半结构化数据查询的处理和优化*

陈 滢 王能斌

(东南大学计算机科学与工程系 南京 210096)

摘要 半结构化数据的特点是数据的结构不规则或不完整,其模型都基于带根有向图,因此,查询处理过程本质上是对图的搜索过程.另外,通配路径使查询处理更加复杂化.文章详细介绍了异构数据源集成系统Versatile中采取的半结构数据OIM(model for object integration)对象的查询和优化策略,包括查询计划的生成、路径扩展和路径索引、层次索引和基于数据源知识这3种查询优化方法.文章介绍的方法同样适用于其他的半结构化数据模型.

关键词 半结构化数据,查询处理,优化.

中图法分类号 TP311

半结构化数据(semistructured data)的特点是数据的结构不规则(irregular)或不完整(incomplete)[1],表现为数据不遵循固定的模式、结构隐含、模式信息量大、模式变化快、模式和数据统一存储等特点.半结构化数据一般有两种来源.① 直接来自半结构化数据源,如Web数据、各种类型电子文档(电子表格,TEX,WORD)、电子邮件等,这些数据源数据是典型的半结构化数据;② 作为异构数据源集成系统的公共数据模型引入,如Versatile[2],TSIMMIS[3]和LORE[4],之所以采用半结构化数据作为公共数据模型是因为半结构化数据模型既能描述半结构化数据,同时也能描述结构化的数据.随着WWW(World Wide Web)的普及和对异构数据源系统进行集成的需要,半结构化数据的研究近年来逐渐受到重视.人们对半结构化数据的数据模型[2~5]、查询语言[2~5,7]、查询处理[1,4,6,7]以及数据存储和表示[8]都作了一定研究.半结构化数据由带有根节点的有向图表示,如文献[2]采用OIM(model for object integration),文献[3,4]采用OEM(object exchange model),文献[5]采用图数据库(graph database).其中节点表示对象或子对象,有向边表示对象的聚合关系,节点或边上的标记表示对象的属性.半结构化对象的查询语言采用类OQL(object query language)风格,如文献[2]采用OIQL(object integration query language),文献[4]采用LOREL,为适应半结构化数据模式庞大且用户不完全了解其模式的特点,查询语言支持通配路径的查询,文献[5]甚至支持正规路径表达式.因此,半结构化数据的查询本质上是对有向图的搜索,它具有以下特点:由于数据和模式信息统一存放(自描述性),因此,很难进行与关系数据库查询处理类似的处理和优化方法;由于数据结构不规则或不完整,需支持内容的通配和类型自动转换;有向图中可能存在圈,因此在处理通配路径时,应避免查询进入无限循环.一般地,针对半结构化数据的有向图查询只能基于穷尽搜索,在遍历整个图的过程中寻找符合条件的路径,显然,这种方法在图很大时效率很低.为了克服这个缺点,文献[4]采用数据导则(DataGuide),文献[9]采用图模式(graph schemas)的方法,它们都是从有向图中抽取出半结构化数据的模式信息以指导查询处理和优化.此外,文献[4]还提出了模式的一个递增维护算法,并且利用索引提高查询速度.但是,对于数据量大或模式变化的数据源,如某些Web数据,用模式信息效率较低(如文献[4]中阐明从OEM转化为DataGuide的算法相当于将非确定自动机NFA转化为等价的确定自动机DFA,因此,最坏情况是其复杂度可能是指数级).

本文详细介绍异构数据源集成系统Versatile中采取的半结构化数据的查询和优化策略.Versatile[6,7]是东

* 本文研究得到国家自然科学基金资助.作者陈滢,1973年生,博士,主要研究领域为数据库,计算机网络.王能斌,1929年生,教授,博士生导师,主要研究领域为数据库,信息系统.

