DTD 约束下的 XML 树模式查询最小化

王梅娟[1],庞引明[2],谈子敬[2]
WANG Mei-juan[1],PANG Yin-ming[2],TAN Zi-jing[2]

1.中国人民解放军理工大学 理学院 基础电子学系,南京 211101
2.复旦大学 计算机与信息技术系,上海 200433
1.Department of Fundamental Electronics,PLA University of Science and Technology,Nanjing 211101,China
2.Department of Computing and Information Technology,Fudan University,Shanghai 200433,China
E-mail:wangmeijuan1984@hotmail.com

WANG Mei-juan,PANG Yin-ming,TAN Zi-jing.Minimization of XML tree pattern queries under DTD constraints. Computer Engineering and Applications,2009,45(23):144-148.

Abstract: Many XML query languages use tree patterns to navigate an XML document and select a set of element nodes.Since the efficiency of tree pattern matching against an XML tree-structured database depends on the size of the pattern,it is essential to identify and eliminate redundant nodes in the pattern and do so as quickly as possible.This paper studies tree pattern minimization in the presence of DTD.The SC to ESC that can express descendant relationships under DTD constraints is extend. This paper shows that minimization of tree pattern queries contain {ESC,/,//,[],*} is EXPTIME,and,in particular,when tree pattern branches are limited,queries are PTIME.At last provide an algorithm for minimization of limited branched tree patterns and analyze its complexity.

Key words: Extensible Markup Language(XML);tree pattern query;Document Type Difintnion(DTD) constraint

摘 要: 目前大部分 XML 查询语言都使用树模式来匹配待查询的 XML 文档树以得到所需要的、与模式树相吻合的查询结果,此效率在很大程度上取决于 XML 模式树的大小,那么尽可能快速地查找并删除查询模式树中的冗余节点就变得十分重要。重点讨论 DTD 约束下树模式的最小化问题,将 DTD 兄弟约束 SC 拓展成扩展兄弟约束 ESC,使其能够表达 DTD 约束中的祖先-后代关系;并指出只包含{ESC,/,//,[],*}的查询树模式的最小化问题的复杂度是指数级的,且当模式树是分支受限的时候,其最小化问题的复杂度是多项式时间的;最后给出了一个多项式时间的受限分支的模式树最小化算法。

关键词: 可扩展标记语言;树模式查询;文档类型定义(DTD)约束
DOI: 10.3778/j.issn.1002-8331.2009.23.040 **文章编号:** 1002-8331(2009)23-0144-05 **文献标识码:** A **中图分类号:** TP311

1 引言

现有的许多 XML[1]查询语言如 XPath[2]、XQuery[3]、Quilt[4]和 XML-QL[5]等都使用模式树来表达查询需求,通过把模式树与 XML 文档树数据库相匹配来得到想要的 XML 文档树的片段。一般而言,在一个确定的 XML 文档中匹配一个查询模式树的效率很大程度上依赖于模式树的规模,查询模式树的最小化运算是在一个相对较小的查询模式集上进行操作,可以加快在被查询文档中筛选和匹配对应文档节点的速度。

研究的问题可以描述为:给定一个树模式查询 TPQ(Tree Pattern Query),计算出它的最小树模式 TPQ′,使得 TPQ′=TPQ,且 TPQ′最小(节点数最少)。

DTD 为 XML 文档树提供了一种类型定义和结构上的约束,人们所感兴趣的是研究 DTD 约束下的查询模式树最小化问题。主要工作有:扩展了文献[6-7]中的 DTD 约束 SC,将其扩展成 ESC,ESC 的最大好处是它能够表达 DTD 约束中的祖先-后代关系;分析指出在 ESC 约束下,包含{ESC,/,//,[],*}的查询表达式对应的模式树的最小化问题的时间复杂度是指数级的;特别的指出,当模式树是分支受限的模式树时,其最小化问题的复杂度是多项式时间的;最后给出了一个受限分支的模式树最小化算法,并证明它是多项式时间的。

2 准备工作

2.1 已有研究成果

现有成果中与之相关最早的研究是文献[8]提出的基于不包含符号"//"的简单 XPath 表达式的最小化问题,而 XPath 表达式在文献[9-10]中被证明为与查询模式树是等价的;文献[9]研究了另一种被称为 $XP^{\{/,//,[]\}}$ 的 XPath 表达式片段的最小化问题,成果表明,包含操作符"/","//","[]",但不包含通配符 '*' 的

