

量词约束 谓词演算 复合对象 数据库

82-85

复合对象选择条件中的量词约束及谓词演算

Quantity Limitation and Predicate Calculus in Selection Condition of Composite Objects

钟武 胡守仁

(国防科技大学 计算机系 长沙410073)

TP311.13

摘要 This paper gives the quantity-limiting relationships of complex objects lying in different nesting lays and their expression methods, which can be translated into object predicate calculus by means of our methods. The expression methods can express a kind of complex selection queries.

关键词 Composite object, Selection query, Object predicate calculus

1. 前言

为表征对象间复杂的关系,克服传统的关系模型所带来的限制,OODB 数据库技术得以迅速发展。然而,由于 OODB 中引入的类之间存在着 'is-a' 和 'is part-of' 关系,这就导致在定义 OODB 的查询模型方面存在着许多困难。

对复杂对象查询的表达方法已有许多文献从类似于 SQL 方面、代数方面和谓词演算方面进行了广泛的研究。对于选择查询,在借鉴传统的选择查询的基础上,引入了对象嵌套路径的概念和对象消息的机制。但是,对不同嵌套层次上的满足一定条件的对象之间存在的量的约束关系却研究甚少。一般局限于全称量词“ \forall ”和存在量词“ \exists ”。

在传统的关系数据库中,对数据库的操作的基础研究主要有:代数操作和关系演算。后者同时也是前者的数理逻辑基础。在对数据库的操作上,后者比前者有更高的非过程性,因为后者只需要用户以谓词的形式提出自己对运算结果的具体目标即可,而如何实现的任务交系统解决。对于选择查询,由于关系模型的平坦和简洁特性,选择查询的谓词演算与选择代数操作之间的关系是非常显见的。在 OODB 中,由于对象嵌套的复杂性以及不同层次上的满足一定条件的复杂对象间的复杂的量的约束,致使复杂对象的选择操作与相应的对象谓词演算关系较为复杂。因此,研究复杂对象的选择操作与对象谓词演算的转换方法是非常重要的,但这一点在广泛的 OODB 的查询研究中却非常少见。

为加强 OODBMS 对复杂对象选择查询的表达能力,以下我们将讨论处于对象复合路径上的满足

一定条件的对象间存在的量的约束关系及其表达方法。与此同时,讨论将该类选择查询转化成对象谓词演算的问题。

2. 复合对象

为第3节的讨论,本节着重描述 OODB 中用元组和集合构造复合对象的问题。至于 OODB 中的其它概念,读者可参阅有关文献。

设: δ 为无限的属性名集, Ω 为系统提供的基本数据类型名(如:integer, real 等)集。这里 Ω, δ 不相交。

定义1 OODB 的模式 Σ 为三元组 (L, \leq, ϕ) 。

其中:

L 为用户定义的类的类名集。类名集合为 $L \cup \{Object, Set, Tuple\}$, 该集合与 Ω, δ 不相交;

\leq 是一个偏序关系,用于表征类名所对应类之间的继承关系;这里有: $Set \leq Object, Tuple \leq Object$;

$\phi: L \rightarrow \Gamma$, 将类名 $C (\in L)$ 映射为一个二元组 (ST, M) 。其中, M 是作用于 ST 上的方法集, ST 的定义是:①对于属性名集 $\{A_1, A_2, \dots, A_k\} \subset \delta, T_1, T_2, \dots, T_k \in \Omega \cup L$ 。这里, $T_i (i=1, \dots, k)$ 不为 C 。 $ST = [A_1: T_1, A_2: T_2, \dots, A_k: T_k]$, 即: ST 为元组结构。此时,我们称类 C 为元组类;②对于 $T \in L$, 且 $T \leq Tuple, ST = \{T\}$, 即: ST 为元组集合结构。此时,我们称类 C 为元组集合类。

定义2 ①函数 Struct 用于求类的结构,函数 Method 用于求类的操作方法集;②对于集合类 P , 有 $Struct(P) = \{T\}, T \in L$ 且 $T \leq Tuple$, 则函数 $Base(P) = T$;③对于元组类 P , 有 $Struct(P) = [A_1:$

$T_1, A_2 : T_2, \dots, A_k : T_k$], 则函数 $Att(P) = \{A_1, A_2, \dots, A_k\}$;

