# OODB 数据模型及查询代数

钟 武 胡守仁

(国防科技大学计算机系 长沙 410073)

**摘 要** On the basis of the relational model, relaxing the restruction on that the number of attributes held by every tuple in a relation set must be identical and introducing inheritance and encapsulation mechanisms of OO techniques, this paper describes an OODB model formally by means of establishing Tuple class and Set class. According to the model, 17 query algebra operators which are closure and safety, are defined in the form of predvate calculus.

**关键词** OODB model, Query algebra, Predicate calculus.

## 一、引言

传统的关系模型打破 1NF 限制，已发展成嵌套的关系模型，能更方便地描述复杂对象之间的关系。随着 OO 技术的不断发展，人们也迫切希望将继承和封装机制引入到数据库中，使数据库模型含有更多的语义，能更自然地描述复杂对象间的关系。封装使得元组与在该元组上的操作能结合在一起，元组中的属性值可以是某类复合对象，而该对象又可以是一个元组或一个基于某元组类的集合，从而能自然地描述汇聚这类语义，同时支持语义的相对性。继承使得数据库设计和编程成为可重用(向下继承)，一个元组对象能被其所属类的超类所引用(向上继承)，从而能有效地描述泛化和特化这类语义。继承概念的引入，允许基于某类元组的集合能包含属性数目不等的元组，只要其所属类是该集合基于的元组类的子类。

从上述观点出发，我们建立了一个 OODB 模型，并用谓词演算的形式描述了基于该模型的查询代数。

## 二、OODB 模型

设 $\delta$ 为无限的属性名集，$\Omega$ 为系统提供的基本数据类型名(如：integer，real 等)集，$O_t$ 为无限的元组对象集，$O_s$ 为无限的集合对象集。这里 $\Omega$、$\delta$、$O_t$、$O_s$ 两两不相交。

定义 1 OODB 的模式 $\Sigma$ 为三元组 $(\Gamma, \leq, \psi)$。其中：①类名集合为 $L \cup \{Object, Set, Tuple\}$，该集合与 $\Omega$、$\delta$、$O_t$、$O_s$ 不相交。这里，L 为用户定义的类的类名集。②$\leq$ 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，$Set \leq Object$，$Tuple \leq Object$。③$\psi: L \to \Gamma$。为类名 $C(\in L)$ 指派一个局部类型 $(ST, M)$。局部类型 $(ST, M)$ 定义如下：

设类型名集 $H = \Omega \cup L$，则 M 是作用于 ST 上的操作方法集，ST 是如下三种形式的结构：

• ST 为 $\varnothing$。表示无类型结构，此时，方法集 M

[2] IMB Inc., CIM CDF General Information Manual, Oct. 1989

[3] W. H. Inmon and C. Kelley. RdB/WMS Developing the Data Warehouse. QED Publishing Group, Boston, Massachusetts, 1993

[4] Sybase Inc., Sybase Warehouse Works 体系结构, Sybase'95 新技术, 1995

[5] N. Roussopoulos et al., The Maryland ADMS project: views R Us, IEEE Data Eng. Bulletin, 18(2), 1995

[6] G. Zhou et al., Supporting data integration and warehousing using H2O. Same to [5]

[7] J. Hammer et al., The Stanford Data Warehousing Project. Same to [5]

[8] A. Gupta & I. Mumick. Maintenance of materialized views: problems, techniques, and applications. Same to [5]

[9] The Guide to CORBA, OMG TC Document, Sept. 1994

[10] 于戈等，CIMS 信息集成平台多层集成机理的研究，第四届中国 CIMS 学术会议论文集，武汉，1994 年 10 月，pp. 75-76

[11] INFORMIX, Data Warehousing for Enterprise-wide Decision Making, Sept. 1994

中的方法通过类的继承关系而作用于其超类的相应ST结构上；

·对于属性名集$\{A_1,A_2,\cdots,A_k\}\subset\delta$，$T_1T_2,\cdots,T_k\in H$。这里，类型名$T_i(i=1,\cdots,k)$不为C。$ST=[A_1:T_1,A_2:T_2,\cdots,A_k:T_k]$，即：ST为元组结构；