基金项目: 国家自然科学基金(the National Natural Science Foundation of China under Grant No.60603043)。
作者简介: 王梅娟(1984-),女,助教,研究方向为 XML 数据库规范化存储;庞引明,男,博士,研究方面为数据库与知识库;谈子敬,男,博士,研究方向为数据库与知识库。
收稿日期: 2008-05-05 **修回日期:** 2008-12-29

XPath 查询表达式可以在多项式时间内完成最小化操作,而在同样的片段 $XP^{\{/,//,[]\}}$ 上,文献[11]提出了一种更加有效的查询模式树的最小化算法;文献[10]综合分析了文献[8-9]中的 $XP^{\{/,//,[]\}}$,并指出当两个查询模式树之间存在一个同态映射时,确定这两个模式树之间是否存在包含关系可以在多项式时间内解决,而对于更一般性的 XPath 查询表达式片段,查询模式树之间的相互包含问题就可能是 NP 完全问题;其他文献分别研究了 XPath 查询表达式片段、查询模式树的种种不同特性,其中文献[12-13]着重研究了 DTD 约束下的 XPath 查询表达式片段的相互包含;文献[13]还研究了一种重要的 DTD 约束——兄弟约束 SCs,在此基础上将其扩展为扩展兄弟约束 ESCs(Extended Sibling Constraints),使人们能够表达 DTD 约束中的祖先-后代关系。

由已有研究可知,寻找一棵查询模式树的最小化模式树,实质上就是判断该模式树与其最小化模式树之间的相互包含关系,那么,只要能够判断该模式树与其最小化模式树之间存在一个同态映射,问题就可以在多项式时间内解决。遵循同样的思路,查询模式树的最小化问题其实可以等价转化为不同模式树之间的相互包含问题,从而可以有效的加以解决。

2.2 形式化定义

引用前人工作中的相关形式化描述,给出该文的相关知识的定义。

首先,定义的 XML 文档树模型是一棵定义在无穷字符集 Σ 上的树模型,一棵 XML 文档树是一个无序、无秩的有穷结构,其节点标记均来 Σ,Σ 上所有树的集合记为 T_Σ。

对于查询而言,XPath 表达式的片段可以被递归地定义成下面的语法规则:

exp→exp/exp|exp//exp|exp[exp] |δ|*| •

其中,δ 为标记集 Σ 中的标记符,而符号"•"则代表着"当前节点"。

一棵树模式 p 就是一棵定义在符号集 $\Sigma\cup\{*\}$ 上的无序树。它含有一个奇异的边的子集,叫做后代边集;还含有一个 k-元组($k\geqslant 0$)的节点集,叫做结果元组集,k 被称为结果元组集的势,也被称做模式树 p 的势。已有研究表明,一个 XPath 查询表达式与一棵势为 1 的查询模式树在查询效果上是等价的,所以 XPath 查询表达式之间的相互包含问题与查询树模式之间的相互包含问题是等价的。该文将不再区分它们之间的区别。

假设给定一棵查询树模式 p 和一棵文档树 $t\in T_\Sigma$,p 在 t 中的一个嵌入 e 定义为:

(1)$e(\mathrm{ROOT}(p))=\mathrm{ROOT}(t)$;

(2)$\forall x\in \mathrm{NODES}(p)$,$\mathrm{LABEL}(x)=*$ 或 $\mathrm{LABEL}(x)=\mathrm{LABEL}(e(x))$;

(3)$\forall x,y\in \mathrm{NODES}(p)$,如果满足$(x,y)$为 p 中的一条父子边或前后代边,则$(e(x),e(y))$也为 t 中的一条父子边或前后代边[8]。

$p(t)$表示查询树模式 p 在文档树 t 中的一个解答,即 $p(t)=\{B|B$ is "true" if e exists,and "false" if e not exists$\}$。由于 p 为布尔型的,故只可能为"查询成功"或"查询不成功",两者只居其一,且必居其一。

设给定的两个树模式为 p_1、p_2,说 p_1 包含于 p_2(记为 $p_1\subseteq p_2$),当且仅当 $\forall t$ $p_1(t)\subseteq p_2(t)$成立;说 p_1 与 p_2 是相互等价的(记为 $p_1\equiv p_2$)当且仅当 $p_1\subseteq p_2$和 $p_2\subseteq p_1$同时成立。