本文通讯联系人:王能斌,南京 210096,东南大学计算机科学与工程系

本文1998-06-02收到原稿,1998-09-01收到修改稿

南大学研制的一个基于CORBA[8]的异构数据源集成系统原型，旨在以"即插即用"方式集成来自不同数据源的数据，该系统的研制受到国家自然科学基金资助. 在Versatile中采用对象集成模型OIM[10]作为各数据源的输出模式的数据模型，查询语言是OIQL. 查询集成器QI(query integrator)接收OIQL查询后分解至各个数据源. 在Versatile中，有些数据源，如RDBMS和OODBMS，由于本身具有数据管理功能，因此相应的包装器(wrapper)将OIQL翻译为本地查询语言(如SQL，OQL等)，再将结构转换为OIM对象；而某些数据源，如文件系统、Web数据，由于自身没有数据查询功能，因此在Versatile中包装器将数据包装为OIM，然后实施OIM的查询处理[11]. 本文介绍在各数据源中如何进行查询处理，因此不涉及多数据源问题*. 其中查询优化策略采用了路径索引、层次索引和数据源知识，利用这些优化措施可以在很大程度上减少有向图的搜索范围. 本文的讨论虽然是基于OIM，但同时也是用于其他的基于有向图的半结构化数据模型.

1 OIM对象模型

1.1 OIM对象

OIM对象模型是Versatile中的半结构化数据模型. 在OIM对象模型中，一个对象用四元组(OID,n,t,c)表示，其中OID是对象标识符，n表示对象名，t表示对象类型，c表示对象值，该四元组称为对象描述子. t除了可以表示基本数据类型(如integer，char，float，string等)外，还可表示集合数据类型(如set，list，bag等)、可变长数据类型(CLOB，BLOB)和引用类型(ref). 如果一个对象的类型是引用类型，表示该对象由其他对象聚集而成，称为该对象的亲子对象，它的值是亲子对象标识符的集合. 在OIM对象模型中，相关的一组对象聚集在一起，组成一个OIM对象. 例如，关系数据库的一张表、文件系统的一个文件，WWW上由某个结点出发通过锚连接的HTML文件的集合均可看做OIM对象. 一个OIM对象O可以用带根连通有向图表示成$O(r,V,E)$，其中节点集V表示对象，边集E表示对象之间的引用关系. 根节点r是一个聚集对象，它是引用类型的. 除了用OID外，还可以用OIM路径表示对象集合或用带谓词的路径表示对象，路径允许带通配符，下面将通过举例加以说明.

OIQL是OIM对象的查询语言，它包括DML(data manipulation language)和DDL(data definition language)，其中查询语句是select...from...where...形式(sfw).

1.2 OIM对象举例

例1：图1是从一个Web页面得到的OIM对象**，描述的是计算机系教师和学生的情况.

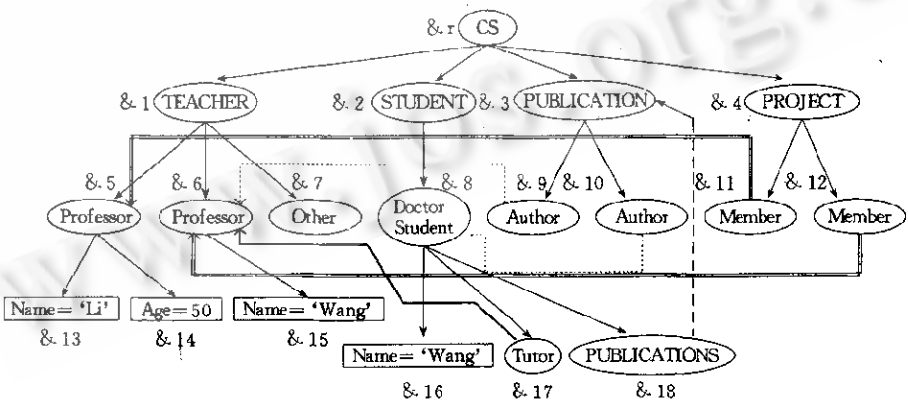


图1 OIM对象示例

图1中的OID，比如&r，&1，&2等由系统生成，保证唯一性且对用户透明. 根对象表示为(&r，CS，REF，

* 一种简易的解决查询来自多数据源OIM对象的方法是将所有OIM对象传输至同一地点，再实施查询.

