

文章编号:1001-9081(2007)05-1228-04

## XML 弱函数依赖及其推理规则

苏 召, 刘国华

(燕山大学 信息科学与工程学院, 河北 秦皇岛 066004)

(suzhao2005@yahoo.com.cn)

**摘要:** XML 函数依赖问题是进行 XML 数据库后续研究的基础。首先基于 M. Arenas 等人给定的 XML 中 DTD 和 XML 树的定义, 提出空值、不完全树元组、数据值偏序、最小扩展树等概念, 在此基础上, 给出弱函数依赖及其满足性的定义; 其次研究了 XML 弱函数依赖的逻辑蕴含问题, 提出一组适合 XML 空值模型的函数依赖推理规则集; 最后给出推理规则集的正确性和完备性证明。

**关键词:** XML; 空值; 不完全树元组; 弱函数依赖; 完备性

**中图分类号:** TP311.131    **文献标识码:**A

## Weak functional dependencies and inference rules for XML

SU Zhao, LIU Guo-hua

(College of Information Science and Engineering, Yanshan University, Qinhuangdao Hebei 066004, China)

**Abstract:** The problem of functional dependencies for Extensible Markup Language (XML) is the foundation of further research for XML database. In this paper, first, based on the definitions of Document Type Definition (DTD) and XML tree given by M. Arenas et al., concepts of null value, incomplete tree tuple, data value partial order and minimal extended tree were proposed. Second, the definition of weak functional dependency and its satisfaction were given. Then the problem of logical implication for weak functional dependencies was studied, and a set of inference rules for XML were presented. Finally, the correctness and completeness of the set were proved.

**Key words:** Extensible Markup Language (XML); null value; tree tuple; weak functional dependency; completeness

## 0 引言

XML 已经成为 Internet 上主要的数据表示和交换标准之一。有关 XML 数据模式的研究已取得了一些成果, 但是还没有形成一个完备的体系。XML 函数依赖是进行 XML 数据模式研究的基础, 因此也就具有十分重要的研究价值。目前, 对于 XML 函数依赖的研究比较有代表性的是 M. W. Vincent 和 Marcelo Arenas 的观点。M. W. Vincent 基于路径、路径表达式<sup>[1]</sup>的方法, 在不存在 DTD 的情况下, 给出函数依赖的定义, 提出一组推理规则, 并证明是正确的, 并在一元函数依赖的前提下证明是完备的。Marcelo Arenas 在存在 DTD 的情况下, 通过将 XML 文档映射到关系模式, 给出函数依赖的定义, 但没有定义函数依赖的满足性, 也没有讨论函数依赖的推理。谈子敬等人在 M. Arenas 定义的函数依赖思想上, 利用树元组<sup>[2]</sup>的方法定义了函数依赖, 给出满足性定义, 并在此基础上提出一组正确完备的推理规则集<sup>[3]</sup>。但是这些研究都是在强函数依赖的定义下进行的。在关系数据库理论的空值关系模型<sup>[4]</sup>中, 对于函数依赖有强函数依赖和弱函数依赖两种。

## 1 XML 中的空值模型

在关系数据库中, 不完全关系是指包含空值的关系, 即某些值到目前为止还不知道。同样, 对于 XML 数据, 它的树型结构特点决定它存在空值的可能性, 它的直观体现就是对于

一棵给定的 XML 树<sup>[2]</sup>, 它的某些路径是不完全的, 用空值表示那些不完全的路径值可以形成一棵扩展树, XML 扩展树中的空值都用允许取的实际值取代后就成为一棵完全树。设树  $T$  是一棵不完全树,  $T$  中的空值用各种可能值代替后所形成的完全树的集合称为这棵树的可能世界, 并将其记做  $Poss(T)$ 。众所周知, 函数依赖在 XML 数据库设计及应用中起着重要作用。那么, 函数依赖在不完全树中怎样体现呢? 这里有以下定义: 给定一个 DTD  $D$ , 设树  $T$  是  $D$  上的一棵不完全树,  $S_1, S_2 \subseteq paths(D)$ ,  $S_1 \rightarrow S_2$  是一个函数依赖。如果存在一个完全树  $T' \in Poss(T)$ , 而  $T'$  适合(弱满足<sup>[5]</sup>) 函数依赖  $S_1 \rightarrow S_2$ , 则称  $T$  适合函数依赖  $S_1 \rightarrow S_2$ 。下面来举例说明。

例 1 给定一个简单的 DTD  $D$ , 与  $D$  相容的 XML 树  $T$  如图 1。 $D$  上的函数依赖集合  $\Sigma = \{db.G @ A \rightarrow db.G @ B, db.G @ B \rightarrow db.G @ C\}$ 。显然  $T$  适合  $db.G @ A \rightarrow db.G @ B$  (将