上文已经提出,查询模式树的最小化问题可以被归约为两个树模式之间的相互包含问题,重点要研究的是 DTD 约束下的查询模式树的最小化问题,下面分别给出其形式化表述:

给定一个查询树模式 p,构造一个新的查询树模式 $p_{\min}$,使得 $p\equiv p_{\min}$,并且有 $\forall p'\equiv p\&p'\neq p_{\min}(|p'|>|p_{\min}|)$成立。其中的$|p|$表示树模式 p 的大小,即树中节点数量的多少。

设 D 表示任一 DTD,用 $\mathrm{SAT}(D)$来表示那些合乎 D 的、T_Σ 上的文档树的集合。假设有两个查询树模式 p_1、p_2,若 $\forall t\in \mathrm{SAT}(D)$,有 $p_2(t)\subseteq p_1(t)$成立,则 p_1 D-contains p_2,记为 $p_2\subseteq_{\mathrm{SAT}(D)}p_1$;若 $p_1\subseteq_{\mathrm{SAT}(D)}p_2$且 $p_2\subseteq_{\mathrm{SAT}(D)}p_1$,则 p_1 D-equivalent p_2,记为 $p_1\equiv_{\mathrm{SAT}(D)}p_2$;如果不存在更小的查询树模式 q D-equivalent p,则称 p 是 D-minimal[6,14-15]的。

由上述定义可以看出,所谓 DTD 约束下的模式树最小化其实就是在一般模式树最小化的基础上加入 DTD 对文档树可能的约束条件。该文同文献[9-10]一样,也只考虑无复制节点的 DTD,即任一节点的若干子节点的标号均不相同。

3 DTD 约束下的 XML 查询树模式的最小化

3.1 扩展兄弟约束 ESC

DTD D 包含若干个完整性约束,这些约束可以被用来判定查询树模式之间的 D-equivalent 性质。P.T.Wood 首先提出两种 DTD 约束,称为儿子约束和父亲约束[8,13]。文献[8]中运用它们可以证明某些查询模式树之间的 D-equivalent 性质,随后文献[13,17]将其进一步发展成为兄弟约束 SC[8,13],具体含义如下:

设 $t\in T_\Sigma$为一棵合式的文档树,$\forall a,c\in\Sigma$是 t 中元素节点的名称,$B\subseteq\Sigma$为一个元素名称的集合,说文档树 t 满足兄弟约束 SC $a:B\Downarrow c$,当且仅当,无论何时只要文档树 t 中标号为 a 的元素节点拥有以任一 $b\in B$ 为标号的元素节点为孩子节点,则节点 a 必然同时拥有一个以 c 为标号的节点作为子节点。当 $B=\phi$ 时,兄弟约束 SC 就退化成儿子约束了[18]。

如果 SC 能够被任意一棵文档树 $t\in T_\Sigma$所满足,则称定义于集合 Σ 之上的兄弟约束 SC 是平凡的,如果 $c\in B$,则 SC $a:B\Downarrow c$ 必然是平凡的。文献[8,13]指出,设 S 为一个定义于 Σ 上的兄弟约束 SCs 的集合,$SAT(S)$表示 T_Σ中的某些文档树的集合,$\forall t\in SAT(S)$,t 均满足 S 中的每一条兄弟约束 SC。

不过,认为兄弟约束 SC 在表达由 DTD 所蕴涵的完整性约束时并不完备,因为它只表达了父子两代关系,对于实际存在的超过两代的子孙约束却不能表达。因此,将兄弟约束 SC 拓展成扩展兄弟约束 ESC(Extended Sibling Constraints),定义如下:

定义 1 设 $t\in T_\Sigma$,$a,c\in\Sigma$,B_1,$B_2\subseteq\Sigma$为与 SC 中 B 一样的元素标号的集合,称文档树 t 满足扩展的兄弟约束 ESC $a:B_1\Downarrow C$(其中 $C=B_2\downarrow C|c$),当且仅当,无论何时,t 中的以 a 为标号的元素节点拥有以任意一个以 $b\in B_1$为标号的元素节点为孩子节点时,节点 a 必然同时拥有一个以 c 为标号的子节点(或者拥有一个标号为 c 的子孙节点,通过一串父子边的路径,经过每一个以 $b'\in B_2$为标号的若干节点直到节点 c)。当 $B_1=\phi$ 时,扩展兄弟约束 ESC 就退化成子孙约束,它是儿子约束的一种拓展形式,或称为一般形式。