** Web页面通过Web Wrapper抽取出来，并组装成OIM. 实际的Web页面较复杂，这里仅是简化的例子.

{&1,&2,&3,&4}),在图中的路径是CS,对象&1,&2,&3,&4是&r的亲子对象."王教授的姓名"可以表示为(&15,Name,STRING,Wang),它是原子对象,它所对应的一条路径是CS:TEACHER:Professor:Name.对象在OIM图内可能对应若干路径,同时,一个路径可能表示许多对象.例如,路径CS:TEACHER:Professor表示所有的"教授"对象集合.路径CS:*:Name表示所有人的姓名,其中符号*可以匹配零个或多个对象名序列.

OIM图内可能含有圈,如路径*:Publication展开后在OIM内可以对应无数条路径,例如,CS:Publication,CS:STUDENT:Doctor Student:Publications:Publication,CS:Publication:Author:Doctor Student:Publication等,其原因就是存在有向边构成的环&8→&18→&4→&10→&8.因此,在查询处理时应该能够有效抑制统配路径的展开.

下面举两个OIQL的查询例.

*Q*1:查询李教授的信息.

Select o from CS:Teacher:Professor o where o.Name="Li"

*Q*2:查询所有作者信息.

Select o from CS:Publication:Author o

*Q*3:查询所有作者,包括博士生王的文章.

Select o from CS,Publication o where o.Author:*:Doctor Student:Name='Wang'

*Q*4:假设在另一个有关人事信息的OIM对象P(来自文件系统或Web页面)内存放有关人员的详细信息.要求查询年龄大于24岁的博士生的详细信息及发表的文章.*

Select $o1$,$o2$ from CS:*:Doctor Student $o1$, P:Person $o2$ where o1.Name=o2.Name and o2.age>24

注意,查询结构仍是OIM对象,*Q*4的结果是$o1$和$o2$的"OIM对象并".

2　OIM对象查询处理

2.1　有关符号和定义

为方便以下说明,首先引进一些符号和定义.o表示OIM对象,$Name(o)$,$TYPE(o)$和$VAL(o)$分别表示对象的名称、类型和值,对应的值域用$NAME$,$TYPE$和VAL表示.O_r特指根对象.OIM对象中相对于o的路径$p^o=o.n_1:n_2:\ldots:n_k$,$k\geqslant 1$,其中$n_1\in NAME\cup\{*,?\}$,$1\leqslant i<k$,$n_k\in NAME$,表示从节点o出发依次经过对象名是$n_1:n_2:\ldots:n_k$的节点;?和*称为通配符,?匹配任何对象名称,*匹配任意长度的对象名称序列.带有通配符的路径是半结构化数据查询的重要特点,它大大方便了用户对结构不完整或不熟悉的数据进行查询.路径p的长度$|p|=k$.特别地,当$o=O_r$时,称p^{O_r}是绝对路径.O_r中的绝对路径集合记为P.

定义1(**路径展开函数**).路径展开函数$X:p^o\rightarrow\{o_1,o_2,\ldots,o_m\}$,$X$由后面的路径展开算法(算法2)得到.称$o$是扩展点.若$o=o_r$,称绝对路径$p^{O_r}$是对象$o_i(i=1,2,\ldots,m)$在$O_r$中的一条路径.对象$o_i$的所有绝对路径记为$P(o_i)$.

$X(p^o)$是所有从节点o经路径p^o所能到达的节点的集合,因为o_r中的节点个数是有限的,因此,$X(p^o)$是有限集合.求出该集合是半结构化数据查询的主要工作,相当于对关系数据库的基本物理操作表进行扫描.

定义2(**汇聚路径函数**).OIM对象o相对于o'的汇聚路径函数$K^{o'}:o\rightarrow 2^p$,$K^{o'}(o)=\{p\mid\in X(p^{o'})\}$.

$K^{o'}(o)$是所有从节点o'到达o的路径集合.由于图中存在圈,因此,P和$K(o)$可能是无穷集合.这说明了OIM中对象路径表示的多样性,因而方便用户书写查询语句,但也增加了查询和优化的难度.实际上,我们感兴趣的是$K^o(o)$中一些"等价路径"的集合,它们可以用于查询改写,有利于在路径扩展时减少工作量.