定义3 ①对 $D \in \Omega, \|D\|$ 是系统提供的基本类型 D 的值的集合; ②对 $C \in L, \|C\| = \{O | O \text{ 是类 } C \text{ 的对象实例或是类 } C \text{ 的子类的对象实例}\}$ 。由于继承性, 当 $C \leq C'$ 时, 有 $\|C\| \subseteq \|C'\|$; ③若 P 为元组类, $Struct(P) = [A_1 : T_1, A_2 : T_2, \dots, A_k : T_k]$, 则 $\|P\| = \{[A_1 : V_1, A_2 : V_2, \dots, A_k : V_k] | V_i \in \|T_i\| (i=1, \dots, k)\}$; ④若 P 为集合类, $Struct(P) = \{T\}, T \in L$ 且 $T \leq Tuple$, 则 $\|P\| = Powerset(\|T\|)$ 。

3. 选择条件

3.1 选择条件使用实例及说明

为突出重点, 我们在文法中只给出连接词 And。至于谓词公式中的 Or 和 Not, 读者可在文法中自行添加。选择条件的文法如下:

选择条件 \rightarrow Where 限定表达式

限定表达式 \rightarrow 限定项 [And 限定表达式]

限定项 \rightarrow 对象限定 | 比较算式 | 比较算式

对象限定 \rightarrow (量词 对象变量名 Of 元组类名 Through 路径 选择条件) [对象限定]

量词 \rightarrow No | Existing | At least 数字 | At most 数字 |

Just 数字 | 数字 To 数字 | Each

路径 \rightarrow 键 [, 元组类名, 路径]

键 \rightarrow 元组类名. 属性名

图1是有关 TRADER 销售商的对象复合模式。箭头线的起始端表示相应属性嵌入的对象类应为集合类, 箭头线终端所指的类为集合对象所含对象的类(元组类)。例如, ITEM 类的 is_sold 属性定义为 {SALE}。线段的一端表示相应属性嵌入的对象类应为元组类, 另一端指示具体的元组类。例如, ITEM 类的 description 属性定义为 DETAIL 类。

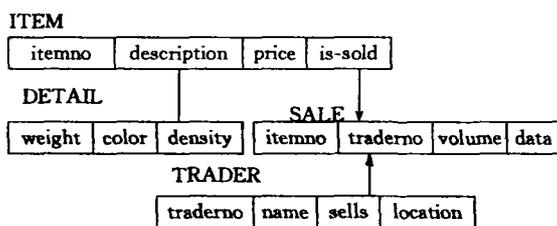


图1 对象复合模式

例 选择满足如下条件的销售商 k_1 : 对至少5种其销售的颜色为红的商品有, 每次对它的销售有, 销售金额高达10000元。选择条件为:

Where (At least 5 k_2 Of ITEM Through TRADER.

SELLS, SALE, ITEM. is_sold Where k_2 . description. color = "红") (Each k_3 Of SALE Through TRADER. SELLS Where $k_3 \in k_2$. is_sold) (k_2 . price \times k_3 . volume $>$ 10000)

其中, 内层条件 "Where k_2 . description. color = "红"" 是用于限定商品 k_2 的, 内层条件 "Where $k_3 \in k_2$. is_sold" 是用于限定销售 k_3 。当有多个嵌入最外层 Where 条件的内层 Where 条件存在时, 内层 Where 限定的对象按如下方式确定: 内层 Where 条件总是限定紧靠其左边的对象变量中的对象。

例中的关键字 Through 用于引入连接两元组对象的路径。路径用于描述两元组对象间存在的关系, 其语义由对象模式的设计者确定。路径的一端是紧靠 Through 关键字左边的元组对象变量中的对象, 另一端是 Through 直接所处的 Where 条件所限定的元组对象。就上例来说, 路径 TRADER. SELLS, SALE, ITEM. is_sold 的一端是 k_2 , 因为它紧靠 Through 的左端; 另一端是销售商 k_1 , 因为 Through 所处的当前 Where 条件层是限定 k_1 的。此路径对 k_2 进行限定, 使得 k_2 的取值局限于销售商 k_1 所销售的商品范围内, 而不是任意一个 $\|ITEM\|$ 中的商品对象。同理, 路径 TRADER. SELLS 关联 k_1 和 k_3 。

在上例中, 它含两个对象限定 ("At least...") (以后简称 #1限定) 和 "(Each...)" (以后简称 #2限定), 以及一个比较算式 " k_2 . price \times k_3 . volume $>$ 10000" (以后简称 expression)。细心的读者将发现, 这和离散数学中有关变元约束表达式非常相似。此处的 expression 就是量词的作用域, 而量词则做了扩充。

对于两个或多个对象间的比较算式不是本文要着重讨论的内容, 以下我们将比较算式简写为 COMEX。下面我们将集中精力讨论选择条件中通过路径关联的满足一定条件的对象之间的量的约束关系将如何转化成对象谓词演算的问题。