同样的,如果 $\forall t\in T_\Sigma$,t 均满足 ESC,定义于 Σ 上的扩展兄

弟约束 ESC 是平凡的。所以,如果满足 $c \in B_1 \& B_1 = B_2$,则 ESC $a:B_1 \Downarrow C$(其中 $C=B_2 \downarrow C|c$)是平凡的。再者,设 S 为一个定义于 $\sum$ 上的 ESCs 的集合,那么 $SAT(S)$ 表达了 $T_\sum$ 中的若干文档树的集合,$\forall t \in SAT(S)$,t 满足 S 中的每一个 ESC 约束。

扩展兄弟约束 ESC 递归地定义了一个由 DTD 所蕴涵的后代关系约束。设给定 $e,f,g \in B_2$,若 $B_1=\phi$,则表达式 $a:B_2 \downarrow B_2 \downarrow B_2 \downarrow C=a:e \downarrow f \downarrow g \downarrow c$ 并非儿子约束,而是一条约束路径:$e \rightarrow f \rightarrow g \rightarrow c$,表示后代关系的约束;若 $B_2=\phi$,则 ESC=SC;若 $B_1 \neq \phi, B_2 \neq \phi$,当 C 递归地终止于节点 c 时,ESC 显示了一个比 SC 更加一般化的兄弟约束形式。

例 1 通过一个 DTD 片段 cXML 的实例来分析 SC 和 ESC 的具体含义:

<! ELEMENT cXML ((Hearder,(Message|Request))|Response)>

<! ELEMENT Hearder(OrderReq|SupplierListReq)>

<! ELEMENT OrderReq(OrderHead,ItemOut+)>

<!ELEMENT OrderHead(Total,ShipTo?,BillTo,Payment?)>

<! ELEMENT ItemOut(ItemID,ItemDetail?,SupplierID?,ShipTo?)>

从片段 cXML 中可以看出,一个 cXML 元素可以没有 Hearder 元素作为子元素,因为它可以用 Response 取代 Hearder 作为子元素节点;然而,一旦它具有 Hearder 作为子元素,则它必然同时拥有 Message(或 Request)元素作为子节点。换言之,DTD cXML 中蕴涵兄弟约束 SCs cXML:{Message,Request}$\Downarrow$Hearder。

但再仔细的看可以发现,只使用上面的兄弟约束不足以表达一些路径约束,如约束路径 Hearder/OrderHead/Total 就显然存在。因此,需要为其定义一个扩展兄弟约束 ESCs cXML:{Message,Request}$\Downarrow$Hearder$\downarrow$OrderHead$\downarrow$Total。

3.2 树模式查询最小化的时间复杂度

定理 1 设查询模式树 $p_1,p_2 \in p^{\{ESC,/,//,[],*\}}$,确定 p_1、p_2 之间的相互包含关系的时间复杂度是指数级的($p^{\{ESC,/,//,[],*\}}$ 表示 ESC 约束下的,仅由符号"/","//","[]","*"组成的查询模式树的集合)。

证明 设 S 为 DTD 中所蕴涵的 ESC 的集合。根据文献[9-10,18-19]中所列出的相关定理,要证明上述定理,只需检查是否有 $p_1 \not\subset_s p_2$ 成立,并且 $p_1 \not\subset_s p_2 \Leftrightarrow \forall t \in SAT(S)$,如果 $p_1(t)$ 存在,则 $p_2(t)$ 必不存在。

根据树自动机理论[19-22],构造三个非确定的自顶向下的树自动机:A_s 来检测文档树 t 是否满足 S;A_{p1} 用于检查文档树 t 是否满足查询模式树 p_1;A_{p2} 则用于验证不存在一个 $t \in SAT(S)$ 能够匹配 p_2。

设 A_{p1} 的状态为形如(S_1,S_2)的节点对,其中的 S_1、S_2 为 p_1 中的节点组成的集合。$\forall v \in NODES(t)$,如果 $v \in S_1$,则表明文档树中以当前节点为根的子树与 p_1 中以 v 为根的子模式相匹配;如果 $v \in S_2$,则表明 $\exists u \in NODES(t)$,u 为 v 的子孙节点,匹配 p_1 中以 v 为根的子模式,并且,v、u 满足 ESCs 中的每一条约束。由此很明显可以看出,A_{p1} 中的状态数为指数级的。加之,p_1 的每一个节点均有 k 个孩子节点,这使得相对应的正则表达式的大小与 k 成比例。

