

一种基于完整性约束的路径表达式的查询优化策略^{*}

张剑妹^{1,2} 陶世群¹ 段洪秀¹

(山西大学计算机与信息技术学院 太原 030006)¹ (长治学院 长治 046011)²

摘要 利用路径表达式导航 XML 查询是 XML 查询语言的共同特点。目前对 XML 路径表达式的计算有两种方法:一种是基于树遍历的方法,一种是路径连接方法。在路径连接方法中路径表达式的计算效率很大程度上依赖于路径表达式的长度。在对 XML 模式反映的完整性约束研究的基础上,本文提出了排他性包含约束的概念;给出了利用排他性包含约束缩短路径表达式的策略和算法,从而降低了路径连接的代价。通过分析比较,这种路径缩短策略是有效可行的。

关键词 XML, 路径表达式, 结构连接, 完整性约束, 排他性包含约束, 优化策略

A Query Optimizing Strategy of Path Expression Based on Integrity Constraints

ZHANG Jian-Mei^{1,2} TAO Shi-Qun¹ DUAN Hong-Xiu¹

(College of Computer and Information Technology, Shanxi University, Taiyuan 030006)¹ (Changzhi College, Changzhi 046011)²

Abstract Navigating XML queries by path expression is the common characteristic of XML query languages. At present, there are two kinds of path expression evaluating methods; one is based on tree traverse and the other is path join. The computing efficiency of path expression mostly depends on its size in the methods of path join. Based on the study of integrity constraints depicted by XML schemas, the concept of exclusive inclusion constraint is proposed. Meanwhile the paper presents the path shorten strategy and arithmetic applying the concept. The path shorten strategy reduces the path join cost. Analysis and comparison show that the path shorten strategy is efficient and feasible.

Keywords XML, Path expression, Structural join, Integrity constraint, Exclusive inclusion constraint, Optimizing Strategy

1 引言

随着可扩展标记语言 XML 迅速成为 Web 上数据表示和数据交换的标准,XML 数据的查询变得越来越重要。在这种情况下,许多 XML 查询语言如 Lorel、XPath、XML-QL、XQL 和 XQuery 等应运而生,这些查询语言的共同特点是利用路径表达式来导航 XML 文档的查询并返回指定路径所能访问到的节点集。因此,如何高效地计算 XML 路径表达式越来越受到学术界的关注。目前对 XML 路径表达式的计算有两种方法:一种是基于树遍历的方法^[1,2]。为了降低数据查询的代价,在这种访问方法中通常建立有效的路径索引以缩小节点的搜索范围。另一种方法是路径连接方法^[3],这种方法把复杂的路径表达式分解成基本路径表达式,然后计算每个基本路径表达式的结果,再将基本路径表达式的结果连接起来,这种操作叫结构连接。类似于关系数据库中的连接运算,结构连接运算的代价很高,在这种方法中,路径表达式的优化在于开发高效的结构连接算法和结构连接顺序的选择;同时,路径表达式的长度直接影响路径表达式的计算性能,因此,对路径表达式进行最小化处理也是一个极其重要的优化策略。

最先研究路径表达式最小化问题是文[4,5],文[4]中只对不包括祖先-后代边“//”的简单路径表达式进行最小化,而文[5]研究了不包含 * 的路径表达式的最小化问题。文[6]对包含全部操作符{/,//,[,*,}的路径表达式的最小化进行了研究。尽管文[4~6]在不同的操作子集上使用不同的方法对

路径表达式进行最小化处理,但其本质上都是要删除路径表达式中冗余的条件表达式。和文[4~6]不同,针对 XPath 查询语言,文[7]通过引入一个 layer axis 消除路径表达式中的非冗余的通配符步,从而有效地缩短了有通配符的路径表达式。文[8]提出了两条路径缩短策略:缩短路径法和补路径法。缩短路径法用等价的相对路径取代绝对路径,缩短路径表达式本身,从而降低查询代价;补路径法则是使用和查询路径互补的代价低的等价路径替代那些用户书写的复杂的、代价高的查询路径表达式。

