 (一致集) 设 $t_1, t_2 \in T, P \subseteq Paths(D)$; 若 $t_1[P] = t_2[P]$, 则称 t_1 和 t_2 在 P 上一致。 t_1 和 t_2 的一致集 (agree set) 定义为 $ag(t_1, t_2) = \{Q \in Paths(D) | t_1[Q] = t_2[Q]\}$ 。树

元组 T 的一致集定义为 $ag(T) = \{ag(t_1, t_2) | t_1, t_2 \in T, t_1 \neq t_2\}$ 。

定义 3 (属性最大集) 设 $Q \in Paths(D)$, 属性 Q 的最大集 (maximal set) $\max(\text{dep}(T), Q)$ 是 T 中不能决定 Q 的最大属性集 P 的集合, 即 $\max(\text{dep}(T), Q) = \{P \subseteq Paths(D) | T \models P \rightarrow Q \text{ 且 } \forall Z \subseteq Paths(D), P \subset Z, T \not\models Z \rightarrow Q\}$ 。

设 $Q \in Paths(D)$, T 中函数依赖集 $\text{dep}(T)$ 的左部集定义为 $\text{lhs}(\text{dep}(T), Q) = \{P \subseteq Paths(D) | T \models P \rightarrow Q \text{ 且 } \forall Z \subset P, T \not\models Z \rightarrow Q\}$ 。显然, $\{P \rightarrow Q | P \in \text{lhs}(\text{dep}(T), Q) Q \in Paths(D)\}$ 与 $\text{dep}(T)$ 等价。

最小非平凡函数依赖关系的抽取, 根据上节给出的 XML 函数依赖推理规则, 右部取单个属性, 来确定左部。问题是在选取左部用定义进行推导时, 存在组合问题, 搜索空间很大。在我们的方法中是通过信息表中树元组的划分, 这种方法的步骤是计算一致集 $ag(T)$, 根据 $ag(T)$ 计算最大集 $\max(\text{dep}(T), Q)$ 和它的补集 $\text{cmax}(\text{dep}(T), Q)$, 由 cmax 集算出属性 A 的左部 $\text{lhs}(\text{dep}(T), Q)$, 其中只有计算 $ag(T)$ 与树元组集 T 有关。

(1) 带状划分

设 $X \subseteq S, t \in T, [t]_X = \{t' \in T | t[X] = t'[X]\}$, 则 $[t]_X$ 是 X 的一个基本等价类; X 在 T 中形成的基本等价类的集合构成 T 的一个划分 (partition), 表示为 $X^* = \{[t]_X | t \in T\}$ 。若划分 X^* 中的一个等价类只有一个元组, 则 T 中没有与它在 X 上一致的其他元组。所以, 我们定义属性集 X 的带状划分 (stripped partition) $\bar{X}^* = \{c \in X^* | |c| > 1\}$ 。

对于缺少元素的处理, 认为可以归为任意等价类。

我们在求划分时, 求属性 Q 的划分, 表示为 $X_Q^* = \{\sigma(P, Q_i) | P \in S\}$, 带状划分表示为 $\bar{X}_Q^* = \{c \in X_Q^* | |c| > 1\}$ 。 Q 的划分是 $Y = \{Q_1, \dots, Q_m\}$, 其中 $\sigma(P, Q_i) (1 \leq i \leq m)$ 为 Q_i 在 P 上的下近似。

例 1 我们取 $Q = \text{Aname}$ 属性, T 的等价类为 $\{\{1\}, \{2, 3\}, \{4\}\}$ 。属性 $\text{Ano}, \text{Mno}, \text{Mname}$ 的等价类为 $\{\{1, 3\}, \{2, 3\}, \{3, 4\}\}, \{\{1, 2, 3\}, \{4\}\}, \{\{1, 2, 3\}, \{4\}\}$, 分别求下近似 $\{\{1\}, \{2, 3\}, \{4\}\}, \{\{1, 2, 3\}, \{4\}\}, \{\{1, 2, 3\}, \{4\}\}$ 。所以, $\bar{X}_Q^*$ 分别为 $\{\{2, 3\}\}, \{\{1, 2, 3\}\}, \{\{1, 2, 3\}\}$ 。

(2) 计算一致集

针对属性 Q , 我们求其它属性的一致集 $ag(\bar{Q})$ 。首先定义一个最大等价类的集合 $MC = \text{Max} \subseteq \{c \in X_Q | X \in S - Q\}$,