对 A_{p2} 而言,情况与 A_{p1} 相类似。

最后考虑 A_s,由于操作符"[]"的存在,如果满足扩展兄弟约束 ESC 的查询模式 p 中的一个节点有 m 个孩子节点,则相对应的正则表达式的大小是与 m 成比例的。

综合起来看,A_s、A_{p1} 和 A_{p2} 的复杂度都为指数级的。所以,式子 $A=A_s \times A_{p1} \times A_{p2}$ 的复杂度也是指数级的,A 就是最终所求的非确定的自顶向下的树自动机,它不接受任何一棵能够导致 $p_1 \subseteq_s p_2$ 成立的模式树。

证毕。

研究的最小化问题所感兴趣的是上面定理中所提到的查询模式树集合 $p^{\{ESC,/,//,[],*\}}$ 中的一个子集合,叫做受限分支树模式[23]。

定义 2 设 p 为一棵受限分支树模式,则 p 为树模式集合 $P^{\{/,//,[],*\}}$ 中的一棵查询模式树,使得:

(1)p 中的每一个非叶子节点可以拥有任意数量的子节点;

(2)如果节点 $n \in p$ 拥有 k 个子节点,记为 $n_1,\cdots,n_k$,则至少其中的 $k-1$ 个子模式 $sp_{ni}(i=1,\cdots,k)$ 是线性的(即 $sp_{ni} \in P^{\{/,//,*\}}$)。其中的 sp_{ni} 是 p 中以 $n_i(i=1,\cdots,k)$ 为根的子模式。

根据定义 2,将遵循受限分支规则的扩展兄弟约束 ESC 称为限制 ESC(记为 RESC),并且有下面的引理:

引理 1 给定树模式 $p_1,p_2 \in p^{\{RESC,/,//,*\}}$,判断 $p_1 \subseteq p_2$ 是否成立的问题是多项式时间的。

证明 设 S 为 DTD 中所蕴涵的限制扩展兄弟约束 RESCs 的集合。根据文献[19-22]中的相关定理,只需验证是否有 $p_1 \not\subset_s p_2$ 成立即可,并且 $p_1 \not\subset_s p_2 \Leftrightarrow \forall t \in SAT(S)$,如果 $p_1(t)$ 为真,则 $p_2(t)$ 必为假。

根据树自动机[19-20]的理论,来构造三个非确定的、无秩的、自顶向下的树自动机 A_s、A_{p1} 和 A_{p2},分别用于检测文档树 t 是否满足约束集、模式树 p_1 和验证不存在文档 $t \in SAT(S)$ 能够匹配树模式 p_2。

设 $A=A_s \times A_{p1} \times A_{p2}$,则 A 为一个非确定有限自动机,用于验证 A 不接受任何将导致 $p_1 \subseteq_s p_2$ 的文档树。

A_s 检查的是约束集 S 中的儿子和后代子孙关系。其最坏的时间耗费分别为 $O(n^2)$ 和 $O(n^3)$。A_{p1} 和 A_{p2} 在文献[22-23]中都有已证明的多项式时间的算法。再有,A_s、A_{p1} 和 A_{p2} 的构造时间和规模,也是 S,p_1 和 p_2 的规模的多项式。

证毕。

根据上面的引理,有下面的定理:

定理 2 设 p 为一个受限分支树模式,在 RESC 约束下,计算 p 的一个最小树模式 p_{min} 可以在多项式时间内完成。

证明 定理 2 是引理 1 的必然结论。

4 树模式最小化算法

4.1 chase 技术

几乎所有的查询模式树最小化算法都采用 chase [6-7,9,23]技术将一个约束集中的一条条约束逐一加入到待运算的模式树中去。

设 p 为一个查询模式树,S 为一个约束 RESCs 的集合,该约束集同样是由一个 DTD D 所蕴涵的。设 $s\in S$ 为一个非平凡的的约束 ESC(RESC),形如 $a:B_1\Downarrow C$(其中 $C=B_2\downarrow C|c$),其中 $B_1=\{b_1,\cdots,b_n\}$,$B_2=\{b_1',\cdots,b_n'\}$。又设 $u\in$ NODES(p),u 拥有孩子节点 $v_1,\cdots,v_n$ 使得 LABEL(u)=a,LABEL(v_i)=b_i,$1\leqslant i\leqslant n$。而 u 并不拥有一个标号为 c 的儿子(或后代)节点。将约束 ESCs(RESCs)运用到模式树 p 中后会得到一棵新模式树 p',同时有 NODES(p')=NODES(p)$\cup\{\omega\}$、$\omega\notin$ NODES(p),EDGES(p')=EDGES(p)$\cup\{(u,\omega)\}$,和 LABEL(ω)=c 成立。