3.2 选择条件的语义

本节采用谓词逻辑描述选择条件的语义。

3.2.1 路径

定义4 类 $X', X, Y \in L$, 有 $X' \leq Set, X, Y \leq Tuple, X = Base(X')$ 。属性 $A \in Att(Y)$ 。若 A 的定义形式为 $A : X'$, 则称 $Y.A$ 为类 X 与类 Y 之间的集合聚合键。若 A 的定义形式为 $A : X$, 则称 $Y.A$ 为类 X 与类 Y 之间的元组聚合键。

聚合键反映了相应两元组对象之间通过属性聚合的模式。

约定 对于元组对象 t, A 为 t 的属性, $t.A = t'$ 表示 t 的属性 A 中嵌入了对象 t' 。

定义5 1)类 $C_1, C_n (\leq \text{Tuple})$ 之间的路径形式为 $\bar{C}_1, \bar{C}_2, \bar{C}_3, \dots, \bar{C}_{n-1}, \bar{C}_n$ 。其中 $C_i \in L$ 且 $C_i \leq \text{Tuple} (i=2, \dots, n-1)$; \bar{C}_i 是类 C_i 与类 C_{i+1} 之间的聚合键 ($i=1, \dots, n-1$)。2) \bar{C}_1 是类 C_1, C_2 的路径, $t_1 \in \|C_1\|, t_2 \in \|C_2\|$ 。若 \bar{C}_1 是元组聚合键, 当 \bar{C}_1 为 $C_1.A$ 时, 有 $t_1.A = t_2$ 或当 \bar{C}_1 为 $C_2.A$ 时, 有 $t_2.A = t_1$, 则称 t_1 与 t_2 是关于路径 \bar{C}_1 连接的。若 \bar{C}_1 是集合聚合键, 当 \bar{C}_1 是 $C_1.A$ 时, 有 $t_2 \in t_1.A$ 或当 \bar{C}_1 为 $C_2.A$ 时, 有 $t_1 \in t_2.A$, 则称 t_1 与 t_2 是关于路径 \bar{C}_1 连接的。3) $\bar{C}_1, \bar{C}_2, \bar{C}_3, \dots, \bar{C}_v, \bar{C}_v$ 是类 C_1 与类 C_{v+1} 间的路径, 记为 path1 ; $\bar{C}_{v+1}, \bar{C}_{v+2}, \bar{C}_{v+3}, \dots, \bar{C}_{n-1}, \bar{C}_n$ 是类 C_{v+1} 与类 C_n 间的路径, 记为 path2 。 $t_1 \in \|C_1\|, t_{v+1} \in \|C_{v+1}\|, t_n \in \|C_n\|$ 。若 t_1 与 t_{v+1} 是关于 path1 连接的, t_{v+1} 与 t_n 是关于 path2 连接的, 则 t_1 与 t_n 是关于路径 $\text{path1}, C_{v+1}, \text{path2}$ 连接的。

根据上述定义, 我们有: 若路径中的聚合键全部是元组聚合键, 则一个元组对象关于该路径连接的元组对象最多一个。若一个元组对象关于该路径连接的元组对象至少两个, 则路径中至少有一个集合聚合键存在。

定义6 对于路径 P 和 t_1, t_2 两个元组对象, t_1, t_2 两元组对象关于路径 P 的谓词 $\text{relation}(t_1, t_2, P)$ 的取值为: 当 t_1, t_2 是关于 P 连接时, $\text{relation}(t_1, t_2, P) = \text{true}$; 否则, $\text{relation}(t_1, t_2, P) = \text{false}$ 。

3.2.2 对象限定转换 我们简写对象限定为 $(Q \ t \ C \ P \ W)$ 。其中: Q 是量词, t 是对象变量名, C 是元组类名, P 是路径, W 是作用于 t 的选择条件。

设限定项 $(Q_2 \ t_2 \ C_2 \ P_2 \ W_2) \dots (Q_m \ t_m \ C_m \ P_m \ W_m)$ COMEX 所处的限定条件层通过关键字 Where 紧跟并作用于 t 。转化时, 从限定项的最右端开始。我们把 COMEX 看成一种特殊的对象限定, R_i 为将限定项右端开始的 $m+2-i$ 个元组对象限定转化成的谓词公式 ($i=2, \dots, m+1$)。