·对于$T\in L$，且$T\leq Tuple$，$ST=\{T\}$，即：ST为元组集合结构。

约定：本文以下所论及的集合类型、集合类、集合对象均是针对超类Set下的元组对象集合类而言的。

定义2 ①函数Struct用于求类型的结构，函数Method用于求类型的操作方法集。②对于集合类型P，有$Struct(P)=\{T\}$，$T\in L$且$T\leq Tuple$，则函数$Base(P)=T$。③对于元组类型P，有$Struct(P)=[A_1:T_1,A_2:T_2,\cdots,A_k:T_k]$，则函数$Att(P)=\{A_1,A_2,\cdots,A_k\}$。

在OODB模型中，我们规定：同名属性只能有相同的定义。

定义3 ①设$P_1,P_2$为元组类型，若P为$P_1,P_2$的连接类型，即：$P=P_1\cdot P_2$，则类型P为元组类型，其类型结构由$Att(P_1)\cup Att(P_2)$中的属性构成，其操作方法集$Method(P)=Method(P_1)\cup Method(P_2)$。若$Method(P_1)\cap Method(P_2)\neq\varnothing$，方法体的定义遵循$P_2$元组类型中方法体的定义。②若$C\in L$，$C\leq Tuple$，设$X_c=\{C'\mid C\leq C'\wedge C'\in L\wedge C'\leq Tuple\}$，则C所拥有的类型$\bar{\psi}(C)$是一个继承类型，$\bar{\psi}(C)=\forall_{C'\in X_c}\psi(C')$。③若$C\in L$，$C\leq Set$，则C所拥有的类型就是它的局部类型，有$\bar{\psi}(C)=\psi(C)$。

本OODB模型有如下规定：

1. 对于$C\in L$，若$C\leq Tuple$，则类C表征了一种元组类型；对于$C\in L$，若$C\leq Set$，则类C表征了一种集合类型。

2. 对于$C\in L$，若$C\leq Tuple$，则对应地存在唯一一个类$C'(\in L)$，$C'\leq Set$，有$Base(\psi(C'))=C$；反之亦然。因此，Tuple类的子类类似于定义了一个关系模式，而Set类的子类的实例类似于确定了一个关系集合。

3. $C',C\in L$，$C'$，$C\leq Tuple$，$C\leq C'$ iff $Att(\bar{\psi}(C'))\subseteq Att(\bar{\psi}(C))$，$C'=C$ iff $Att(\bar{\psi}(C'))=Att(\bar{\psi}(C))$。

定义4 ①对$D\in\Omega$，$\|D\|$是系统提供的基本类型D的值的集合。②对$C\in L$，$\|C\|=\{O\mid O$是类C的对象实例或是类C的子类的对象实例$\}$。③若P为元组类型，$Struct(P)=[A_1:T_1,A_2:T_2,\cdots,A_k:T_k]$，则$\|P\|=\{[A_1:V_1,A_2:V_2,\cdots,A_k:V_k]\mid V_i\in\|T_i\|(i=1,\cdots,k)\}$。④若P为集合类型，$Struct(P)=\{T\}$，$T\in L$且$T\leq Tuple$，则$\|P\|=Powerset(\|T\|)$。

定义5 ①函数$Class:O_t\cup O_r\to L$，将对象O映射到其所属类C，即：$Class(O)=C$。②函数$At:O_t\to Powerset(\delta)$，求元组对象t的属性集。③函数$Elt:O_r\to L$，将集合对象r映射到其基类，即：$Elt(r)=Base(\bar{\psi}(Class(r)))$。