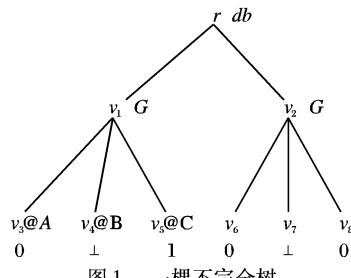


图 1 一棵不完全树

两条路径上  $B$  属性的两个(换成两个相等的值),  $T$  也适合  $db.G @ B \rightarrow db.G @ C$  (将两条路径上  $B$  属性的两个  $\perp$  换成两

收稿日期: 2006-11-20; 修订日期: 2007-01-12

基金项目: 教育部科学技术研究重点项目(205014); 河北省教育厅自然科学指令计划资助项目(2005102)

作者简介: 苏召(1981-), 男, 河北石家庄人, 硕士研究生, 主要研究方向: 半结构化数据、XML; 刘国华(1966-), 男, 黑龙江齐齐哈尔人, 教授, 博士生导师, 博士, 主要研究方向: 数据库理论、数据库安全、Web 数据管理。

个不等的值)。

可以看出,在不完全关系中强函数依赖需要在任意  $T' \in Poss(T)$  上成立,而弱函数依赖只要存在一个关系  $T' \in Poss(T)$  满足此函数依赖即可。

上面所举示例只是当叶子节点存在空值时的情况,当非叶子节点存在空值时的情况类似。

## 2 XML弱函数依赖

关于 DTD、XML 树路径以及树元组的定义请参考文献 [2]。为了把空值关系模型的思想应用到 XML 中,下面先给出几个定义。首先给出一个 XML 树的定义,本定义是文献 [2] 中定义的扩展。

**定义 1** XML 树  $T$  为满足如下条件的六元组  $(V, lab, ele, att, val, root)$ 。

(1)  $V \subseteq Vert$ , 是有限节点集合;

(2)  $lab: V \rightarrow El \cup Att \cup Str$ ;

(3)  $ele: V \rightarrow Str \cup V^*$ ;

(4)  $att$  是  $V \times Att \rightarrow V$  的部分映射。对每个  $v \in V$  和  $l \in Att$ , 如果  $att(v, l) = v_1$  那么  $lab(v) \in El, lab(v_1) = l$ ;

(5)  $val$  是一个函数,对于任意节点  $v \in V$ ,如果  $lab(v) \in El, val(v) = v$ ;如果  $lab(v) \in Att$  或  $lab(v) = Str, val(v) \in Str$ ;

(6)  $root \in V$ , 称为  $T$  的根。

**定义 2** 给定一个 DTD  $D$ ,令  $T$  是一棵 XML 树  $(V, lab, ele, att, val, root)$ ,且  $T \triangleleft D, T'$  是一棵扩展树<sup>[1]</sup>  $(V \cup N, lab, ele, att, val, root)$ ,如果满足以下条件,则称  $T'$  是  $T$  的一个最小扩展树:

(1)  $T$  嵌入<sup>[1]</sup>  $T'$ ;

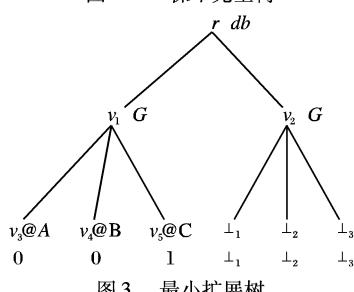
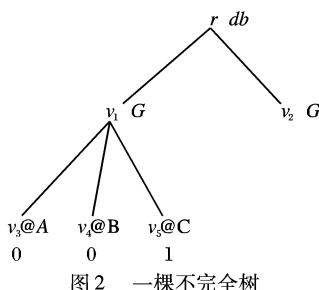
(2)  $T' \triangleleft D$ ;

(3) 如果存在路径  $p_1, p_2 \in paths(D)$ ,使得  $p_1 \subset p_2$ ,并且在  $T'$  中存在定义  $p_1$  在上的路径实例<sup>[1]</sup>  $\bar{v}_1, \dots, \bar{v}_n$ ,那么在  $T'$  中存在定义在  $p_2$  上的路径实例  $\bar{v}_1', \dots, \bar{v}_m'$ ,使得  $\bar{v}_1, \dots, \bar{v}_n$  是  $\bar{v}_1', \dots, \bar{v}_m'$  的前缀;

(4) 不存在其他扩展树  $T''(V \cup N'', lab, ele, att, val, root)$ ,使得  $T''$  满足条件(1),(2)和(3)且  $N'' \subset N$ 。