例如: 就上例来说, t 就是 k_1 ; #1限定 #2限定 expression 相当于 $(Q_2 \ t_2 \ C_2 \ P_2 \ W_2) (Q_3 \ t_3 \ C_3 \ P_3 \ W_3)$ COMEX, 且它们一一对应。在 #1限定与 $(Q_2 \ t_2 \ C_2 \ P_2 \ W_2)$ 对应中, Q_2 是 At least 5, t_2 是 k_2 , C_2 是 ITEM, P_2 是 TRADER.sells, SALE, ITEM.is_sold, W_2 是 k_2 .description.color="红"; #2限定与 $(Q_3 \ t_3 \ C_3 \ P_3 \ W_3)$ 一一对应的关系可类推; COMEX 是 expression, R_4 表示将 COMEX 转化成的谓词公式: k_2 .price \times k_3 .volume > 10000; R_3 表示将 $(Q_3 \ t_3 \ C_3 \ P_3 \ W_3) R_4$ 转化成

的谓词公式; R_2 表示将 $(Q_2 \ t_2 \ C_2 \ P_2 \ W_2) R_3$ 转化成的谓词公式。

从上面的分析我们可看到: 若已取得转化好的谓词公式 R_{i+1} , 我们就可以将 $(Q_i \ t_i \ C_i \ P_i \ W_i) R_{i+1}$ 转化成谓词公式 R_i 。如此反复, 直至全部转化完毕, 最终获得谓词公式 R_2 。将 $(Q_i \ t_i \ C_i \ P_i \ W_i) R_{i+1}$ 依 Q_i 的不同转化成谓词公式 R_i 的方法如下:

• $Q_i = \text{Existing}$

公式: $R_i = (\exists t_i \in \|C_i\|) (\text{relation}(t, t_i, P_i) \wedge \text{satisfy}(t_i, W_i) \wedge R_{i+1})$

就上例来说, 假设 R_3 是从 #2限定 expression 转化过来的谓词公式, 其语义为: 对上文提及的商品 k_2 , 每次销售商 k_1 对商品 k_2 的销售 k_3 有, 销售额在 10000 元以上。

将 #1限定中的量词 At least 5 改为 Existing 后, 从 #1限定 R_3 转化过来的谓词公式 R_2 为: $(\exists K_2 \in \|\text{ITEM}\|) (\text{relation}(k_1, k_2, P) \wedge \text{satisfy}(k_2, W) \wedge R_3)$, 其中, W 为: k_2 .description.color="红", P 为 TRADER.sells, SALE, ITEM.is_sold。从公式来解释的话, 其含义为: 存在商品 k_2 , 销售商 k_1 销售商品 k_2 且商品 k_2 价格在 50 元以下且每次销售商 k_1 对商品 k_2 的销售 k_3 有, 销售额在 10000 元以上。

• $Q_i = \text{Each}$

公式: $R_i = (\forall t_i \in \|C_i\|) (\text{relation}(t, t_i, P_i) \wedge \text{satisfy}(t_i, W_i) \rightarrow R_{i+1})$

该量词有“每次”, “每个”, “所有”的含义。就上例来说, 将 #1限定中的量词 At least 5 改为 Each 后, 对销售商 k_1 限定的语义为: 对每个商品 k_2 , 若销售商 k_1 销售商品 k_2 且商品 k_2 价格在 50 元以下, 则每次销售商 k_1 对商品 k_2 的销售 k_3 有, 销售额在 10000 元以上。

• $Q_i = \text{No}$

公式: $R_i = \neg (\exists t_i \in \|C_i\|) (\text{relation}(t, t_i, P_i) \wedge \text{satisfy}(t_i, W_i) \wedge R_{i+1})$

就上例来说, 将 #1限定中的量词 At least 5 改为 No 后, 对销售商 k_1 限定的语义为: 没有商品 k_2 , 销售商 k_1 销售商品 k_2 且商品 k_2 价格在 50 元以下且每次销售商 k_1 对商品 k_2 的销售 k_3 有, 销售额在 10000 元以上。

• $Q_i = \text{At least } n$

公式: $R_i = |\{t_i \mid t_i \in \|C_i\| \wedge \text{satisfy}(t_i, W_i) \wedge \text{relation}(t, t_i, P_i) \wedge R_{i+1}\}| \geq n$

其含义为: “至少 n ”。

• $Q_i = \text{At most } n$

公式: $R_i = |\{t_i | t_i \in \|C_i\| \wedge \text{satisfy}(t_i, W_i) \wedge \text{relation}(t, t_i, P_i) \wedge R_{i+1}\}| \leq n$

其含义为:“至多 n”。

• $Q_i = \text{Just } n$

公式: $R_i = |\{t_i | t_i \in \|C_i\| \wedge \text{satisfy}(t_i, W_i) \wedge \text{relation}(t, t_i, P_i) \wedge R_{i+1}\}| = n$