一棵模式树 p 的基于约束 S 的一个 chasing 序列是形如 $P=P_0,\cdots,P_k$ 的一个序列,使得对每一个 $0\leqslant i\leqslant k-1$ 来说,P_{i+1} 都是在 P_i 的基础上运用 S 中的某些约束 ESC(RESC)所得到的直接结果,而且,没有约束 ESC(RESC)被应用到 P_k 上。值得注意的是,一个 chasing 序列是有限的,chasing 过程必然会在有限步内终止。在约束 S 基础上对模式树 p 进行 chase 过程的结果,表示为 $chase_s(p)$,即 P_k。

给出与文献[8]相类似的定理:

定理 3 假定查询模式树 $p_1,p_2\in p^{\{RESC,/,//,*\}}$,设 S 为一个 DTD D 中所蕴涵的限制扩展兄弟约束 RESCs 的集合,则有 $p_2\subseteq_{SAT(S)}p_1\Leftrightarrow chase_s(p_2)p_1$ 成立。

证明 定理 3 的证明过程与文献[8]中的相关定理证明过程类似,此处省略。

4.2 图模拟技术

文献[7]提出了一个比文献[9]中相关算法更为优化的模式树最小化算法,该算法采用了图模拟技术[24-25]。在设计算法时也采用图模拟技术,用于在 RESC 约束下进行受限分支树模式的最小化工作。

设一个有向图 $G=(V,E)$,$\forall u\in V$ 都有一个唯一的标号 LABEL(u)与之对应。post(u)表示一个节点集合,该集合中的任一节点均可从 u 通过一条边到达模拟是指 V 上的一个二元关系,记为"$<$",它使下面的规则成立:若 $u<v$,则 LABEL(u)=LABEL(v),并且对于每一个 $u'\in$ post(u),均存在 $v'\in$ post(v)使得 $u'<v'$成立。如果 $u<v$,说 u 被 v 所模拟、v 模拟 u、或 v 是 u 的一个模拟。设 sim(u)表示 u 的所有模拟节点的集合。

对受限分支树模式定义的模拟关系[9,21]如下:

如果 $u<v$,则(1)若 $u\rightarrow u'$,则 v 有一个子节点 v'使得 $u'<v'$;(2)若 $u\Rightarrow u''$,则 v 有一个后代节点 v''使得 $u''<v''$。

接下来看一个引理:

引理 2 [23] 设 u 为一个受限分支模式树 p 中的一个非冗余节点,则:

(1)p 中的节点 u 的一个子节点 v 是冗余的,当且仅当 u 还有另一个子节点 $\omega\in$ sim(v);

(2)p 中的节点 u 的一个后代节点 v 是冗余的,当且仅当 u 还有另一个后代节点 $\omega\in$ sim(v)。

设 N 为模式树 p 中的一些点的集合,cpar(N)表示的是 p 中那些具有子节点在 N 中的节点的集合;anc(N) 表示的是 p 中那些具有后代节点在 N 中的节点的集合。于是,如果 $u<v$:

(1)如果 $u\rightarrow u'$,则 $v\in$ cpar(sim(u'));

(2)如果 $u\Rightarrow u''$,则 $v\in$ anc(sim(u''))。

chase(p)不是一棵树,而是一个有向非循环图 DAG。通过如下方式得到 chase(p):设 $\sum'$包含 RESC 和 p 中所有的元素标签,根据儿子和后代约束,在 $\sum'$上来构造一个约束图 $G_R=(V_R,E_R)$。在 p 中任一个标记为 l 的节点,将 $G_R(l)$中以 l 为根的子树(图)拷贝到 p 中来。在上述过程中新加入到 p 中的节点称为 chase 节点,p 中的其他节点被称为原始节点。

4.3 模式树的最小化算法

根据以上相关定义定理,给出如下的模式树最小化算法:

Algorithm MinimizeDPQ

1. compute chase(p)
2. call DPQSimulation /* compute the simulation relations on chase(p)*/
3. DPQMinimization(root(chase(p)))