根据以上定义和OODB的规定，我们有：

·对元组对象t，$Class(t)=C$ iff $C\leq Tuple$且$At(t)=Att(\bar{\psi}(C))$。

·对元组对象t，当$t\in\|C\|$时，有$Class(t)\leq C$。因此，$At(t)$不一定等于$Att(\bar{\psi}(C))$，即：$\|C\|$中的元组对象$t_1,t_2$并不一定拥有一致的属性数目。

·对元组对象$t_1,t_2$，当$Class(t_1)=Class(t_2)=C$时，$t_1,t_2$的状态值属于$\|\bar{\psi}(C)\|$。设$A_1\in Att(\bar{\psi}(C))$，用$t_1.A_1,t_2.A_1$分别表示$t_1,t_2$在属性$A_1$上的值$V_1,V'_1$。若$A_1$定义为$A_1:T_1$，$T_1\in L$且$T_1\leq Tuple$，则$V_1,V'_1\in\|T_1\|$。要注意的是：此时元组对象$V_1,V'_1$并不一定属于同一个类，但有$Class(V_1)\leq T_1$和$Class(V'_1)\leq T_1$成立。也就是说尽管$t_1,t_2$状态值属于同一个类型，但它们的属性对象$t_1.A_1$、$t_2.A_1$的状态值并不一定属于同一个类型，即：属性对象$t_1.A_1,t_2.A_1$并不一定有相同的属性个数。

·对集合对象r，设$Elt(r)=C$，则r的状态值属于$Powerset(\|C\|)$。由于$\|C\|$中的元组对象类型并不一致，因此r中元组对象的属性个数将不一致。

我们通过指明所关心的类所具有的类型来作为操作处理的类型，以解决这种属性个数不一致性的问题。

定义6 在数据库模式$\Sigma$下的数据库实例为二元组$(Rs,Tu)$：①$Rs=\{r_1,r_2,\cdots\}$。它是集合对象$r_i$的集合，$r_i\in O_r(i=1,2,\cdots)$；②$Tu=\{t_1,t_2\cdots\}$。它是元组对象$t_i$的集合，$t_i\in O_t(i=1,2,\cdots)$。其中，对于任意$r_i\in Rs$，有$r_i\subseteq Tu$。

## 三、OODB查询代数

约定：符号"≡"表示恒等。若对象$O_1,O_2$有$O_1\equiv O_2$，则表示$O_1$和$O_2$是一个对象；

定义7 ①对于元组对象$t_1,t_2$，$t_1=t_2$表示$At(t_1)=At(t_2)$且$(\forall A\in At(t_1))(t_1.A=t_2.A)$成立。②对于集合对象$r_1,r_2$，$r_1=r_2$表示$Elt(r_1)=Elt(r_2)$且有$(\forall t_1\in r_1)(\exists t_2\in r_2)(t_1=t_2)$和$(\forall t_2\in r_2)(\exists t_1\in r_1)(t_1=t_2)$同时成立。

在定义 OODB 查询代数前，做如下说明：

• 由于集合对象所属的类与它的基类是一一对应的，因此以后讨论集合对象时，主要只关心它的基类。

• 操作产生的集合对象所属类的处理原则：代数操作都是针对集合对象进行，产生的结果是一个集合对象。该集合对象中元组对象均属于其基类。该基类与被操作集合对象的基类之间有超类与子类的继承关系。新集合对象的基类有三种情况：①该基类在模式中已定义；②该基类是模式中某些类的超类。超类的引入不会影响其子类所拥有的类型；③该基类是模式中某些类的子类。

• 对于元组对象 t，设 $A\in At(t)$，则 $t.A$ 表示通过属性 A 嵌入 t 中的对象；设 $X\subseteq At(t)$，则 $t[X]$ 表示 t 在 X 上投影产生的对象 t′。对 t′，有 $X=At(t')$ 且对 $\forall A\in X$ 有 $t.A\equiv t'.A$。

