

支持更新的连续不确定 XML 编码方案

张晓琳, 吕 庆, 刘立新, 郑春红

(内蒙古科技大学 信息工程学院, 内蒙古 包头 014010)

摘 要: 提出基于 p-文档模型的连续不确定 XML 的 CUPE(continuous uncertain prime encoding)编码方案. 该方案重复利用小素数编码, 根据码值之间素数因子的包含关系能够快速精准地判定出 XML 树中任意两个节点的结构关系, 并且可以支持文档更新. 实验结果表明, 相对区间编码和纯素数编码, CUPE 是一种快速有效的编码方案.

关键词: 连续不确定; 素数编码; 结构关系; 文档更新

中图分类号: TP392

文献标识码: A

文章编号: 1000-7180(2013)01-0157-04

Encoding Scheme Supporting Update of Continuous Uncertain XML

ZHANG Xiao-lin, LV Qing, LIU Li-xin, ZHENG Chun-hong

(School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, China)

Abstract: A new encoding schema based on existing p-document model is proposed which is also for continuous uncertain XML, called CUPE. Any two nodes' relation can be judged by this schema fast and accurately through taking advantage of the prime factors' inclusion relation between them, and this schema can also support the update of the document. At last the results of experiment shows that CUPE is a fast and effective encoding schema in contrast to zone coding schema and prime coding schema.

Key words: continuous uncertain; prime encoding; structure relation; document update

1 引言

随着数据的处理和采集技术的提高, 对于不确定性数据的关注也越来越多, 如何对这些数据进行管理成为研究热点. XML 文档以其可扩展性高、自描述性好与灵活性高的优点, 已成为目前数据交换和表现的主流形式, 特别适用于信息抽取与数据集成等可能产生大量连续不确定信息的应用中. 这些数据可以在 XML 文档中表示, 形成连续不确定 XML.

近些年来, 提出了多种关于 XML 的索引与查询技术大都基于某种编码方案的. 大部分的 XML 编码技术都是对 XML 树中的每一个节点分配一个唯一的数字码值, 通过任意两个节点的码值能够直接判断出两个节点之间的结构关系.

本文基于 p-文档模型^[1]提出一种新的连续不

确定 XML 编码方案, 其具有以下优点: (1) 利用节点编码的素数因子包含关系判断文档中节点之间的结构关系, 简单易行; (2) 重复利用小素数进行编码, 解决了素数编码增长过快, 大素数易溢出, 浪费存储空间的问题; (3) 编码方案具有动态性, 能高效地支持文档的更新操作; 四、支持对连续节点的编码, 对节点的概率特性计算方便快捷, 利用编码即可实现.

2 相关研究

目前已经提出了多种 XML 编码方案, 而对于连续不确定 XML 的编码技术研究还很少. 根据编码原理的不同, 可以分为区间编码、前缀编码以及素数编码三大类.

2.1 区间编码机制

早期的对于 XML 文档的编码研究都是基于区间编码^[2], 利用区间编码之间的嵌套关系可以快速

收稿日期: 2012-05-09; 修回日期: 2012-06-15

基金项目: 国家自然科学基金项目(61163015); 内蒙古自然科学基金重点项目(20080404Zd21)

判断两节点的结构关系,但编码的效率很低,需要两次遍历文档树,当文档发生变化时,更新代价很高.文献[3]是第一个关于不确定 XML 区间编码方案的研究,目前这种编码方案只是支持离散的不确定性文档,并没有涉及到包含连续节点的不确定 XML 文档.

2.2 前缀编码机制

前缀编码^[4]是一种基于路径的编码方案,文档树中的节点编码以父亲节点的编码作为前缀,编码长度是 XML 文档树的深度,判断节点 a 是否是节点 b 的子孙节点,只需判断 a 的编码是否是 b 的前缀.文献[5]首次提出不确定 XML 的前缀编码方案,在 p-文档基础上将分布式节点作为路径存入节点的编码中.但前缀编码不支持文档的频繁更新,并且文档树深度过大时编码的长度随之变大,浪费存储空间.

2.3 素数编码机制

文献[6]是利用素数对 XML 文档树进行编码来解决更新的问题.根据码值间的整除关系可以判断任意两个节点的结构关系,这种方法从理论上解决了 XML 文档更新的问题,但每一个节点的编码都是父亲节点的编码乘上一个大素数得到,素数的大小增长很快,当文档树深度增加或者节点过多时很容易造成码值数据的溢出,且浪费存储空间.

3 CUPE 编码

3.1 编码方案

每一个节点的编码由一个四元组构成 $\langle \text{prime}, \text{primeMulti}, \text{pro}, \text{path} \rangle$, prime 代表的是为节点分配的素数码值,编码过程中按层遍历文档树的每一个节点.如果节点 u 是 v 的父亲节点,则 v 的 primeMulti 是节点 u 的 prime 与 primeMulti 的乘积,可以利用 primeMulti 来判定结构关系. pro 代表节点概率特性,标识其存在的概率值.若节点是普通类型的节点则 pro 的值为 1;若节点是不确定性节点则 pro 为一个概率数值 $P(0 < P < 1)$. path 代表从根节点到自身节点的路径上的分布节点的类型, p-文档中常用的分布节点有独立类型(ind)的节点、互斥类型(mux)的节点和连续型(cont)节点, cont 节点只出现在叶子节点上(如图 1 中的节点 $N(15,3)$),表示路径时用 i 代表 ind 节点, m 代表 mux 节点.如果路径上不存在分布类型的节点则 $\text{path} = o$, o 代表 ordinary 即普通类型的节点.