Algorithm DPQSimulation

1. $V\leftarrow$NODES(chase(p))in bottom-up order
2. for each $u\in V$ in order do
3. if u is a leaf then
4. sim(u)=$\{v\in V|$LABEL(v)=LABEL(u)$\}$
5. compute cpar(sim(u))
6. auganc(sim(u))=anc(sim(u))$\cup\{v\in V|$LABEL(v)$\Rightarrow$LABEL(u) is in RESC$\}\cup$\{anc($\{v\in V|$LABEL(v)$\Rightarrow$LABEL(u) is in RESC$\}$)$\}$
7. else
8. sim(u)= $\{v\in V|$LABEL(v)$\Rightarrow$LABEL(u),$v\in$cpar(sim(u'))
for each child u' of u,and $v\in$auganc(sim(u'')) for each descendant u'' of u $\}$
9. compute cpar(sim(u))
10. auganc(sim(u))$\leftarrow$anc(sim(u))

Function DPQMinimization(u)/* u is a nonredundant node of p */

1. for each child v of u do
2. if v is a child then /* $C=c$ in the RESC */
3. if u has another child $\omega\in$sim(v) that has not been deleted
4. then delete v /* delete the subtree rooted at v */
5. else DPQMinimization(v)/* v is nonredundant */
6. if v is a descendant then /* $C=B_2\downarrow C$ in the RESC */
7. if u has another descendant $\omega\in$sim (v)$\cup$auganc (sim(v)) that has not been deleted
8. then delete v
9. else DPQMinimization(v)

在上面的算法中,auganc 的前缀 aug 代表的是 augmented,几点说明如下:

(1)对于 chase(p)中的一个内部节点 u 来说,auganc(sim(u))与 anc(sim(u))是一样的;

(2)对于 chase(p)中的一个叶子节点 u 来说,auganc(sim(u))包含的不仅仅是 anc(sim(u)),还包括 chase(p)中的节点 v 及其祖先节点,v 满足 LABEL(v)$\Rightarrow$LABEL(u)在 RESC 中。

定理 4 算法 MinimizeDPQ 能够在多项式时间内正确地计算出一个给定查询模式树 p 的最小化的等价模式树 p'。

证明 由定理 2 显见。

5 结论与展望

在总结前人工作的基础上,重点讨论了 DTD 约束下查询最小化问题,为了在待查询的文档树中以相对较小的模式树,更快地实现与模式树匹配的相关元素节点查找,给出一个算法,针对查询模式树 TPQ,计算与之等价的最小查询树模式 TPQ'。

该文所感兴趣的是研究在 DTD 约束下的模式树的包含判

定问题,首先拓展了 DTD 的兄弟约束 SC,提出了扩展兄弟约束 ESC 约束作为 SC 的更一般形式;并且证明了包含{ESC,/,//,[],*}的模式树的最小化问题的复杂的是指数级的;最后给出了一个当模式树是分支受限时的 XML 树模式查询最小化算法 DPO,该算法是多项式时间的的。

在该文基础上,接下来的工作可以考虑以下一些放慢:进一步研究规范模型下的树模式包含问题;将现阶段对于 DTD 约束的讨论扩展到 XML schema 约束中进行查询最小化问题的研究等等。

参考文献:

[1] Bray T,Paoli J,Sperberg-McQueen C M.Extensible markup language (XML)1.0 X3C recommendation[EB/OL].(1998-02).http://www.w3.org/tr/1998/rec-xml-19980210.

[2] Clark J,DeRose S.XML path language(XPath)[EB/OL].(1999-11-16).http://www.w3.org/TR/xpath.

[3] Boag S,Chamberlin D,Fernandez M F,et al.XQuery 1.0:An XML query language[EB/OL].(2002-08-16).http://www.w3.org/TR/xquery/.

[4] Chamberlin D,Robie J,Florescu D.Quilt:An XML query language for heterogeneous data sources[C]//WebDB,2000.

[5] Deutsch A,Fernandez M,Florescu D,et al.XML-QL:A query language for XML[C]//Proceeding of the 8th International World Wide Web Conference,Toronto,1999.

[6] Amer-Yahia S,Cho S,Lakshmanan L V S,et al.Tree pattern query minimization[J].VLDB Journal,2002,11(4):315-331.

[7] Ramanan P.Efficient algorithms for minimizing tree pattern queries[C]//ACM SIGMOD,2002.