• $\{t|t\in O_t\wedge p(t)\}$ 表示由满足谓词公式 P 的元组对象 t 构成的集合对象。由于拥有相同状态值的不同对象在 $O_t$ 中有无数个，所以除谓词 P 的需要外，不允许 $O_t$ 中无数拥有相同状态的对象参加对谓词 P 的验证。因此，操作产生的集合对象是一个有限集合对象。“谓词 P 的需要”的含义为：对 P 中各种可能的对象组合，需要在 $O_t$ 中选取对象，以验证对象组合是否满足 P。

• 在元组集合对象查询的过程中，由于存在对象之间的比较，因此查询操作分为两类：一类是根据对象的恒等“≡”进行操作，二类是根据对象的值等“=”进行操作。在具体实现时，可通过对开关的设置来决定操作执行的方式。为此，在以下操作描述中，用“≌”表示“≡”或“=”的概念。对一个操作描述中的所有“≌”，要么“≌”全取“≡”，要么“≌”全取“=”。

以下是 17 种代数操作：

1)集合并∪

条件：$Elt(r_1)\leq Elt(r_2)$ 或 $Elt(r_2)\leq Elt(r_1)$

操作：$r\equiv r_1\cup r_2\equiv\{t|t\in r_1\vee t\in r_2\}$

r 的基类为：

$$Elt(r)=\begin{cases}Elt(r_1) & Elt(r_2)\leq Elt(r_1)\\ Elt(r_2) & Elt(r_1)\leq Elt(r_2)\end{cases}$$

一类操作与二类操作相同。

2)元组属性值并$\cup_t$

对元组对象 t，定义 $\bar{S}(t)=\{a|a\in At(t)$ 且 a 定义的类型是非集合类型$\}$，$S(t)=At(t)-\bar{S}(t)$。当 $r_1$ 中的元组 $t_1,t_2,\cdots,t_n$ 有 $At(t_1)=At(t_2)=\cdots=At(t_n)$ 且 $\forall A\in\bar{S}(t_1)(t_1.A\cong t_2.A\cong\cdots\cong t_n.A)$ 时，$t_1,t_2,\cdots,t_n$ 将并为一个元组对象 t。对 $\forall A\in\bar{S}(t_1)$，有 $t.A\cong t_1.A$；对 $\forall A\in S(t_1)$，有 $t.A\equiv t_1.A\cup t_2.A\cup\cdots\cup t_n.A$。

$r_1$ 上的等价关系 $R=\{(t_1,t_2)|At(t_1)=At(t_2)\wedge(\forall A\in\bar{S}(t_1))(t_1.A\cong t_2.A)\}$，$r_1$ 关于 R 的商集 $r_1/R$ 记为 Sm。

操作：$r\equiv\cup_t(r_1)\equiv\{t|t\in O_t\wedge(\exists r_3\in Sm)(\exists t_1\in r_3)(At(t)=At(t_1)\wedge(\forall Z\in\bar{S}(t_1))(t_1.Z\cong t.Z)\wedge(\forall A\in S(t_1))(t.A\equiv\bigcup_{\forall t_2\in r_3}t_2.A)\}$

r 的基类为：$Elt(r_1)$；操作产生集合对象 $r'_1,r'_1\in\{r_1|t\in r\wedge(\exists Z\in S(t))(t.Z\equiv r_1)\}$。设 $t\in r,Z\in S(t)$，有 $t.Z\equiv r'_1$，则 $r'_1$ 的类就是属性 Z 所定义的类。

3)集合交∩

条件：$Elt(r_1)\leq Elt(r_2)$ 或 $Elt(r_2)\leq Elt(r_1)$

一类操作：$r\equiv r_1\cap r_2\equiv\{t|t\in r_1\wedge t\in r_2\}$

二类操作：$r\equiv r_1\cap r_2\equiv\{t|(\exists t_1\in r_1)(\exists t\in r_2)(t=t_1)\}$

r 的基类为：

$$Elt(r)=\begin{cases}Elt(r_2) & Elt(r_2)\leq Elt(r_1)\\ Elt(r_1) & Elt(r_1)\leq Elt(r_2)\end{cases}$$