本文的中心思想是试图将较长的由“/”(表示元素间的父-子关系)连接的路径转换为由“//”(表示元素间的祖先-后代关系)连接的路径。一般情况下,由于路径操作符“//”的不确定性,在基于树遍历的方法中这种操作符的计算代价很高,在路径表达式中应设法避免这种操作符的出现,因此很多文献中都力图将“//”转化为“/”^[8]。结构连接的本质问题是根据元素的编码来确定元素之间的父-子关系或祖先-后代关系。如果使用区域编码,确定两个元素 e_1 ($start_1, end_1, level_1$) 和 e_2 ($start_2, end_2, level_2$) 的祖先-后代关系只需要判断 $start_1 < start_2 \square end_2 < end_1$;确定它们的父-子关系则需判断 $start_1 < start_2 \square end_2 < end_1 \square level_1 + 1 = level_2$ ^[9]。由此可见在路径连接方法中路径“A//B”的计算代价和“A/B”的计算代价是相同的,如果能把比较长的父-子关系的路径缩短为等价的祖先-后代关系的路径表达式,这样一方面减少了连接次数,另一方面在查询计划生成阶段也减少了可选的连接计划,

^{*} 本文得到了国家自然科学基金(No. 60275019)资助。张剑妹 讲师,博士研究生,主要研究方向为 XML 数据库技术。陶世群 教授,博士生导师,主要研究方向为数据库理论与技术。段洪秀 硕士研究生,主要研究方向为 XML 数据库技术。

从而大大地优化路径表达式的查询。基于这种考虑本文对 XML 模式隐含的完整性约束进行了研究,提出排他性包含约束的概念,并利用排他性包含约束将父-子关系的路径缩短为等价的祖先-后代关系的路径以降低路径连接的计算代价。

2 相关概念

2.1 数据模型和数据模式

一个 XML 文档是一个标签在节点上的树,如果考虑元素间的引用关系,则 XML 文档的基本结构为图。每个节点与一个元素、属性或值(文本)相对应,边表示元素和子元素之间的嵌套关系。XML 文档树用 $D=(r_d, V_d, E_d, \tau_d, \Sigma_d)$ 表示,其中树的根节点为 r_d , V_d 是节点的集合, E_d 是节点间边的集合, Σ_d 是元素类型的集合,每个元素都有一个类型,它们之间的映射关系由 τ_d 表示。图 1 所示为出版数据的 XML

文档树片断,该 XML 文档树没有区分元素与属性。XML 文档的数据模式可以定义为一个有向图,图中的节点表示元素类型,边表示元素类型间的嵌套包含关系。图中唯一的一个没有入边的节点被称为图的根。数据模式图用 $G=(r_s, V_s, E_s)$ 表示, r_s 表示的根, V_s 是图中的顶点的集合,模式中每个顶点对应着文档中的一种类型的顶点, E_s 是边的集合。图 2 给出了出版文档示例的数据模式图。下文用到的符号及其意义如下:

P 表示查询路径表达式, E 表示元素类型,不同的元素类型用带下标的大写字母 E 表示; e 表示文档中的元素实例; $\tau(e)$ 表示元素实例 e 的类型; $Paths(E_i; E_j)$ 表示从元素类型为 E_i 节点开始访问类型为 E_j 的元素实例的路径的集合; $P(e_i; e_j)$ 表示从元素 e_i 开始访问 e_j 的路径。