其中的算子“ $\text{Max} \subseteq$ ”是求集合中的按 $\subseteq$ 最大的成员。

例2 例1中的 $\text{MC} = \{\{1, 2, 3\}\}$ 。

定理2 $\text{ag}(\bar{Q}) = \bigcup_{c \in \text{MC}} \text{ag}(c)$, 其中

$\text{ag}(c) = \{\text{ag}(t, t') \mid t, t' \in c, t \neq t'\}$

证明: 设 $t, t' \in T, t \neq t'$ 。

若 $\text{ag}(t, t') \neq \emptyset$, 不妨设 $A \in \text{ag}(t, t'), A \neq Q$

有 $t[A] = t'[A]$, 则 $\exists c' \in A_Q, t, t' \in c'$ 。

根据 MC 的定义, $\exists c \in \text{MC}, c' \subseteq c$, 则, $t, t' \in c, \therefore \text{ag}(\bar{Q}) = \bigcup_{c \in \text{MC}} \text{ag}(c)$ 。

属性 Q 上的最大等价类的集合 $\text{MC}' = \bar{Q}$ 。当计算 $\text{ag}(\bar{Q})$ 时, 只需考虑 MC 等价类中配对的元组减去 MC' 中配对的元组, 即不考虑 Q 属性上相等的树元组对。当 T 中属性的取值大不相同, 通过带状划分树元组去除掉很多单元组成的等价类, 使其不与其他元组配成对, 这样就使计算 $\text{ag}(\bar{Q})$ 的时间大大减少了。记作: $\text{ag}'(\bar{Q})$ 。

例3 我们取 $Q = \text{Aname}$ 属性, $T = \{\{1\}, \{2, 3\}, \{4\}\}$ 。
 $X_Q = \{\{1, 2, 3\}\}$ 。 $\text{MC} = \{\{1, 2, 3\}\}$, $\text{MC}' = \{\{2, 3\}\}$, 所以, $X_Q = \{\{2, 3\}\}$ 。 Q 的一致集, 产生的配对元组有 $(1, 2)$ 和 $(1, 3)$;
 $\text{ag}'(\bar{Q}) = \{\text{MnoMname}\}$ 。

(3) 计算最大集及其补集

定理3 $\max(\text{dep}(T), Q) = \text{Max} \subseteq \{\text{ag}'(\bar{Q})\}$

证明: 设 $X \in \text{Max} \subseteq \{\text{ag}'(\bar{Q})\}$ 。则 $\exists t, t' \in T, \text{ag}(t, t') = X, t[X] = t'[X]$ 。

$\therefore t[Q] \neq t'[Q], \therefore T \not\models X \rightarrow Q$ 。

对 X 的任意一个超集 $X' \subseteq S$, 若 $Q \in X'$, 则 $T \vdash X' \rightarrow Q$;
若 $Q \notin X'$, 根据 X 的性质, $X' \notin \text{ag}'(\bar{Q})$, 所以 $\forall t, t' \in T, t \neq t', t[X'] \neq t'[X'], \therefore T \vdash X' \rightarrow Q$, 所以, $X \in \max(\text{dep}(T), Q)$ 。

另一方面, 设 $X \in \max(\text{dep}(T), Q) = \{X \mid T \not\models X \rightarrow Q, X$ 的任意一个超集 $X' \subseteq S, T \vdash X' \rightarrow Q\}$ 。

$\therefore T \not\models X \rightarrow Q, \therefore Q \notin X$, 并且 $\exists t, t' \in T, X \subseteq \text{ag}(t, t')$ 。

若 $\exists B \subseteq \text{ag}(t, t'), B \notin X$, 则 $T \not\models BX \rightarrow Q$, 这与 X 的性质矛盾, 所以 $X = \text{ag}(t, t')$ 。同时, 由 X 的最大的性质, 不存在 X 的超集 $X' \subseteq S, X' \subseteq \text{ag}'(\bar{Q})$ 且 $Q \notin X'$, 所以 $X \in \text{Max} \subseteq \{\text{ag}'(\bar{Q})\}$ 。

定义4 $\text{cmax}(\text{dep}(T), Q) = \{S - X \mid X \in \text{Max} \subseteq \{\text{ag}'(\bar{Q})\}\}$

例4 $\max(\text{dep}(T), Q) = \{\text{MnoMname}\}$, $\text{cmax}(\text{dep}(T), Q) = \{\text{AnoAname}\}$ 。

(4) 计算函数依赖集左部 $\text{lhs}(\text{dep}(T), Q)$

我们引进超图概念。 S 的属性集组成的 H 构成一个简单的超图, H 中的属性集称为超图的边, 每个属性称为超图的顶点。 H 的一个横截 Tr 是 S 的一个属性集, 它与 H 的每一条边都相交。 H 的一个最小横截 Tr 满足: Tr 是 H 的一个横截, $\forall Tr' \subset Tr, Tr'$ 不是 H 的横截。 H 的所有最小横截集记为 $Tr_{\min}(H)$ 。把 $\text{cmax}(\text{dep}(T), Q)$ 看成一个超图, 则有下面的结论:

定理4 $Tr_{\min}(\text{cmax}(\text{dep}(T), Q)) = \text{lhs}(\text{dep}(T), Q)$ 。

证明: 一方面, 设 $Tr \in Tr_{\min}(\text{cmax}(\text{dep}(T), Q))$ 。

若 $T \not\models Tr \rightarrow Q$, 则 $\exists X \in \max(\text{dep}(T), Q), Tr \subseteq X$ 。

$\therefore Tr \cap \bar{X} = \emptyset$, 其中 $\bar{X} = S - X \in \max(\text{dep}(T), Q)$, 与 Tr

$\in Tr_{\min}(\text{cmax}(\text{dep}(T), Q))$ 矛盾。所以 $T \vdash Tr \rightarrow Q$ 。

又 Tr 是最小的横截, $\therefore \forall Tr' \subset Tr, \exists \bar{X} \in (\text{cmax}(\text{dep}(T), Q)), Tr' \cap \bar{X} = \emptyset$ 。

$\therefore Tr' \cap \bar{X} = Tr', X = S - \bar{X} \in \max(\text{dep}(T), Q)$

$\therefore T \not\models Tr' \rightarrow Q, \therefore Tr \in \text{lhs}(\text{dep}(T), Q)$ 。

另一方面, 设 $Y \in \text{lhs}(\text{dep}(T), Q)$

若 $Y \notin Tr_{\min}(\text{cmax}(\text{dep}(T), Q))$, 则 $\exists \bar{X} \in \max(\text{dep}(T), Q), Y \cap \bar{X} = \emptyset$ 。

$\therefore Y \subseteq X, X = S - \bar{X} \in \max(\text{dep}(T), Q), \therefore T \not\models Y \rightarrow Q$, 与 $Y \in \text{lhs}(\text{dep}(T), Q)$ 矛盾。

$\therefore Y \in Tr_{\min}(\text{cmax}(\text{dep}(T), Q))$ 。

例5 $\text{lhs}(\text{dep}(T), Q) = \{\text{AnoAname}\}$, 由 $\text{lhs}(\text{dep}(T), Q)$ 由得到关于 Q 的最小函数依赖集: $T \vdash \text{Ano} \rightarrow \text{Aname}$ 。

结论与未来工作展望 本文首先给出基于树元组的 XML 函数依赖的一个定义, 定义中采用约束限制范围很好地解决了 XML 函数依赖的有效性属性集范围有关的问题。在此基础上, 转化为信息表, 利用不可分辨关系对信息表进行带状划分, 并在一致集的基础上给出最小非平凡 XML 函数依赖的发现算法。通过在人工生成的 XML 文档上的仿真实验, 验证了算法的正确性, 同时验证算法的执行时间与文档的大小呈线性关系, 说明算法是有效的。在 XML 文档集中自动发现最小非平凡 XML 函数依赖关系可广泛应用于 XML 模式规范化、XML 的存储模式设计、查询优化等。

参考文献

- 1 Arenas M, Libkin L. A normal form for xml documents. In: Proc. ACM PODS Conference, 2002. 85~96
- 2 Buneman P, Davidson S, Fan W, Hara C. Reasoning about keys for xml. In: International Workshop on Database Programming Languages, 2001
- 3 Buneman P, Davidson S, Fan W, et al. Keys for xml. Computer Networks, 2002, 39(5): 473~487
- 4 Buneman P, Fan W, Weinstein S. Path constraints on structured and semistructured data. In: Proc. ACM PODS Conference, 1998. 129~138
- 5 Fan W, Simeon J. Integrity constraints for xml. In: Proc. ACM PODS Conference, 2000. 23~24
- 6 Vincent M W, Liu J, Liu C. Strong functional dependencies and their application to normal forms in XML. ACM Trans. Database Syst., 2004, 29(3): 445~462
- 7 Liu Jixue, Vincent M W, Liu Chengfei, et al. Mohania; Checking Multivalued Dependencies in XML. APWeb, 2005. 320~332
- 8 Vincent M W, Liu Jixue. Checking Functional Dependency Satisfaction in XML. XSym, 2005. 4~17
- 9 吴永辉. 用于 XML 模式和 DTD 规范化设计的层次模式设计. 软件学报, 2004, 15(7): 1099~1106
- 10 谈子敬, 施伯乐. 函数依赖和规范化在关系和 XML 间的传播. 软件学报, 2005, 16(4): 533~539
- 11 Pawlak Z. Rough Sets. International Journal of Information and Computer Science, 1982, 11(5)
- 12 王国胤. Rough 集理论在不完备信息系统中的扩充. <http://www.paper.edu.cn>