记  $T$  的最小扩展树为  $M(T)$ 。

下面,给定一棵 XML 树(图 2),并将其扩展为一棵最小扩展树(图 3)。



**定义 3** 用记号  $\perp$  表示未知空值,即到目前尚不知道其值的那一种空值;用记号  $\perp_{inc}$  表示不相容空值,即由于数据不相容引起的空值。设给定一个 DTD  $D$ ,对于定义在  $D$  上的树元组  $t$ ,及任一定义在这个树元组上的路径  $p \in paths(D)$ , $t.p \neq \perp_{inc}, t.p \neq \perp$ ,则称  $t$  是  $D$  的完全树元组,否则  $t$  是不完全树元组;如果存在  $t.p = \perp_{inc}$ ,则称  $t$  是不相容树元组,否则称  $t$  是相容的。如果树  $T \triangleleft D$ ,对于任意  $p \in paths(D)$ ,所有  $t \in tuples_D(T)$  都是完全的,称  $T$  是完全的,否则称  $T$  是不完全的。如果  $t$  是不相容的,则称  $T$  是不相容的,否则称  $T$  是相容的。

图 2 中令树元组  $t_1(db) = r, t_1(db.G) = v_1, t_1(db.G @ A) = 0, t_1(db.G @ B) = 0, t_1(db.G @ C) = 1$ ;因此  $t_1$  是完全树元组。而另一个树元组  $t_2(db) = r, t_2(db.G) = v_2$ ,而  $t_2(db.G @ A), t_2(db.G @ B), t_2(db.G @ C)$  都没有定义,因此  $t_2$  是不完全树元组。同理,图 3 中因为存在  $t_2(db.G @ A) = \perp_1, t_2(db.G @ B) = \perp_2, t_2(db.G @ C) = \perp_3$ ,所以这里的  $t_2$  也是不完全树元组。

**定义 4** 在数据域中定义一个偏序,记做  $\subseteq$ ,具体如下:当且仅当  $a = b$ ,或  $a = \perp$ ,或  $b = \perp_{inc}$  时  $a \subseteq b$ ,这里  $a, b$  是数据域中的两个任意元素。给定一个 DTD  $D, T \triangleleft D$ ,若  $t_1, t_2 \in tuples_D(T)$ ,对于所有在  $t_1, t_2$  上有定义的  $p \in paths(D)$ ,都有  $t_1.p \subseteq t_2.p$ ,则记  $t_1 \subseteq t_2$ 。

**定义 5** 给定一个 DTD  $D$ ,令  $T$  是一棵 XML 树,且  $T \triangleleft D, T$  的一棵完全树构造为:令  $V_1$  是一个与  $V$  不相交的节点集合,令  $v$  是  $M(T)$  中的任意节点,那么对于所有节点  $v$ ,如果  $v \in N$  且  $v \neq \perp_{inc}$ ,那么用  $V_1$  中的任一不等于  $v$  的节点代替  $v$ ,如果  $lab(v) \notin E$ ,那么将任意文本字符串赋值给  $val(v)$ 。记树  $T$  的所有可能的完全树为  $Poss(T)$ 。

仍以图 3 说明,将图中的属性节点  $\perp_1, \perp_2, \perp_3$  以任一  $v \in V_1$  替代,并将任意文本字符串赋值给  $val(v)$  后,所形成的 XML 树即为一棵完全树。如果以所有可能的值代替树中的空值,将形成  $Poss(T)$ 。

**定理 1** 当且仅当  $Poss(T) = \emptyset$  时,XML 树  $T$  是不相容的,否则,XML 树是相容的。

**证明** 显然,当且仅当 XML 树  $M(T)$  上某个路径的树元组  $t.p = \perp_{inc}$  时,不存在任何可以使  $t.p$  的值有效的替代,可以使得  $M(T)$  构造为一棵完全树。即  $Poss(T) = \emptyset$ 。

下面给出 XML 弱函数依赖及其可满足性的定义,即不完全树  $T$  何时适合一个函数依赖。

**定义 6** 一个 DTD  $D$  上的基于树元组的弱函数依赖是形为:  $S_1 \rightarrow S_2$  的一个命题,其中,  $S_1, S_2 \subseteq paths(D)$ ,它的含义是,对于一棵树  $T$ ,有  $T \triangleleft D$ ,当存在一棵树  $T' \in Poss(T)$ ,  $S_1, S_2 \subseteq paths(T')$  满足以下条件:  $\forall t_1, t_2 \subseteq tuples_D(T')$ ,若  $t_1.S_1 = t_2.S_1$ ,则必有  $t_1.S_2 = t_2.S_2$ ,称  $T$  弱满足(也称适合)函数依赖  $S_1 \rightarrow S_2$ ,记做  $T \models S_1 \rightarrow S_2$ 。