4)元组属性值交$\cap_t$

与 2)类同。只不过由并运算改为交运算。

5)集合减－

条件：$Elt(r_1)\leq Elt(r_2)$ 或 $Elt(r_2)\leq Elt(r_1)$

一类操作：$r\equiv r_1-r_2\equiv\{t|t\in r_1\wedge t\notin r_2\}$

二类操作：$r\equiv r_1-r_2\equiv\{t|t\in r_1\wedge\neg(\exists t_2\in r_2)(t=t_2)\}$

r 的基类为：$Elt(r_1)$。

6)元组属性值减$-_t$

设 $t_1\in r_1,t_2,\cdots,t_n\in r_2$，$At(t_1)=At(t_2)=\cdots=At(t_n)$，当 $\forall A\in\bar{S}(t_1)(t_1.A\cong t_2.A\cdots\cong t_n.A)$ 成立时，$t_1,t_2,\cdots,t_n$ 将变换为元组对象 t，$\forall A\in\bar{S}(t_1)(t.A\cong t_1.A)$ 且 $\forall A\in\bar{S}(t_1)(t.A\equiv t_1.A-t_2.A-\cdots-t_n.A)$。函数 $\bar{S}(t)$ 和 $S(t)$ 仍沿用 2)中的定义。

$r_2$ 上的等价关系 $R=\{(t_1,t_2)|At(t_1)=At(t_2)\wedge(\forall A\in\bar{S}(t_1))(t_1.A\cong t_2.A)\}$，$r_2$ 关于 R 的商集 $r_2/R$ 记为 Sm。

条件：$Elt(r_1)\leq Elt(r_2)$ 或 $Elt(r_2)\leq Elt(r_1)$

操作：$r\equiv r_1-_t r_2\equiv\{t|t\in r_1\wedge\neg(\exists r'_2\in Sm)(\exists t_2\in r'_2)(At(t_2)=At(t)\wedge(\forall A\in\bar{S}(t))(t.A\cong t_2.A))\vee t\in O_t\wedge(\exists t_1\in r_1)(\exists r'_2\in Sm)(\exists t_2\in r'_2)((At(t_2)=At(t_1)=At(t))\wedge(\forall A\in\bar{S}(t_1))$

$(t_1.A \supseteq t_2.A \supseteq t.A) \wedge (\forall Z \in S(t_1))(t.Z \equiv t_1.Z - \bigcup_{\forall r_2 \in r_2} t'_2.Z))\}$

r 的基类为 $Elt(r_1)$，操作产生集合对象 $r'_1$，$r'_1 \in r_1 | t \in r \wedge t \notin r_1 \wedge (\exists Z \in S(t))(t.Z \equiv r_1)\}$。

$r'_1$ 的类由其被嵌入到的属性 Z 的定义确定。

7)元组并 $Pack_r$

在集合对象 $r_1$ 中，若元组对象 $t_1, t_2 \in r_1$，且 $At(t_1) \subseteq At(t_2)$，当 $\forall Z \in At(t_1)$，有 $t_1.Z \supseteq t_2.Z$，则 $t_1$ 将从 $r_1$ 中消去。

操作：$r \equiv Pack_r(r_1) \equiv \{t | t \in r_1 \wedge \neg(\exists t_1 \in r_1)(At(t) \subseteq Alt(t_1) \wedge (\forall Z \in At(t))(t.Z \supseteq t_1.Z))\}$

r 的基类为 $Elt(r_1)$。

8)元组属性降变换 $\downarrow_{X \to A}$

本操作将集合对象 $r_1$ 中的元组对象 t 变换为 t'。X 是 t 的属性集的子集。非严格地说，就是将 t 的 X 中的属性及其值构成一个元组对象，并嵌入 t' 的属性 A 中。除属性 A 外，$t' = t$。显然，t 的类（或其超类）与 t' 的类之间存在超类与子类的继承关系。若要从 t' 中消除 X 中的属性，可借助于投影操作。记 $Att(\bar{\psi}(Elt(r_1)))$ 为 S。