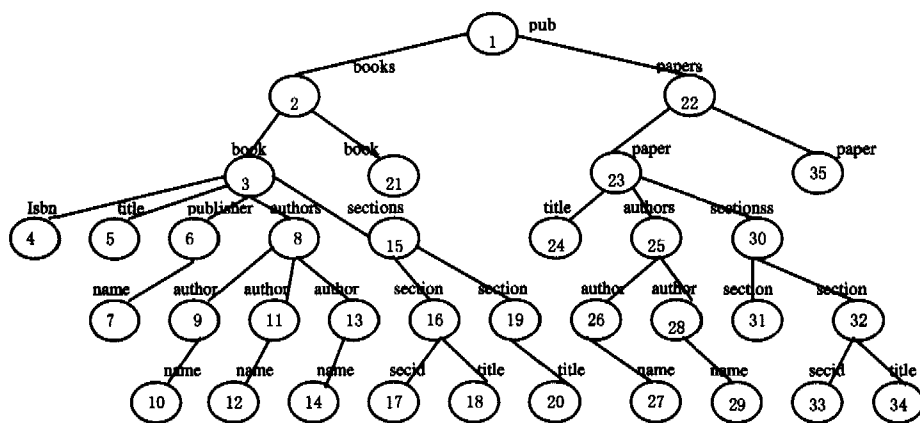


图 1 XML 出版文档示例

2.2 XML 路径表达式

在 XML 数据查询中通常用路径表达式来导航 XML 查询,路径表达式是由“/”或“//”或“[]”分隔开的一系列元素标签的集合。当路径表达式中某元素名无关紧要或未知是可以通配符“*”替代。其中“/”表示元素之间的关系是直接包含关系(父子关系);“//”表示元素之间的间接包含关系(祖先-后代关系)。

3 XML 数据的完整性约束

XML 模式是伴随着 XML1.0 规范的制订而推出的,目前有多种模式规范,其中最常用的是 XML DATA 和 XML Schema。不管哪种模式规范都为 XML 数据提供一定程度的完整性约束。一般情况下,XML 模式结构所反映的完整性约束有三类:包含约束,键约束和逆反约束(inverse constraints)。本文中我们把 XML 文档模式提供的包含约束分为两类:必需性包含约束和排他性包含约束。下面分别给出它们的定义。

定义 1 必需性包含:给定符合模式 G 的文档 D , $E_i, E_j \sqsubseteq V_s$, 如果 $(\forall e_i \in V_d)(\tau(e_i) = E_i \rightarrow (\exists e_j)(\tau(e_j) = E_j \wedge e_j \text{ is the descendant of } e_i))$, 则 E_i 必需包含 E_j , 记作 $E_i \Rightarrow E_j$ 。在 DTD 中必需性包含由元素名后的“#REQUIRED”和“+”表示。如图 2 所示,由于“+”表示有一个或多个该类型的元素出现,因此 $authors \Rightarrow author$ 成立;但“*”表示有零个或多个该类型的元素出现,所以 $papers \Rightarrow paper$ 不成立。注意,模式图 2 中仅仅是为了说明我们的定义表示出了一部分必需性包含。

定义 2 排他性包含:给定符合模式 G 的文档 D , 设 $E_i, E_j \sqsubseteq V_s$, $e_i, e_j \sqsubseteq V_d$, $\tau(e_i) \sqsubseteq E_i$, $\tau(e_j) \sqsubseteq E_j$, 如果 $\forall P_i(e_i; e_j)$, $P_j(e_i; e_j) \in Paths(E_i; E_j) \rightarrow P_i(e_i; e_j) = P_j(e_i; e_j)$, 则 E_i 排他包含 E_j , 记作 $E_i \alpha E_j$ 。例如:在图 2 中 $authors \alpha name$ 成立,但 $book \alpha name$ 不成立。

定理 1 如果 $(E_1 \alpha E_3) \wedge (E_2 \text{ is an ancestor of } E_3 \wedge E_2 \text{ is a descendant of } E_1)$, 则一定有 $E_1 \alpha E_2$ 且 $E_2 \alpha E_3$ 成立。

证明: 根据排他性包含的定义,如果 E_1 排他包含 E_3 , 那么从元素类型 E_1 开始访问元素类型 E_3 的实例只有一条唯一的路径,又因为 E_2 是 E_3 的祖先并且是 E_1 的后代,所以 E_2 必然在从 E_1 到 E_3 的路径上,这样可以得到从 E_1 开始访问 E_2 和从 E_2 开始访问 E_3 的路径也是唯一的。

从排他性包含的定义可见,在模式图中若从某节点到另一节点只有唯一的一条路径,则这两个节点就满足排他性包含关系,将其记入一个排他包含关系表。在该表中我们只记录路径步大于 2 的唯一路径最长的非根节点的排他包含关系。

4 最小化路径表达式

4.1 路径表达式的等价性

对路径表达式进行最小化处理必须保证路径表达式的等价性,也就是要保证处理后的路径表达式与原路径表达式的查询能力相等。用 $Q(T_d, P)$ 表示路径表达式 P 在文档 D 中的查询结果,路径表达式等价的形式化定义如下:

定义 3 如果 $(\forall D)(\forall e_i \in Q(D, P_1) \rightarrow e_i \in Q(D, P_2))$, 则 $P_1 \subseteq P_2$ 。

定义4 如果 $P_1 \subseteq P_2$ 并且 $P_2 \subseteq P_1$, 则 P_1 和 P_2 等价, 记作 $P_1 = P_2$ 。

4.2 路径表达式的最小化技术

本文的主要目的是把用户书写的复杂的比较长的路径表达式, 根据排他性包含约束缩短成等价的较短的路径表达式, 以此减少结构连接次数。这样一方面减小了结构连接的代价, 另一方面在查询计划生成阶段也大大地减少可选的连接计划, 从而优化了 XML 路径表达式的查询。定理2及其推论描述了我们的路径表达式的缩短策略。

定理2 如果 $E_1 a E_3$, 那么 $(E_1/E_2/E_3) = (E_1//E_3)$ 一定成立。

证明: 使用反证法很容易证明这个定理的正确性。假定 $E_1/E_2/E_3$ 与 $E_1//E_3$ 不等价, 则有 $(E_1/E_2/E_3) \not\subseteq (E_1//E_3)$, 或者 $(E_1//E_3) \not\subseteq (E_1/E_2/E_3)$ 成立。我们首先证明第一个假设不成立:

假设 $(E_1/E_2/E_3) \not\subseteq (E_1//E_3)$, 则必有 $(\exists e_i)(e_i \in Q(T_d, (E_1/E_2/E_3)) \rightarrow e_i \notin Q(T_d, (E_1//E_3)))$, 即存在另一条从 E_1 开始访问 E_2 元素实例的路径, 这与 $E_1 a E_3$ 相互矛盾, 假设不成立。 $(E_1/E_2/E_3) \subseteq (E_1//E_3)$ 得证。同理可以证明 $(E_1//E_3) \subseteq (E_1/E_2/E_3)$, 因此 $(E_1/E_2/E_3) = (E_1//E_3)$ 成立。

[证毕]

推论 如果 $E_1 a E_n$, 那么 $(C_1 E_1 C_2 E_2 C_3 \dots C_n E_n) = (C_1 E_1 // E_n)$ (C_i 是路径操作符 '/' 或 '//')。

证明方法同定理2(略)。

4.3 路径表达式最小化算法

在进行路径缩短时要首先对路径表达式进行预处理。预处理的工作有两个: ①保证路径的合法性, 即路径上相邻元素之间满足包含关系。对不合法的路径表达式不予处理。②将以 "/" 开头的路径表达式转换为以 "//" 开头的路径表达式。如将路径表达式 `/pub/papers/paper/sections/section/title` 转换为 `//papers/paper/sections/section/title`, 由于在任意一个模式图中根元素都排他包含其直接子元素, 因此这样的转换是等价的。

算法1 利用排他性包含约束缩短路径表达式

输入: 路径表达式 $(C_1 E_1 C_2 E_2 C_3 \dots C_n E_n)$ ($C_1 = "/"$)

输出: 被缩短的路径表达式

BEGIN

{ $i=1; j=3;$

WHILE ($j \leq n$)

{ 在包含关系表中查找 $E_i a E_j$;
IF (已找到) THEN { 从 E_i 到 E_j 的路径改写为 $E_i // E_j$;
 $i=j; j=j+3;$
ELSE IF ($C_i = "/"$ 且 $j > 3$) THEN
{ $E' = E_{j-1}$ 的孩子节点;

在包含关系表中找 $E_i a E'$;
IF (已找到) THEN
{ 将从 E_i 到 E_{j-1} 的路径改写为 $E_i // E_{j-1}$;
 $i=j; j=j+2;$
ELSE $j=j+1$;
}

总结 本文的工作可以看成是对路径缩短策略的改进和补充。对于路径表达式 `/pub/papers/paper/sections/section/title`, 使用文献[8]中的路径缩短策略可以缩短为 `//paper/sections/section/title`, 而使用本文的方法却可以缩短为 `//papers/section/title`, 计算它只需要两次结构连接运算, 并且在文档中元素类型 `papers` 的实例要比 `paper` 的元素实例少得多, 从而有效地优化了路径表达式查询。使用文[8]中的方法可以将 `/pub/books/book/author/name` 缩短为 `//book/author/name`, 而使用本文的方法可以缩短为 `//books/author/name`, 虽然缩短后的路径长度相同, 但 `books` 元素实例比 `book` 的元素实例少得多。通过以上分析可见本文提出的路径表达式的优化策略是有效的和现实可行的。今后我们将进一步探讨 XML 的完整性约束在 XML 查询优化中的作用。