**引理 1** 给定一个 DTD  $D$ ,XML 树  $T \triangleleft D, S_1, S_2 \subseteq paths(M(T))$ ,当且仅当  $tuples_D(M(T))$  中的任两个元组  $t_1, t_2$ ,只要  $t_1.S_1, t_2.S_1$  完全而且  $t_1.S_1 = t_2.S_1$ ,就有  $t_1.S_2 = t_2.S_2$ 。 $t_1, t_2$  是相容的,则有  $T$  适合  $S_1 \rightarrow S_2$ 。这里,  $t_1.S_2 \cup t_2.S_2$  是  $t_1.S_2$  与  $t_2.S_2$  根据偏序定义的最小上界。

**证明** 因为  $T$  适合  $S_1 \rightarrow S_2$  当且仅当存在  $T' \in Poss(T)$ ,而  $T'$  按强函数依赖方式适合  $S_1 \rightarrow S_2$ ,即  $tuples_D(T')$  中任两元组  $t_1', t_2'$ ,只要  $t_1'.S_1 = t_2'.S_1$ ,就有  $t_1'.S_2 = t_2'.S_2$ 。而另一方面  $T' \in Poss(T)$  是当且仅当存在一个替换  $f: (M(T)) \rightarrow T'$  使所有  $t \in tuples_D(M(T))$  都有  $t \subseteq f(t)$ ,且  $f(t)$  是完全树元组。设  $t_1' = f(t_1), t_2' = f(t_2), t_1', t_2' \in tuples_D(T')$ ,于是  $t_1$

$\subseteq t_1', t_2 \subseteq t_2'$ , 从而  $t_1, S_2 \subseteq t_1', S_2, t_2, S_2 \subseteq t_2', S_2$ 。注意到  $t_1', S_2, t_2'$  都是非空值, 所以  $t_1', S_2 = t_2', S_2$  当且仅当对每个  $p \in S_2$ , 或者  $t_1, p, t_2, p$  全为非空值, 且  $t_1, p = t_2, p$ , 这时  $t_1, P \cup t_2, p = t_1, p \neq \perp_{inc}$ ; 或者  $t_1, p = \perp$  而  $t_2, p$  非空, 这时  $t_1, p \cup t_2, p = t_2, p \neq \perp_{inc}$ ; 或者  $t_1, p$  非空而  $t_2, p = \perp$ , 这时  $t_1, p \cup t_2, p = t_1, p \neq \perp_{inc}$ ; 或者  $t_1, p = \perp$  且  $t_2, p = \perp$ , 这时  $t_1, p \cup t_2, p = \perp \neq \perp_{inc}$ 。只可能有这四种情况, 所以考虑这些当且仅当后得知,  $T$  适合  $S_1 \rightarrow S_2$ , 当且仅当  $tuples_D(M(T))$  中的任两个元组  $t_1, t_2$ , 只要  $t_1, S_1, t_2, S_1$  完全而且  $t_1, S_1 = t_2, S_1$ , 就有  $t_1, S_2 \cup t_2, S_2$  是相容的。

**定义 7** 给定一个 DTD  $D$ ,  $\Sigma$  是  $D$  上的函数依赖集合,  $S_1 \rightarrow S_2$  是  $D$  上的一个函数依赖, 若  $D$  的任意一棵 XML 树  $T$ ,  $T \triangleleft D$ , 只要适合  $\Sigma$  的每一个函数依赖, 则  $T$  就一定适合  $S_1 \rightarrow S_2$ , 则称  $\Sigma$  蕴含  $S_1 \rightarrow S_2$ , 记做  $\Sigma \vdash S_1 \rightarrow S_2$ 。

### 3 XML 弱函数依赖推理规则集

在不完全关系中对应的是 Lien-Atzeni 公理系统。下面将它推广到 XML 中。

A1:  $p_{y1}, \dots, p_{ym} \subseteq p_{x1}, \dots, p_{xn} \subseteq paths(D) \Rightarrow p_{x1}, \dots, p_{xn} \rightarrow p_{y1}, \dots, p_{ym}$

A2:  $p_{x1}, \dots, p_{xn} \rightarrow p_{y1}, \dots, p_{ym}, p_{z1}, \dots, p_{zk} \subseteq p_{w1}, \dots, p_{wl} \Rightarrow \{p_{x1}, \dots, p_{xn} \cup p_{w1}, \dots, p_{wl}\} \rightarrow \{p_{y1}, \dots, p_{ym} \cup p_{z1}, \dots, p_{zk}\}$

A3:  $p_{x1}, \dots, p_{xn} \rightarrow \{p_{y1}, \dots, p_{ym} \cup p_{z1}, \dots, p_{zk}\} \Rightarrow \{p_{x1}, \dots, p_{xn} \rightarrow p_{y1}, \dots, p_{ym}, p_{z1}, \dots, p_{xn} \rightarrow p_{z1}, \dots, p_{zk}\}$