条件：$X \subset S$ 且 $A \notin S$；

操作 $r \equiv \downarrow_{X \to A}(r_1) \equiv \{t | (\exists t_1 \in r_1)(t \in O_t \wedge At(t) = S \cup \{A\} \wedge (\forall Z \in S)(t.Z \equiv t_1.Z) \wedge t.A \equiv t_1[X])\}$

操作附带完成的工作：①对于由属性集 X 构成的元组类型所对应的元组类 $C_x$，若存在 $C' \in L$，有 $C' \leq Tuple$ 且 $X = Att(\bar{\psi}(C'))$，则 $C' = C_x$。否则，创建元组类 $C_x$。记 $C_x$ 的超类集 $\{g | g \in L \wedge g \leq Tuple \wedge Att(\bar{\psi}(g)) \subseteq X\}$ 为 G。$Struct(\psi(C_x))$ 是由属性集 $X - \bigcup_{\forall g \in G} Att(\bar{\psi}(g))$ 中的属性构成的元组结构。$Method(\psi(C_x)) = \varnothing$。对于 $g \in G$，有 $C_x \leq g$。②定义属性 A，其定义为 $A : C_x$。③r 的基类以如下方式确定：设 $Elt(r) = Cr$，若有 $C \in L$，$C \leq Tuple$，$Att(\bar{\psi}(C)) = S \cup \{A\}$，则 $Cr = C$；否则，Cr 为操作产生的新类。$Struct(\psi(Cr)) = [A : C_x]$。$Method(\psi(Cr)) = \varnothing$。$Cr \leq Elt(r_1)$。

一类操作与二类操作相同。

9)元组属性升变换 $\uparrow_{A(C)}$

该操作将 $r_1$ 中的元组对象 t 之属性 A 所嵌入的元组对象 t' 以类型 $\bar{\psi}(C)$ 的形式与 t 进行连接。记 $Att(\bar{\psi}(Elt(r_1)))$ 为 S；记 $Att(\bar{\psi}(C))$ 为 Y。

条件：①设属性 A 的定义形式为 A：T，则 $C \leq Tuple$ 或 C 类就是 T 类；②$S \cap Y = \varnothing$，且 $A \in S$。

操作：$r \equiv \uparrow_{A(C)}(r_1) \equiv \{t | t \in O_t \wedge At(t) = (S \cup Y) \wedge (\exists t_1 \in r_1)((\forall Z \in S)(t_1.Z \equiv t.Z) \wedge Y \subseteq At(t_1).A)(\forall Z' \in Y)(t_1.A.Z' \equiv t.Z'))\}$

r 的基类以如下方式确定：记 $Elt(r)$ 为 Cr，则 $Struct(\psi(Cr)) = \varnothing$，$Method(\psi(Cr)) = \varnothing$；在类的继承关系上有 $Cr \leq Elt(r_1)$ 和 $Cr \leq C$。

一类操作与二类操作相同。

10)投影 $\Pi_x$

该操作与关系模型中的类同。

条件：$X \subseteq Att(\bar{\psi}(Elt(r_1)))$

操作：$r \equiv \Pi_x(r_1) \equiv \{t | t \in O_t \wedge At(t) = X \wedge (\exists t_1 \in r_1)(\forall Z \in X)(t_1.Z \equiv t.Z)\}$

r 的基类以如下方式确定：若存在 $C_x \in L$，有 $C_x \leq Tuple$ 且 $X = Att(\bar{\psi}(C_x))$，则 $Elt(r) = C_x$。否则，生成新的元组类 $C_x$ 作为 r 的基类。$C_x$ 的超类集为 $\{g | g \in L \wedge g \leq Tuple \wedge Att(\bar{\psi}(g)) \subseteq X\}$。$Struct(\psi(C_x))$ 是由属性集 $X - \bigcup_{\forall g \in G} Att(\bar{\psi}(g))$ 中的属性构成的元组结构。$Method(\psi(C_x)) = \varnothing$。