图 1 是一个 p-文档树.表示的是传感器测得的

某实验室的两天内的用电量变化情况,该文档树由两种类型的节点构成,分别为普通类型节点和分布类型节点,分布式节点又包含了 ind、mux 和 cont 节点.对此文档树进行 CUPE 编码后如图 2 所示.

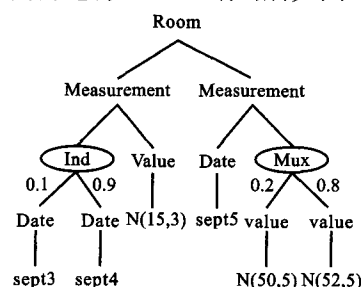


图 1 $\text{PrXML}_{\{\text{ind}, \text{mux}, \text{cont}\}}$

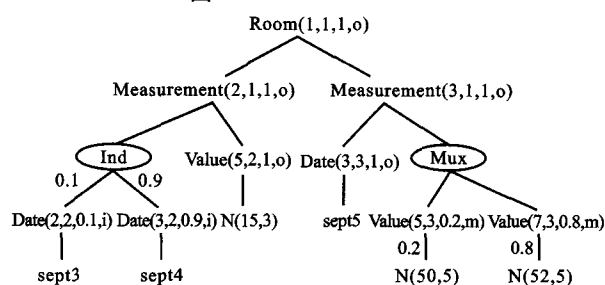


图 2 CUPE 编码树

3.2 编码算法

算法 1: CUPE 编码算法

输入: 具有连续不确定性节点的 XML 文档树 T

输出: 具有编码属性的 CUPE 编码树

CUPE_Coding_PXML(Document T)

Begin;

Element root = T.getRootElement();

//得到文档的根节点

root.addAttribute("code", "1.1.1.n");

//为根节点分配编码值 1.1.1.n

for (Iterator it = root.elementIterator(); it.hasNext();)

//遍历所有的节点

{
Element element = (Element) iter.next();

String parentCode = element.getParent().attributeValue("code");

//得到父亲节点的编码值

String parentString [] = parentCode.split(",");

//字符串以“,”分割,取前面的字符(取父亲节点的编码)

String selfAttribute = element.attributeVal-

```

ue("dist");
//得到自身的分布类型
int prime = Integer.parseInt(parentString[0]);
//得到节点编码的 prime 值
int primeMulti = Integer.parseInt(parentString[0]) *
Integer.parseInt(parentString[1]);
//得到节点的 primeMulti 的值
i=prime;
element.addAttribute("code",i+","+primeMulti+","+1,"+selfAttribute");
i=(Prime.producePrim(i))+1; //令 i 为刚产生的素数自加 1,为产生下一个素数做准备
}End;

```

3.3 结构关系判定

定理 1 (算术基本定理)任意一个大于 1 的自然数 N 都可以唯一分解成有限个素数的乘积,这里的分解称为 N 的标准分解式. 算术基本定理由两部分内容组成,第一部分是分解的存在性,每一个大于 1 的自然数 N 都能分解成素数因子的乘积;第二部分内容是分解的唯一性,如果不考虑素数因子排列的顺序,正整数分解为因子乘积的方式是唯一的.

图 3 为一棵 XML 文档 CUPE 简易编码树,为了方便表示省去编码的 pro 和 path 值.

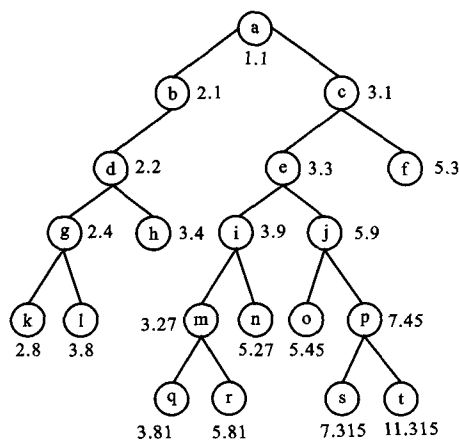


图 3 CUPE 简易编码树

兄弟关系的判定:若节点 A 、 B 具有兄弟关系,则 A . primeMulti = B . primeMulti. 例如:图 3 中 i (3.9)和 j (5.9), m (3.27)和 n (5.27)等.

父子关系判定:若节点 A 是 B 的父亲,则 B . primeMulti = A . prime * A . primeMulti. 例如:图 3 中的 (3.3)和 (3.9), (3.9)和 (5.27)作为 A 、 B 节点,其中 $9=3*3$, $27=3*9$.