2.2　查询处理

OIM的查询处理过程是:OIQL语法检查,查询改写,生成查询计划,优化,执行,返回结果(也是OIM对象).

---

* 这里,我们假定没有相同姓名的人员,且忽略可能有的语义异构问题.

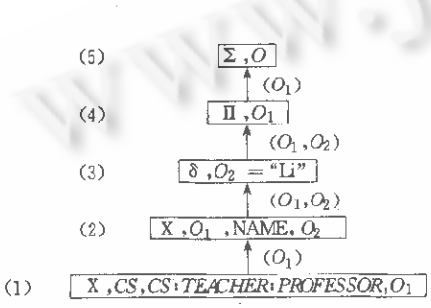
查询改写将 OIQL 语句改写为正规的查询形式，即在 where 子句中的布尔式中不含有路径，例如，$Q1$ 改写为 $Q1'$：Select o from CS:Teacher:Professor o, o.Name $o1$ where $o1$ ="Li". 这是要引进新的别名. 限于篇幅，本文不讨论查询改写.

查询计划表示成树 $T_Q=(V,E)$的形式，每个节点 $v\in V$ 都表示一种操作，称为操作节点，其操作符 $OP(v)\in\{X,\Gamma,\delta,\Pi,\Sigma\}$，分别表示节点扩展、连接、选择、投影和构造等操作.

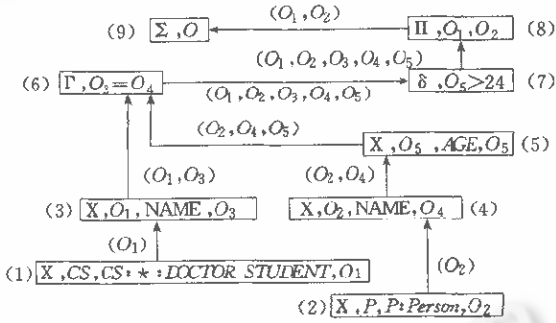
定义 3(输入变量、输出变量和产生变量). 边 e 用三元组$(v_i,v_j,\text{Output}(v_i,v_j))$表示，$\text{Output}(v_i,v_j)$是操作节点 v_i 向 v_j 的输出变量集合 $\text{Output}(v)=\bigcup_{v'\text{是}v\text{的父节点}}\text{Output}(v,v')$；操作节点 v 的输出变量集合 $\text{Input}(v)=\bigcup_{v'\text{是}v\text{的子节点}}\text{Output}(v')$. 特别地，叶操作节点的输入变量集合和根操作节点的输出变量集合为$\varnothing$. 在操作节点 v 中产生的变量记为 $\text{Gen}(v)$.

在查询树 T_Q 中，节点间传递的数据是元组集合，而输入变量和输出变量集合实际上描述了元组各域的构成. 操作节点 v 及其 $\text{Input}(v)$，$\text{Output}(v)$和 $\text{Gen}(v)$满足 $\text{Output}(v)\subseteq\text{Input}(v)\cup\text{Gen}(v)$.

节点的输出变量集合在树中表示为边的附标，为了强调节点间实际传递的是元组集合，在图中用( )表示变量集合，而不是{ }；同时，为表示清晰，用别名表示输入/输出/产生变量*. 如查询 $Q1$ 的查询树如图 2(a)所示. 计算过程从树的叶节点到根节点方向由底向上进行，各节点操作及变量传递说明如下.



(a) Q_1查询计划树



(b) Q_4查询计划树

图2

操作节点(1)：(叶节点)是路径扩展操作 $O_1=X(TEACHER:PROFESSOR^{O_{CS}})$，因为 CS 是整个 OIM 对象的根，因此扩展操作的路径的扩展节点是 O_{CS}，扩展路径是 $TEACHER:PROFESSOR$，扩展结果(节点集合)存放在变量 O_1 中；节点(5)产生的结果用输出变量集合(O_1)表示，它是元组(o_1)的集合，o_1 通过扩展节点 O_{CS}扩展得到；对应 $Q1$，得到{&5,&6}；