一类操作与二类操作相同。

11)笛卡尔积 $\times_A$

为了使两个具有相同属性名的元组对象在相互连接时不产生属性名冲突，本操作将来自 $r_2$ 中的元组对象嵌入属性 A 中。通过属性的嵌套达到换名的目的。记 $Att(\bar{\psi}(Elt(r_1)))$ 为 S。

条件：$A \notin S$；

操作：$r \equiv r_1 \times_A r_2 \equiv \{t | t \in O_t \wedge At(t) = S \cup \{A\} \wedge (\exists t_1 \in r_1)(\exists t_2 \in r_2)((\forall Z \in S)(t_1.Z \equiv t.Z) \wedge t.A \equiv t_2)\}$

操作附带完成的工作：①定义属性 A，定义为 $A : Elt(r_2)$；②确定 r 的基类 Cr。若有 $C' \in L$ 且 $C' \leq Tuple$，$Att(\bar{\psi}(C')) = S \cup \{A\}$，则 Cr 为 C'；否则，生成基类 Cr，有 $Method(\psi(Cr)) = \varnothing$，$Struct(\psi(Cr)) = [A : Elt(r_2)]$，$Cr \leq Elt(r_1)$。

一类操作与二类操作相同。

12)自然联接 $\theta$

该操作与关系模型中的自然联接操作相似。记 $Att(\bar{\psi}(Elt(r_1)))$ 为 S1，$Att(\bar{\psi}(Elt(r_2)))$ 为 S2。

条件：$S1 \cap S2 \neq \varnothing$ 且对 $\forall A \in S1 \cap S2$，有 A 在 $\bar{\psi}(Elt(r_1))$ 中的定义与 A 在 $\bar{\psi}(Elt(r_2))$ 中的定义相同。

操作：$r \equiv r_1 \theta r_2 \equiv \{t | t \in O_t \wedge At(t) = S1 \cup S2 \wedge (\exists t_1 \in r_1)(\exists t_2 \in r_2)((\forall Z1 \in S1\text{-}S2)(t_1.Z1 \equiv t.Z1) \wedge (\forall Z2 \in S2\text{-}S1)(t_2.Z2 \equiv t.Z2) \wedge (\forall Z3 \in S1 \cap S2)(t_1.Z3 \supseteq t_2.Z3 \supseteq t.Z3))\}$

r 的基类 Cr 为：若有 $C' \in L$ 且 $C' \leq Tuple$，$Att(\bar{\psi}(C')) = S1 \cup S2$，则 $Cr = C'$；否则，生成 Cr，使

Method($\psi$(Cr))=$\varnothing$;Struct($\psi$(Cr))=$\varnothing$。在继承关系上有 Cr≤Elt($r_1$),Cr≤Elt($r_2$)。

13)打开嵌套 $Unnest^{B}_{A}$

本操作将 $r_1$ 中元组对象 t 的属性 A 所嵌入的集合对象中的元组对象 t' 与 t 相联结。考虑到 t' 的属性与 t 的属性可能相重,t' 将嵌入到属性 B 中。操作产生的集合对象 r 中的元组对象仍有属性 A。其作用类似嵌套关系模型中 Index 代数操作的效果。设属性 A 的定义形式为 A:T,T≤Set。记 Att($\bar{\psi}$(Elt($r_1$)))为 S。

条件:B∉S;

操作:$r\equiv Unnest^{B}_{A}(r_1)\equiv\{t|t\in O_t\wedge At(t)=S\cup\{B\}\wedge(\exists\ t_1\in r_1)((\forall\ Z\in S)(t.Z\equiv t_1.Z)\wedge(\exists\ t_2\in t_1.A)(t.B\equiv t_2))\}$