[8] Wood P T.Minimizing simple xpath expressions[C]//WebDB,2001.

[9] Amer-Yahia S,Cho S,Lakshmanan L K S,et al.Minimization of tree pattern queries[C]//SIGMOD,2001.

[10] Miklau G,Suciu D.Containment and equivalence for an XPath fragment[C]//PODS,2002.

[11] Yang L H,Lee M L,Hsu W.Efficient mining of XML query patterns for caching[C]//VLDB,2003.

[12] Schwentick N T.XPath containment in the presence of disjunction,DTDs,and variables[C]//ICDT,2003.

[13] Wood P T.Containment for XPath fragments under DTD constraints[C]//ICDT,2003.

[14] Benedikt M,Fan W,Kurper G M.Structural properties of XPath fragments[C]//ICDT,2003.

[15] Jason D S,Wang T L,Giugno R.Algorithmics and applications of tree and graph searching[C]//PODS,2002.

[16] Chen Z,Jagadish H V,Lakshmanan L V S,et al.From tree patterns to generalized tree patterns:On efficient evaluation of xquery[C]//VLDB,2003.

[17] Wood P T.Rewriting XQL queries on XML repositories[C]//17th BNCD,British National Conf on Database,2000.

[18] Neven F,Schwentick T.Query automata on finite trees[J].Theoretical Computer Science,2002,275:633-674.

[19] Slutzki G.Alternating tree automata[J].Theoretical Computer Science,1985,41(2/3):305-318.

[20] Neven F,Schwentick T.Automata-and logic-based pattern languages for tree-structured data[C]//PODS,2002.

[21] Milo T,Suciu D.Typechecking for XML transformers[C]//PODS,2000.

[22] Papakonstantinon Y,Vianu V.DTD inference for views of XML data[C]//PODS,2000.

[23] Flesca S,Furfaro E F.Masciari on the minimization of Xpath queries[C]//VLDB,2003.

[24] Abiteboul S,Buneman P,Suciu D.Data on the Web[M].San Francisco,CA:Morgan Kaufman,2000.

[25] Buneman P,Davidson S,Fernandez M,et al.Adding structure to unstructured data[C]//ICDT,1997.

(上接 139 页)

[16] 傅间莲,陈群秀.一种新的自动文摘系统评价方法[J].计算机工程与应用,2006,42(18):176-177.

[17] Minel J L,Nugier S,Piat G.How to appreciate the quality of automatic text summarization[C]//Proc of the ACL/EACL'97,1997:25-30.

[18] Saggion H,Lapalme G.Concept identification and presentation in the context of technical text summarization[C]//Proc of the Workshop on Automatic Summarization.New Brunswick,New Jersey:Association for Computing Linguistics,2000:1-10.

[19] Lin C C,Wang Y C,Yeh C H,et al.Learning weights for translation candidates in Japanese-Chinese information retrieval[J].Expert Systems with Applications,2008.

[20] Gervain J,Nespor M,Mazuka R,et al.Bootstrapping word order in prelexical infants:A Japanese-Italian cross-linguistic study[J].Cognitive Psychology,2008,57:56-74.

[21] Lee K S,Kageura K,Choi K S.Implicit ambiguity resolution using incremental clustering in cross-language information retrieval[J].Information Processing and Management,2004,40:145-159.

[22] Ma Q,Kanzaki K,Zhang Y,et al.Self-organizing semantic maps and its application to word alignment in Japanese-Chinese parallel corpora[J].Neural Networks,2004,17:1241-1253.

[23] Lin C C.Learning weights for translation candidates in Japanese-Chinese information retrieval[J].Expert Systems with Applications,2008.

[24] Ohtsuki K.Japanese large-vocabulary continuous-speech recognition using a newspaper corpus and broadcast news[J].Speech Communication,1999,28:155-166.

[25] Du J L,Yu P F.Xu J,et al.Towards the processing breakdown of syntactic garden path phenomenon:A semantic perspective of natural language expert system[J].Journal of Communication and Computer,2008,5(11):19-27.

[26] Shimizu T,Ashikari Y,Sumita E,et al.NICT/ATR Chinese-Japanese-English speech-to-speech translation system[J].Tsinghua Science and Technology,2008,13(4):540-544.

[27] Du J L,Yu P F.Zhao H Y,et al.Study on controllability of semantic accessibility scale from the internet-based system of automatic text summarization and evaluation[J].Journal of Communication and Computer,2008,5(9):54-60.