操作节点(2)：接收输入变量集合(O_1)，对所有的 $o_1\in O_1$，计算 $O_2=X(NAME^{O_1})$，并将 $O_1\times O_2$ 加入到结果集合中；节点(4)产生的结果用输出变量集合(O_1,O_2)表示，它是元组(o_1,o_2)的集合，o_2 通过节点 o_1 扩展得到；对应 $Q1$，得到{(&5,&13),(&6,&15)}；

操作节点(3)：进行选择操作，不产生新的变量；对应 $Q1$，得到{(&5,&13)}；

操作节点(4)：进行投影操作，不产生新的变量；对应 $Q1$，得到{(&5)}；注意，投影不产生重复元组；

操作节点(5)：进行结果构造操作，将输入集合构造成新的 OIM 对象 O_{RESULT}.

由上述分析可知，边上的输入/输出变量集合实际代表了对象元组的集合. 由于选择操作针对原子对象，因此查询处理应尽量进行路径扩展操作. 如 $Q1$ 中 where 子句的原子布尔式 o.name='Li'，在路径扩展到 o 时，并不能直接判断 o.name，因为可能在节点 o 扩展路径 name 会产生对象集合，因此生成扩展节点(4)，这也是要求进行查询改写的主要原因. $Q3$ 中的 o 类似. 由此可见，路径扩展是 OIM 查询处理中优先级最高的操作，这样有利于进行查询优化.

* 严格地讲，别名表示 OID；变量用于描述节点间传递的数据. 用同一标识符是为了表达它们的对应关系.

查询 Q_4 涉及两个 OIM 对象的操作,查询计划如图 2(b)所示.其中节点(6)是连接操作,根据条件连接来自节点(3)和节点(5)的集合.连接操作不产生新的变量.

2.3 查询计划生成

查询计划生成过程根据查询语句生成树,包括操作节点和输出/输出/产生变量集合.下面给出 OIM 查询计划生成算法.这里,假定 where 子句中各原子布尔式通过 AND 连接,且每个原子布尔式最多涉及两个别名.生成过程中给每个新生成的操作节点依次编号,$i=1,2,\ldots$,相应的节点用 v_i 表示.

算法 1. 查询计划生成

输入:经匹配的、正规的 OIQL 查询语句 Q;

输出:Q 的查询计划 T_Q;

步骤:

(1) 为 from 子句内每个路径 p 及其对象别名 O 构造扩展操作节点 X_O,过程如下:

(1.1) 若 p 是绝对路径,生成以根对象为扩展节点的扩展操作叶节点,输出变量是 $\{O\}$,产生变量是 $\{O\}$;

(1.2) 若 p 是另一对象别名 O' 的相对路径,生成以 O' 为扩展节点的扩展操作节点 X_O,编号 i,生成有向边 $v_j\rightarrow X_O$,$j=Newest(O',i)$,其中 $Newest(O',i)=\max\{j\mid j<i$ 且 $O'\in \mathrm{Output}(V_j)\}$;产生变量是 $\{O\}$;输出变量是 $\mathrm{Output}(X_j)\cup\{O\}$;

(2) 为 where 子句中每个原子布尔式 B 构造选择或连接操作节点 δ_O^B 或 $\Gamma^B_{O_1,O_2}$,过程如下:

(2.1) 若 B 仅涉及一个别名 O,生成选择操作节点 δ_O^B,编号 i;生成有向边 $v_j\rightarrow\delta_O^B$,$j=Newest(O,i)$;产生变量是 $\varnothing$;输出变量 $\mathrm{Output}(v_j)$;

(2.2) 若 B 仅涉及两个别名 O_1,O_2,生成连接操作节点 $\Gamma^B_{O_1,O_2}$,编号 i;生成有向边 $v_j\rightarrow\Gamma^B_{O_1,O_2}$ 和 $v_k\rightarrow\Gamma^B_{O_1,O_2}$,$j=Newest(O_1,i)$,$j=Newest(O_2,i)$;产生变量是 $\varnothing$;输出变量 $\mathrm{Output}(v_j)\cup\mathrm{Output}(v_k)$;

(3) 构造自然连接操作节点 Γ_{v_j,v_k},过程如下:

(3.1) 令 $NoParent=\{v\mid$不存在 $v',v'\rightarrow v\}$,若存在 $v_j,v_k\in NoParent$,且 $\mathrm{Output}(v_j)\cap\mathrm{Output}(v_k)\neq\varnothing$,生成自然连接操作节点,$\Gamma_{v_j,v_k}$,连接条件是 v_j 和 v_k 的元组集合在域 $\mathrm{Output}(v_j)\cap\mathrm{Output}(v_k)$ 处相等;生成有向边 $v_j\rightarrow\Gamma_{v_j,v_k}$ 和 $v_k\rightarrow\Gamma_{v_j,v_k}$;产生变量是 $\varnothing$;输出变量 $\mathrm{Output}(v_j)\cup\mathrm{Output}(v_k)$;

(3.2) 重复(3.1)直至无满足自然连接的节点存在;

(4) 构造投影节点 Π,过程如下:

(4.1) 令 S 表示 select 子句中别名的集合,若 $v\in NoParent$ 且 $\mathrm{Output}(v)\cap S\neq\varnothing$,生成投影节点 $\Pi_{\mathrm{Output}(v)\cap S}$;生成有向边 $v\rightarrow\Pi_{\mathrm{Output}(v)\cap S}$;产生变量是 $\varnothing$;输出变量是 $\mathrm{Output}(v)\cap S$;

(4.2) 重复(4.1)直至无满足投影条件的节点存在;

(5) 构造结果构造节点 Σ,产生有向边 $v\rightarrow\Sigma$,$v\in NoParent$;产生变量是 O_{RESULT}.

3 OIM 对象查询优化

OIM 中查询优化包括静态优化和路径扩展优化方式.静态优化是对查询计划树实施等价变换,主要方法有:(1) 下压选择和投影节点,例如,图 2(b)中的选择操作节点(7)可以下压;(2) 短路操作,这是 OIM 对象查询的一个重要特点,所谓短路是指,路径扩展结果的子集满足条件后就无需继续扩展.例如,查询书中目录包含 Database 的书籍.

Q5: select o from Store:Book o where $o.*$:subtitle= 'Database'

在对象 o 处扩展路径 *:title 所得到的集合可能很大,但只要其中有一个对象 o 满足条件 o= 'Database',则无需继续扩展.

由查询计划可知,路径扩展操作是 OIM 查询处理的基本操作,算法 2 给出了利用广度优先(breadth first search,简称 BFS)搜索方法的路径展开过程.

算法 2. 路径展开算法*

输入:OIM 对象中相对 o 的路径 $p^o=o.n_1:n_2:\ldots:n_k$

输出:从 o 开始经路径 p^o 所能到达的节点集合 $X(p^o)$

步骤:

(1) $o.level\leftarrow 0$;$Queue\leftarrow\{o\}$;$o_c=Queue.Get$; // 取并移去队列 $Queue$ 的第 1 个元素

(2) for $i=1$ to k do

(2.1) while ($o_c.level=i-1$) do

(2.1.1) if $n_i\in NAME$ then for o_c的每个亲子对象 o_c' do

(2.1.1.1) if $NAME(o_c')=n_i$ and $o_c'.state\neq checked$ then $Queue.Add(o_c')$; $o_c'.level\leftarrow o_c'.level+1$; $o_c'.state=checked$

(2.1.2) if $n_i=?$ then for o_c 的每个亲子对象 o_c' do

(2.1.2.1) if $o_c'.state\neq checked$ then $Queue.Add(o_c')$; $o_c'.level\leftarrow o_c.level-1$; $o_c'.state=checked$

(2.1.3) if $n_i=*$ then 以 o_c 为扩展节点,利用 BFS 扩展任意长度直至找到所有节点 o_c', $o_c'.state\neq checked$ 且 $Name(o_c')=n_{(i+1)}$, $Queue.Add(o_c')$;$o_c'.level\leftarrow o_c.level+2$; $o_c'.state=checked$

(2.1.4) $o_c=Queue.Get$

(3) return $Queue$.