A4:  $\{p_{x1}, \dots, p_{xn} \rightarrow p_{y1}, \dots, p_{ym}, p_{z1}, \dots, p_{xn} \rightarrow p_{z1}, \dots, p_{zk}\} \Rightarrow p_{x1}, \dots, p_{xn} \rightarrow \{p_{y1}, \dots, p_{ym} \cup p_{z1}, \dots, p_{zk}\}$

A5:  $q \rightarrow p$ , 如果  $Last(q) \in E$ ,  $p$  是  $q$  的前缀。

A6:  $Parent(q) \rightarrow q$ , 如果  $Last(q) \in A$ 。

A7:  $\{q, p_{x1}, \dots, p_{xn}\} \rightarrow p_{y1}, \dots, p_{ym} \Rightarrow \{q, q', p_{x1}, \dots, p_{xn}\} \rightarrow p_{y1}, \dots, p_{ym}$ , 其中  $q'$  为  $p_{x1}, \dots, p_{xn}, p_{y1}, \dots, p_{ym}$  的公共前缀。

A8:  $\{q, q', p_{x1}, \dots, p_{xn}\} \rightarrow p_{y1}, \dots, p_{ym} \Rightarrow \{q, p_{x1}, \dots, p_{xn}\} \rightarrow p_{y1}, \dots, p_{ym}$ , 其中  $q$  唯一决定  $q'$ 。

A9:  $\{q, p_{x1}, \dots, p_{xn}\} \rightarrow p_{y1}, \dots, p_{ym}$ , 其中  $q$  唯一决定  $p_{y1}, \dots, p_{ym}$ ,  $p_{x1}, \dots, p_{xn}$  为任意路径。

**定理 2** 推理规则 A1 ~ A9 是正确的。

**证明** A1, A2 以及 A5 ~ A9 的证明与文献[6]相似, 这里只证明 A3 和 A4。令  $S_x$  表示路径集合  $\{p_{x1}, \dots, p_{xn}\}$ ,  $S_y = \{p_{y1}, \dots, p_{ym}\}$ ,  $S_z = \{p_{z1}, \dots, p_{zk}\}$ 。

A3: 设 XML 树  $T$  适合  $S_x \rightarrow \{S_y \cup S_z\}$ , 于是有  $T' \in Poss(T)$ ,  $T'$  适合  $S_x \rightarrow \{S_y \cup S_z\}$ 。这样,  $tuples_D(T')$  中的任意两个树元组  $t_1', t_2'$ , 只要  $t_1', S_x = t_2', S_x$ , 就有  $t_1', \{S_y \cup S_z\} = t_2', \{S_y \cup S_z\}$ 。显然, 只要  $t_1', S_x = t_2', S_x$ , 就有  $t_1', S_y = t_2', S_y$  及  $t_1', S_z = t_2', S_z$ , 这样  $T'$  既是  $T$  的可能世界中的适合  $S_x \rightarrow S_y$  的 XML 树, 也是适合  $S_x \rightarrow S_z$  的树。

A4: 设  $T$  适合  $S_x \rightarrow S_y$  及  $S_x \rightarrow S_z$ , 于是可知在  $Poss(T)$  中有  $T', T''$ 。 $T'$  适合  $S_x \rightarrow S_y$ ,  $T''$  适合  $S_x \rightarrow S_z$ 。设  $M(T)$  到  $T'$  及  $T''$  的映射分别是  $f_1$  及  $f_2$ 。借助  $f_1, f_2$  来构造一个  $M(T)$  上的映射  $f$ , 对  $tuples_D(M(T))$  中的任一树元组  $t$ , 令

(1)  $f(t). [S_y - S_x] = f_1(t). [S_y - S_x]$

(2)  $f(t). [S_z - S_y - S_x] = f_2(t). [S_z - S_y - S_x]$

(3)  $f(t). S_x = \langle a_1, \dots, a_n \rangle$ , 这里, 当  $t. p_{xi}$  是非空值时,  $a_i = t. p_{xi}$ ; 当  $t. p_{xi}$  是空值时,  $a_i$  是任一个与已有的非空值都不同的一个非空值。

设  $t_1, t_2 \in tuples_D(M(T))$ , 由这个  $f$  的构造可以知道  $f(t_1). S_x = f(t_2). S_x$  当且仅当  $t_1. S_x = t_2. S_x$  且对每一个  $p_{xi} \in$