参考文献

- Goldman R, Widom J. DataGuides: Enabling query formulation and optimization in semistructured databases[C]. In: Proc. of the 20th Int. Conf. on VLDB, 1997, 8: 436~445
- Fernandez M, Suciu D. Optimizing regular path expressions using graph schemas[C]. In: Proc. of the 14th Int. Conf. on Data Engineering, 1998, 2: 14~23
- Li Q Z, Moon B. Indexing and querying XML data for Regular Path Expressions[C]. In: Proc. of the 27th Int. Conf. on VLDB, Roma, Italy, 2001
- Wood P T. Minimizing simple xpath expressions[C]. In: Proc. of the 4th Int. Workshop on the Web and Database (WebDB), Santa Barbara, California, USA, 2001, 5: 21~24
- Amer-Yahia S, Cho S, Lakshmanan L K S, Srivastava D. Minimization of tree pattern queries[C]. In: Proc. of the 2001 ACM SIGMOD Conf. on Management of Data, Santa Barbara, California, USA, 2001, 5: 21~24
- Flesca S, Furfaro F, Masciari E. On the minimization of Xpath queries[C]. In: Proc. of the 29th Int. Conf. on VLDB, Berlin, Germany, 2003
- Chan C Y, Fan W F, Zeng Y M. Taming XPath Queries by Minimizing Wildcard Steps[C]. In: Proc. of the 30th Int. Conf. on VLDB, Toronto, Canada, 2004
- Wang G R, Sun B, Lv J H, Yu G. Query Processing and Optimization for Regular Path Expressions[J]. Journal of Computer Science & Technology, 2004, 19(2): 224~238
- Al-Khalifa S, Jagadish H V, Koudas N, Patel J M, Srivastava D, Wu Y Q. Structural Joins: A Primitive for Efficient XML Query Pattern Matching[C]. In: Proc. of the 18th Int. Conf. on Data Engineering, Los Alamitos; IEEE Press, 2002, 141~152
- 京: 中国电力出版社, 2001
- Gamma E, Helm R, Johnson R, et al. Design patterns: Elements of reusable object-oriented software, Boston: Addison Wesley, 1998
- DAML Services Coalition. DAML-S: Semantic markup for Web services. <http://www.daml.org/services/daml-s/0.9/daml-s.html>, 2003
- Liu S, Khalaf R, Curbera F. From DAML-S processes to BPEL4WS. In: Proceedings of the 14th International Workshop on Research Issues on Data Engineering, Web Services for E-Commerce and E-Government Applications. Boston, MA: IEEE, 2004, 77~84
- Rao J H, Kungas P, Matskin M. Logic-based Web services composition: from service description to process model. In: Proceedings of the IEEE International Conference on Web Services. San Diego, California: IEEE, 2004, 446~453
- UDDI Org. Universal description, discovery, and integration (UDDI). <http://www.uddi.org>
- The globus alliance. Globus toolkits 3.0 documentation. <http://www-unix.globus.org/toolkit/docs/3.0/index.html>, 2004
- Milanovic N, Malek M. Current solutions for Web service composition. IEEE Internet Computing, 2004, 8(6): 51~59
- Foster I, Kesselman C, Nick J M, et al. Grid Services for Distributed System Integration. IEEE Computer, 2002, 35(6): 37~46
- Tuecke S, Czajkowski K, Foster I, et al. Open grid services infrastructure (OGSI) version 1.0. <http://www.ggf.org/ogsi-wg>, 2002
- Andrews T, Curbera F, Dholakia H, et al. Business process execution language for Web services version 1.1. [ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf](http://www6.software.ibm.com/software/developer/library/ws-bpel.pdf), 2003
- 国家电力公司. 电力网电能损耗计算导则 DL/T 686-1999[M]. 北

(上接第120页)