由于图中圈的存在和统配路径,路径扩展结果可能很大且较费时,特别是步骤(2.1.3),需要对 * 进行 BFS 扩展.显然,短路方法可以和扩展操作结合,从而避免不必要的进一步扩展.另外,LORE[4]中的值索引 Vindex 和标号索引(Lindex)技术也可以用于 OIM 查询优化.除此之外,本文介绍几种其他的针对提高扩展操作的查询优化方法.

3.1 路径索引

路径索引 Pindex 将路径 p 映射到 $X(p)$,可以用 Hash 表实现,这样,在扩展集合时可以直接通过查表得到,可以大大加快路径扩展操作速度.查询处理器在生成查询计划时会根据路径有无索引生成相应的操作节点(X 节点或 Pindex 节点).路径索引有两种创建方式:人工创建或由系统自动创建.数据源管理员通过 OIQL 的 DDL 定义索引;系统自动创建是系统根据统计得出的访问频率大的路径,在路径访问频率到达某一固定值 f_H 时,查询处理器会自动为该路径建立 Pindex.

路径索引适于表示含通配符的路径,尤其适合绝对路径,这样可以很快地在查询计算的初始阶段扩展到局部的节点集合.

3.2 层次索引

定义 4(层次). 对象 o 的层次 $level(o)=\min(\{k\mid k-|p|,p\in P(o)$且 p 不含通配符$\})$.

层次索引 Lvindex 将层次和对象名的二元组映射到对象 oid,实现对象的快速定位.Lvindex 特别适合在不同类对象中所含的对象名没有重复的情况,例如,在 Versatile 中从关系数据库(RDBWrapper)和文件系统(FSWrapper)中的半结构化数据.层次路径只能扩展绝对路径中无通配符的部分,但是通过拓广可以适用于相对路径.例如,有描述电子邮件的 OIM 对象 Email,欲扩展路径 Email:Address:To:Country,利用 Lvindex(3,'Country')可以得到节点集合.

3.3 利用数据源知识

在解空间搜索过程中,利用特定问题的知识以减少搜索范围是一个普遍准则.同样,在半结构化数据源的查询中,虽然数据的结构隐含和半完整,但是针对某种数据源,其结构或多或少地存在一些结构或语义上的约束,我们称此为数据源知识.充分利用数据源知识可以提高在查询处理特别是路径扩展的效率.为此,我们定义一种简单的描述语言(knowledge definition language,简称 KDL)用于描述数据源知识.KDL 是一些规则的序列,每

---

* 算法未考虑错误路径的情况,实际计算时对错误路径仅需返回空集.

一个规则实际约束了路径的组成.如图1中的一条规则是:Project:Member,说明Member在路径中仅出现在Project之后(这里除去通配符*和?);而TEACHER:Professor不是规则.利用KDL可以在路径扩展的前期去掉不必要的搜索空间和发现错误路径.另外,KDL在某种程度上可以用于构造基于路径的视图,通过在规则中添加"虚拟规则",可以欺骗查询处理器认为某路径非法而拒绝查询处理,这时规则中需用到通配符.

另外,下面举例说明利用HTML语法的Web数据源通用KDL.这里,Web数据源的半结构化模型是从标记图得到的,简言之,标记图是从HTML标记导出的一种通用的Web数据源的半结构化模型.

(1) HEAD:TITLE

(2) TABLE:TR

(3) ...

显然,利用KDL得到的结构约束完全可以从包装器构造OIM对象时动态获得,但是要获得数据源OIM的"完备"的规则集合,其运算量是庞大的,且元数据存储量很大.

4 结束语

半结构化数据的查询及相关技术是异构数据源集成和网络资源共享的一个研究重点.本文详细介绍了Versatile中对半结构化数据的查询和优化策略,给出了查询计划的构造算法和3种针对减少路径扩展代价的优化方法.由于OIM采用带根连通有向图形式,因此,本文的技术同样适用于其他基于相同或相似结构的半结构化数据源模型,如OEM.我们进一步的研究是基于半结构化数据源的视图技术、扩展KDL以及KDL的自动构造和维护技术.