$S_x$ ,  $t_1. p_{xi} (= t_2. p_{xi})$  均为非空值,  $i = 1, \dots, n$ 。而且这时显然有  $f_1(t_1)$ ,  $S_x = f_1(t_2)$ ,  $S_x$  以及  $f_2(t_1)$ ,  $S_x = f_2(t_2)$ ,  $S_x$ 。由于  $T'$  及  $T''$  分别适合  $S_x \rightarrow S_y$  和  $S_x \rightarrow S_z$ , 所以有  $f_1(t_1)$ ,  $S_y = f_1(t_2)$ ,  $S_y$ , 从而  $f_1(t_1)$ ,  $[S_y - S_x] = f_1(t_2)$ ,  $[S_y - S_x]$ , 由于  $f_2(t_1)$ ,  $S_z = f_2(t_2)$ ,  $S_z$  进而  $f_2(t_1)$ ,  $[S_z - S_y - S_x] = f_2(t_2)$ ,  $[S_z - S_y - S_x]$ , 这样由  $f$  的构造可知,  $f(t_1)$ ,  $[S_y \cup S_z] = f(t_2)$ ,  $[S_y \cup S_z]$ 。现在令  $tuples_D(T''') = \{f(t) \mid t \in tuples_D(M(T))\}$ , 构造一棵完全树  $T''' = tree_D(tuples_D(T'''))$ , 显然  $T''' \in Poss(T)$ , 而且  $T'''$  适合  $S_x \rightarrow \{S_y \cup S_z\}$ 。于是  $T$  适合  $S_x \rightarrow \{S_y \cup S_z\}$ 。

**定理 3** 推理规则 A1 ~ A9 是完备的。

**证明** 设  $\Sigma$  是 DTD  $D$  上的函数依赖集合, 只需证明完备性的逆否命题“若从  $\Sigma$  出发用 A1 ~ A9 推不出函数依赖  $\sigma$ , 则  $\Sigma$  不蕴含  $\sigma$ ”。显然, 只要找到一棵 XML 树  $T$ ,  $T \triangleleft D$ , 它适合  $\Sigma$  中的每一个函数依赖, 但不适合  $\sigma$  就可以了。下面就根据  $\sigma$  给出  $T$ 。设  $\sigma$  是  $S_1 \rightarrow S_2$ ,  $S_1 \cup S_2 \subseteq paths(D)$ , 对于  $\sigma$  的左部  $S_1$ , 定义一个集合  $S_{1\Sigma}^+$ :

$$S_{1\Sigma}^+ = \{p \mid S_1 \rightarrow p \text{ 可从 } \Sigma \text{ 用 A1 ~ A9 推出}\}$$

$S_{1\Sigma}^+$  有以下性质:

(1) 若  $S_1 \rightarrow S_2$  可从  $\Sigma$  用 A1 ~ A9 推出, 则  $S_2 \subseteq S_{1\Sigma}^+$ 。

设  $S_2 = \{p_1, \dots, p_K\}$ , 反复应用 A3 可知,  $S_1 \rightarrow p_1, \dots, S_1 \rightarrow p_K$  都可被推出, 于是根据  $S_{1\Sigma}^+$  的定义可知,  $p_1, \dots, p_K$  都属于  $S_{1\Sigma}^+$ , 这样可以推出  $S_2 \subseteq S_{1\Sigma}^+$ 。

(2) 若  $S_1 \rightarrow S_2$  不能从  $\Sigma$  用 A1 ~ A9 推出, 则  $S_2 \not\subseteq S_{1\Sigma}^+$ 。

设  $S_2 = \{p_1, \dots, p_K\}$ , 若  $S_2 \subseteq S_{1\Sigma}^+$ , 则  $p_1, \dots, p_K$  都属于  $S_{1\Sigma}^+$ , 根据  $S_{1\Sigma}^+$  的定义可知,  $S_1 \rightarrow p_1, \dots, S_1 \rightarrow p_K$  都可从  $\Sigma$  用 A1 ~ A9 推出, 再反复利用 A4 可知  $S_1 \rightarrow S_2$  可推出, 这与前提矛盾, 所以  $S_2 \subseteq S_{1\Sigma}^+$ 。

(3) 若  $S_3 \subseteq S_1, S_3 \rightarrow S_4$  可从  $\Sigma$  用 A1 ~ A9 推出, 则  $S_4 \subseteq S_{1\Sigma}^+$ 。

由于  $S_3 \rightarrow S_4$  可推出, 而  $S_1 \supseteq \emptyset$ , 所以由 A2 可知可推出  $S_3 \cup S_1 \rightarrow S_4 \cup \emptyset$ 。因为  $S_3 \subseteq S_1$ , 所以可推出  $S_1 \rightarrow S_4$ , 由性质(1) 知  $S_4 \subseteq S_{1\Sigma}^+$ 。

根据  $S_{1\Sigma}^+$  及以上三个性质, 可以构造出一个树  $T$  的最小扩展树  $T'$ , 若  $S_{1\Sigma}^+ - S_1$  有  $K$  个元素, 则这个  $T'$  只取以下集合中的值:  $\{a_1, \dots, a_K, b, c_1, \dots, c_{2K}\}$ 。这个集合中的元素彼此互不相等。这个  $T'$  有  $2K$  个树元组  $t_1, \dots, t_{2K}$ 。具体取值如下:

对于  $i = 1$  到  $K$ :

(1) 对  $S_1$  中的每个路径  $p$ , 令  $t_{2i-1}. p = a_i, t_{2i}. p = a_i$ 。

(2) 对  $S_{1\Sigma}^+ - S_1$  中的每个路径  $p_j$ , 令  $t_{2i-1}. p_j = b (j = i)$  或  $t_{2i-1}. p_j = \perp (j \neq i); t_{2i}. p_j = \perp$ 。

(3) 对  $paths(D) - S_{1\Sigma}^+$  中的每个路径  $p$ , 令  $t_{2i-1}. p = c_{2i-1}, t_{2i}. p = c_{2i}$ 。

以  $S_{1\Sigma}^+ - S_1 = \{p_1, p_2, p_3\}$  为例, 由假设可知, 在此情况下  $T'$  有 6 个树元组, 其值如下:

$S_1$	$S_{1\Sigma}^+ - S_1$	$Paths(D) - S_{1\Sigma}^+$
$t_1 (a_1 \dots a_1 \ b, \perp, \perp)$		$c_1 \dots c_1$
$t_2 (a_1 \dots a_1 \ \perp, \perp, \perp)$		$c_2 \dots c_2$
$t_3 (a_2 \dots a_2 \ \perp, b, \perp)$		$c_3 \dots c_3$
$t_4 (a_2 \dots a_2 \ \perp, \perp, \perp)$		$c_4 \dots c_4$
$t_5 (a_3 \dots a_3 \ \perp, \perp, b)$		$c_5 \dots c_5$
$t_6 (a_3 \dots a_3 \ \perp, \perp, \perp)$		$c_6 \dots c_6$

下面, 利用以上构造的树及其产生的元组, 证明这个  $T'$

适合 $\Sigma$ 中的每一个函数依赖。设 $P \rightarrow Q$ 是 $\Sigma$ 中的一个函数依赖,根据 $P$ 分两种情况讨论:

(1)  $P$ 包含 $S_1$ 以外的路径,由于这时 $T$ 的可能世界 $Poss(T)$ 中存在一个完全关系 $T''$ ,它是将 $tuples_D(T')$ 中树元组 $t_i$ 上的所有 $\perp$ 用 $c_i$ 代换后得到。 $T''$ 的任两个树元组在 $P$ 上的取值都互不相同,所以 $T''$ 适合 $P \rightarrow Q$ ,这样 $T$ 适合 $P \rightarrow Q$ ;

(2)  $P \subseteq S_1$ ,由于此时必有 $Q \subseteq S_{1\Sigma}^+$ (上述性质3),这时 $T$ 的可能世界 $Poss(T)$ 中也存在一个完全树 $T'''$ ,它是将 $tuples_D(T')$ 中树元组中的所有 $\perp$ 用 $b$ 替换后得到, $T'''$ 的任两个元组在 $S_{1\Sigma}^+ - S_1$ 上的取值都是一样的,而 $Q \subseteq S_{1\Sigma}^+$ ,所以在 $T'''$ 中,只要两个树元组在 $P$ 上相等(实际上是 $t_{2i-1}$ 与 $t_{2i}, 1 \leq i \leq K$ ),则在 $Q$ 上的取值也一定相等。所以 $T'''$ 适合 $P \rightarrow Q$ ,这样 $T$ 也适合 $P \rightarrow Q$ 。

由于 $P \rightarrow Q$ 是 $\Sigma$ 中的任意函数依赖,所以 $T$ 适合 $\Sigma$ 中的每一个函数依赖。

最后证明 $T$ 不适合 $\sigma$ 。由于 $\sigma$ 不能从 $\Sigma$ 用公理A1~A9推出,所以根据上述性质(2)可知 $S_2 \not\subseteq S_{1\Sigma}^+$ ,这样 $S_2$ 中必然有 $paths(D) - S_{1\Sigma}^+$ 中的路径。然而在 $paths(D) - S_{1\Sigma}^+$ 的路径上, $tuples_D(T')$ 中的每个树元组都是非空值,而且互不相等,所以不论 $T'$ 中的空值如何被代换, $tuples_D(T')$ 中的任何两个元组在 $S_2$ 上也不会相等,所以 $T$ 的可能世界 $Poss(T)$ 中不存在适合 $S_1 \rightarrow S_2$ 的完全树,所以 $T$ 不适合 $S_1 \rightarrow S_2$ 。

(上接第1221页)

离,使得测试样本归入所属类的可能性增大,以提高其识别率。从理论上来说,切线向量个数取的越多,测试样本与其模板之间切线距离越小,识别率越高,但事实上,切线向量个数取的越多,测试样本与所属类模板间距减小的同时,与其他类别之间的距离也将减小,随着切线向量个数的增加,错误率可能反而会增加。在该细分类试验中,选取30个切线向量左右比较合适。