祖先子孙关系判定:若判断节点 A 、 B 是否存在祖先后代关系,则利用定理 1 把 B 的 primeMulti 分解成素数因子的乘积,第一个素数因子从小于等于 prime 的数开始,再将这些素数因子转化为字符串 i ,然后把 A 的 prime 和 primeMulti 的素数因子转化为字符串 j . 最后判断 i 是否包含且以 j 结束. 例如:图 3 中节点 j : (5,9)、 s : (7,315), 其中对于 s 来说,primeMulti 为 315, $315=7*5*3*3$, 对应字符串为“7533”;对于 j 来说对应字符串为“533”. 显然满足以上的结构关系条件,证明 j 是 s 的祖先节点.

3.4 支持更新

对文档树中节点更新操作,包括删除、修改和插入操作. 删除和修改操作不会影响文档树结构,因此也不会影响节点的编码,所以更新主要针对插入操作.

XML 数据一般可以分为两种:模式独立的数据和基于模式的数据. 模式独立的数据不受任何限制,对于这种更新操作可以发生在树中的任何位置. 而对于基于模式的数据要受约于模式信息(如 DTD 或者 XML Schema),对于插入的节点或是子树有严格的要求,插入后整个文档仍要符合模式的定义. 对于一棵具有 n 个节点的文档树,若其模式独立则插入操作的插入位置会有 $2n-1$ 个;对于基于模式的文档其插入操作也会很多,如果选择的位置不同,那么区间编码需要改变的节点个数也不同. 本文处理的文档主要是针对连续不确定的 XML,同样也分为这两种类型的数据,在对它的更新处理中,区间编码需要考虑数据模式的约束,在这两种模式下可能某一个节点的插入会影响它所在单枝或是整棵子树的所有节点的编码值. CUPE 编码方案不会存在这样的问题. 不管是什么类型的 XML 数据,当插入一个新的节点时为其分配一个素数值 prime,其编码中的 primeMulti 直接等于父亲节点的 primeMulti 与 prime 的乘积,不会影响其他任何节点的编码值.

4 实验结果与分析

实验共有三组测试,针对 CUPE 编码方案、概率区间编码方案和素数编码方案在时间性能、空间性能和更新效率上进行比较. 每组实验均重复 10 次,得到的实验数据采用去掉最大值和最小值,取平均值的方法记录整理.

第一组测试从编码时间的角度出发,把 CUPE 编码方案与概率区间编码方案和素数编码进行比较. 第二组测试从三种编码占用的存储空间进行比

较. 第三组测试是从三种方案对更新的支持进行比较.

实验结果表明, 从时间性能和更新性能上 CUPE 编码方案要优于现有的不确定 XML 区间编码, 从空间性能上对比又优于纯素数编码, 因此这种编码方案是一种高效的编码方案.

5 结束语

文中提出的基于 p-文档模型的连续不确定 XML 的 CUPE 编码处理技术. 理论分析和实验测试表明: 该方法能够实现节点的结构关系快速判断、可以有效的支持文档更新. 进一步的工作是对多维连续分布的编码方案做更深入的研究.

参考文献:

- [1] Abiteboul S, Chan T.-H. H, Kharlamov E, Nutt W and Senellart P. Aggregate queries for discrete and continuous probabilistic XML[C]//Proceedings of the 13th International Conference on Database Theory, 2010:50-61.
- [2] Zhang C, Naughton J F, DeWitt D J, Luo Q, Lohman G M. On supporting containment queries in relational database management systems[C]//Proceeding of the 27th ACM SIGMOD, Santa Barbara: IEEE, 2001.
- [3] Yawen Li, Guoren Wang, Junchang Xin, Ende Zhang, Zeling Qiu. Holistically Twig Matching in Probabilistic XML[C]//Proceedings of the 25th International Conference on Data Engineering(ICDE2009). Los Alamitos:IEEE Press, 2009:1649-1656.
- [4] Tatarnov I, Viglas S D, Beyer K, Shanmugasundaram J, Shekita E, Zhang C. Storing and quering ordered XML using a relational database system[C] //Proceeding of the 28th ACM SIGMOD. Madison: IEEE, 2002:204-215.
- [5] NING Bo, LIU Cheng-fei, YU J X, et al. Matching top-k answers of twig patterns in probabilistic XML [C]//Proc of the 15th International Conference on Database Systems for Advanced Applications, Tsukuba, Japan:IEEE, 2010:125-139.
- [6] Wu X, Lee M L, Hsu W. A prime number labeling scheme for dynamic ordered XML trees[C]//Proceeding of the 2004 Intl Conf on Data Engineering. Boston, Massachusetts, USA:IEEE, 2004:66-77.

作者简介:



张晓琳 女,(1966—),博士,教授. 研究方向为数据库理论与技术,信息安全.

吕庆 男,(1986—),硕士研究生. 研究方向为数据库理论.

刘立新 女,(1983—),硕士研究生,助教. 研究方向为信息安全,不确定数据管理.

郑春红 女,(1987—),硕士研究生. 研究方向为数据库理论研究.