参考文献

1 Serge Abiteboul. Querying semi-structured data. In: Foto Afrati, Phokion Kolaities ed. Lecture Notes in Computer Science 1186, Database Theory——ICDT'97. New York: Springer-Verlag, 1997. 1～18

2 Wang Ning, Chen Ying, Yu Ben-quan *et al*. Versatile: an extensible integration system for heterogeneous data based on CORBA. In: Zhou Li-zhu ed. Proceedings of the 1997 IEEE International Conference on Intelligent Processing Systems (ICIPS'97). Beijing: International Academic Publishers, 1997. 1589～1593

3 Chawathe S, Garcia-Molina H, Hammer J *et al*. The TSIMMIS project: integration of heterogeneous information sources. In: Proceedings of the 10th Anniversary Meeting of the Information Processing Society of Japan. 1994. 7～18

4 McHugh J, Abiteboul S, Goldman R *et al*. Lore: a database management system for semistructured data. ACM SIGMOD, 1997,26(3):54～66

5 Buneman P, Davidson S, Fernandez M *et al*. Adding structure to unstructured data. In: Afrati Foto, Kolaities Phokion ed. Lecture Notes in Computer Science 1186, Database Theory——ICDT'97. New York: Springer-Verlag, 1997. 336～350

6 Goldman Roy, Widom Jennifer. DataGuides: enabling query formulation and optimization in semistructured databases. In: Jarke Matthias, Carey Michael, Dittrich Klaus R eds. Proceedings of the 23rd Very Large Database Conference. San Francisco: Morgan Kanfmann Publishers, Inc., 1997. 436～445

7 Buneman P, Davidson S, Hillebrand G *et al*. A query language and optimization techniques for unstructured data. In: Jagadish H V, Inderpal Singh Mumick eds. Proceedings of the ACM SIGMOD Conference'96. New York: Association for Computing Machinery, Inc., 1996. 505～516

8 陈澄,王宁,俞本权等.异构数据源系统中半结构化数据的存取与表示服务.计算机科学,1998,25(增刊):205～207
(Chen Ying, Wang Ning, Yu Ben-quan *et al*. Access and representation service for semistructured data in heterogeneous data sources system. Computer Science, 1998,25(supplement):205～207)

9 Fernandez M, Suciu D. Cptimizing regular path expressions using graph schemas. In: Urban S D, Bertino Elisa ed. Proceedings of the 14th International Conference on Data Engineering. Los Alamitos: The Printing House, 1996. 14～23

10 王宁,徐宏炳,王能斌.基于带根连通有向图的对象集成模型及代数.软件学报,1998,9(12):894～898

(Wang Ning, Xu Hong-bing, Wang Neng-bin. A data model and algebra for object integration based on a rooted connected directed graph. Journal of Software, 1998,9(12):894～898)

11 陈滢,王宁,俞本权.异构数据源系统中包装器研究和实现.见:傅育熙编.第7届全国青年计算机工作者会议(NCYCS'98).上海:上海科学技术文献出版社,1998.352～359

(Chen Ying, Wang Ning, Yu Ben-quan *et al*. Wrapper for heterogeneous data sources system——research and implementation. In: Fu Yu-xi ed. Proceedings of the 7th National Conference of Youth Computer Scientist. Shanghai: Shanghai Scientific and Technological Literature Publishing House, 1998. 352～359)

Querying and Optimizing Semistructured Data

CHEN Ying WANG Neng-bin

(*Department of Computer Science and Engineering Southeast University Nanjing* 210096)

Abstract Semistructured data has irregular or incomplete structure. In recent research on semistructured data sources and integration for heterogeneous data sources, models for semistructured data are based on direct graph with root vertex, so querying semistructured data is equivalent with searching in graph. In addition, path with wildcard characters brings more complexity in query processing. In this paper, the authors present the strategies deployed in querying and optimizing OIM (model for object integrating) data in Versatile——a system for integrating heterogeneous data sources. Algorithms for generating query plan and extending path are discussed in detail and three optimization methods, path index (Pindex), level index(Lvindex) and knowledge of data source are introduced. Also the approach can be applicable to other graph-based semistructured data easily.

Key words Semistructured data, query processing, optimization.