其含义为:“恰好 n”。

• $Q_i = n_1 \text{ To } n_2$

公式: $R_i = n_1 \leq |\{t_i | t_i \in \|C_i\| \wedge \text{satisfy}(t_i, W_i) \wedge \text{relation}(t, t_i, P_i) \wedge R_{i+1}\}| \leq n_2$

其含义为:“n₁至 n₂个”。

谓词 $\text{satisfy}(t, W)$ 表示: t 满足选择条件 W。当 $W = \epsilon$ 时, $\text{satisfy}(t, W)$ 取值为真。

3.2.3 转换实例 本节给出以谓词公式描述的复杂选择条件的形式化的语义。

转换规则:将选择条件中每一个由 Where 引导的选择条件及其所限定的元组对象变量名 t 转化成谓词公式 $\text{satisfy}(t, W)$ 。而 $\text{satisfy}(t, W)$ 的获得,可通过将选择条件 W 根据3.2.2节所述的转化方法以及将 W 中的 And, Or, Not 分别变换为 \wedge, \vee, \neg 而获得。对整个选择条件来说,转化的结果是嵌套形式的谓词公式。

例 将3.1节中的例子的选择条件转化为如下嵌套形式的谓词公式:

$\text{satisfy}(k_1, W_1) = (|\{k_2 | (k_2 \in \|ITEM\|) (\text{satisfy}(k_2, W_2) \wedge \text{relation}(k_1, k_2, P_2) \wedge (\forall k_3 \in \|SALE\|)$

$(\text{satisfy}(k_3, W_3) \wedge \text{relation}(k_1, k_3, P_3) \rightarrow (k_2.\text{price} \times k_3.\text{volume} > 10000))\}| \geq 5)$

$\text{satisfy}(k_2, W_2) = (k_2.\text{description}.\text{color} = \text{“红”})$

$\text{satisfy}(k_3, W_3) = (k_3 \in k_2.\text{is_sold})$

其中:

• W_1 是: (At least 5 k_2 Of ITEM Through TRADFR. sells, SALE, ITEM. is_sold Where k_2 .

description. color = “红”) (Each k_3 Of SALE Through TRADER. sells Where $k_3 \in K_2$. is_sold)) (k_2 . price \times k_3 . volume > 10000)

• W_2 是: k_2 . description. color = “红”

• W_3 是: $k_3 \in K_2$. is_sold

• P_2 是: TRADER. sells, SALE, ITEM. is_sold

• P_3 是: TRADER. sells

结束语 在 OODB 的选择查询中,查询的表达力即选择条件语义的表达力至关重要。从元组对象和集合对象的复合关系出发,我们以文法的形式给出了选择查询中不同层次上的对象间的量的约束关系的表达方法;描述了将此复杂选择操作转化成对象谓词演算的方法。由于一条选择操作语句能表达一类基于路径和复杂量词限定的复杂查询语义,因此,本文提出的表达方法具有较强的非过程性。

主要参考文献

- [1] The Object Database Standard, ODMG-93, edited by R. G. G. Cattell, Morgan Kaufmann Publishers
- [2] V. D. Bussche and A. Heuer, Using SQL with Object-Oriented Database. Info. Sys., 18(7)1993
- [3] V. D. Bussche, Evaluation and Optimization of complex object selection, Lecture Notes in Computer Science, No. 566
- [4] M. Roth et al., Extended algebra and calculus for nested relational databases, ACM Trans. Database Sys., 13(4)1988
- [5] N. Bhalla and S. Balasundaram, Operation and Queries in Object-Oriented Database Supporting Complex Objects, Information and Software Technology, 35(1)1993
- [6] 钟武、胡守仁, OODB 数据模型及查询代数, 计算机科学, 24(2)1997

(上接第88页)

- [2] Stephanie Forrest et al., Genetic Algorithm: Principle of Natural Selection Applied to Computation, Science, 1993
- [3] V. Kreinovich et al., Genetic Algorithm: What Fitness Scaling is Optimal?, Cybernetics and Systems, 24(1)1991
- [4] X. Yao, A review of evolutionary artificial neural net-

work, Intel. J. of Intell. Systems, 1993, 8

- [5] Philipp Schmid, The Mapping Problem: A Neural Network Approach, Intel. Neural Network Conference, 1990, Paris, France
- [6] Hwang, K. and Ghoshu, J., Critical Issues in Mapping Neural Networks on Message-Passing Multi-computers, Intel. Symposium on Computer Architecture, ACM/IEEE 1988