表1 不同切线向量个数切线距离的细分类效果

切线向量个数	10个	20个	30个	40个
粗分识别率(%)	91.5	91.5	91.5	91.5
细分识别率(%)	96.2	98.3	99.2	98.7
最终识别率(%)	88.1	90.0	90.9	90.3
每百字识别时间/s	14.1	17.2	25.3	34.5

为了进一步说明基于切线距离的SVD分解选取切线向量方法在大字符集手写体汉字识别细分类中的有效性。我们对该方法与采用最小距离,欧式距离,最小相关性距离作为试验细分类器所得实验结果进行了比较分析。

表2 采用切线距离分类器与其他分类器比较结果

细分类器	欧式距离	最小距离	相关性距离	30个向量切线距离
粗分识别率(%)	91.5	91.5	91.5	91.5
细分识别率(%)	73.6	72.2	73.9	99.2
总最终识别率(%)	67.3	66.1	67.6	90.9
每百字识别时间/s	4.62	4.77	4.61	25.3

可以看出,采用切线距离这样对汉字形变不敏感的距离,对汉字识别率的提高是显著的。在选取轮廓方向特征作为细分特征,30个SVD分解切线向量的前提下,细分类识别率达到了99.2%,取得了非常优异的成绩,比采用最小相关性距离分类器识别率提高了25.3%。我们对每种汉字建立了一

这样就证明了从用A1~A9公理系统推不出的函数依赖,就一定不被 $\Sigma$ 蕴含,也即证明了A1~A9的完备性。

#### 4 结语

XML函数依赖在XML数据库研究中占有重要地位。本文将空值关系模型中的理论推广到XML中,给出了XML弱函数依赖的概念,并给出一组正确且完备的推理规则集,下一步的工作是在本文的基础上对弱函数依赖的可加性进行研究,进一步完善XML数据库理论。

#### 参考文献:

- VINCENT MW, LIU J, LIU C. Strong functional dependencies and a redundancy free normal form for XML[J]. ACM Transactions on database systems, 2004, 29(3): 445~462.
- ARENAS M, LIBKIN L. A normal form for XML documents[J]. ACM Transactions on Database Systems, 2004, 29(1): 195~232.
- 谈子敬, 庞引明, 施伯乐. XML上的函数依赖推理[J]. 软件学报, 2003, 14(9): 1564~1570.
- LEVENE M, LOIZOU G. The Additivity Problem for Functional Dependencies in incomplete Relations[C]. Acta Informatica, 1997, 34(2): 135~149.
- 马垣. 非经典关系数据库理论[M]. 北京: 清华大学出版社, 2005.70~78.
- 胡小明. XML函数依赖的推理规则与蕴涵问题研究[D]. 秦皇岛: 燕山大学, 2004.

个模板,如果先采用聚类的方法对汉字进行聚类,然后为每种汉字建立多个模板,汉字识别的效果可能更好。但是也看到,由于需要通过迭代来计算模板参数的问题,采用切线距离作为分类器识别速度较慢,所以切线距离一般仅限于做汉字识别细分类器。

#### 4 结语

本文提出,采用基于切线距离的SVD分解方法来进行大字符集脱机手写体汉字识别系统细分类,在候选集为50,采用 $8 \times 8$ 网格划分轮廓方向特征为细分特征,选取30个切线向量的情况下,得到了99.2%的细分类结果,比采用欧式距离等有显著改善。但采用该方法进行识别,识别速度较慢,如果能够提高该方法的识别速度,将能够使该分类器性能得到进一步提高。

#### 参考文献:

- RICHARD OD, PETER EH, DAVID GS. 模式分类[M]. 第2版. 北京: 机械工业出版社, 2002.
- 张祈中. 汉字识别技术[M]. 北京: 清华大学出版社, 1992.
- SIMARD PY, LECUN Y, DENKER J, et al. Transformation Invariance in Pattern Recognition-tangent Distance and Tangent Propagation[A]. Lecture Notes in Computer Science1524[C]. Springer, Heidelberg, 1998. 239~274.
- SIMARD P, LE CUN Y, DENKER J, et al. An efficient algorithm for learning invariances in adaptive classifiers[A]. Proceedings 11th IAPR International Conference on Pattern Recognition[C]. 1992. 651~655.
- 杨程云,宣国荣. 切线距离中选取切线向量的SVD方法[J]. 计算机应用, 2003, 23(6).
- 谭锐,宣国荣. 切线距离在印刷体数字识别中的应用[J]. 微型电脑应用, 2002, 18(12).
- 金连文,高学. 几种手写体汉字网格方向特征提取法的比较研究[J]. 计算机应用研究, 2004, 21(11): 38~40.