操作附带完成的工作:①定义属性 B,属性 B 的定义为:Base($\bar{\psi}$(T));②确定对象 r 的基类 Cr 为:若存在 C'∈L,C'≤Tuple,有 Att($\bar{\psi}$(C'))=S∪{B},则 Cr=C';否则,生成 Cr,Method($\psi$(Cr))=$\varnothing$,Struct($\psi$(Cr))=[B:Base($\bar{\psi}$(T))]。Cr≤Elt($r_1$)。

一类操作与二类操作相同

14)嵌套 $Nest_{X\to A}$

该操作与嵌套关系模型中的 Nest 操作类似。所不同的是,该操作产生的元组对象 t 中仍含有属性集 X 中的属性。记 Att($\bar{\psi}$(Elt($r_1$)))为 S。

条件:X⊂S 且 A∉S;

操作:$r\equiv Nest_{X\to A}(r_1)\equiv\{t|t\in O_t\wedge At(t)=S\cup\{A\}\wedge(\exists\ t_1\in r_1)((\forall\ Z1\in S)(t_1.Z1\equiv t.Z1)\wedge t.A\equiv\{q|q\in O_t\wedge At(q)=X\wedge(\exists\ t_2\in r_1)((\forall\ Z2\in(S-X))(t_2.Z2\subseteq t_1.Z2)\wedge(\forall\ Z3\in X)(q.Z3\equiv t_2.Z3))\}$

操作附带完成的工作:①对于由属性集 X 构成的元组类型所对应的元组类 $C_x$,若存在 C'∈L,有 C'≤Tuple 且 X=Att($\bar{\psi}$(C')),则 C'=$C_x$;否则,创建元组类 $C_x$。$C_x$ 的超类集{g|g∈L∧g≤Tuple∧Att($\bar{\psi}$(g))⊆X}。Struct($\psi$($C_x$))是由属性集 $X-\bigcup_{g\in G}Att(\bar{\psi}(g))$ 中的属性构成的元组结构。Method($\psi$($C_x$))=$\varnothing$。②定义属性 A,属性 A 的定义为:C'$_x$。这里,C'$_x$≤Set,Base($\bar{\psi}$(C'$_x$))=$C_x$。③确定 r 的基类 Cr 若有 C∈L,C≤Tuple,Att($\bar{\psi}$(C))=S∪{A},则 Cr=C;否则,生成新类 Cr,使 Method($\psi$(Cr))=$\varnothing$,Struct($\psi$(Cr))=[A:C'$_x$],Cr≤Elt($r_1$)。④附带产生集合对象 r',r'∈{$r_1$|t∈r∧t.A≡$r_1$},属性 A 所定义的类就是 r' 的类。

15)泛化 $\uparrow_C$

该操作将集合对象的基类变为 C 类。C 是该基类的超类,该操作的意义在于:通过调整集合对象的基类为我们所关心的 C 类,使得以后对该集合对象的代数操作都是在 C 类的元组结构上进行。这点在前面的代数操作定义中已有所反映。

条件:Elt($r_1$)≤C

操作:$r\equiv\uparrow_C(r_1)\equiv\{t|t\in r_1\}$

r 的基类为 C。

一类操作与二类操作相同。

16)特化 $\downarrow_C$

该操作保留了 $r_1$ 中 C 类以及 C 类的子类的对象,同时确立 r 的基类为 C。

条件:C≤Elt($r_1$)

操作:$r\equiv\downarrow_C(r_1)\equiv\{t|t\in r_1\wedge Att(\bar{\psi}(C'))\subseteq At(t)\}$

r 的基类为 C。

一类操作与二类操作相同。

17)选择 $\sigma_{CON}$

该操作与关系模型中的选择操作相同,CON 为选择的条件表达式。

操作:$r\equiv\sigma_{CON}(r_1)\equiv\{t|t\in r_1\wedge CON(t)\}$

r 的基类为:Elt($r_1$)

一类操作与二类操作是否相同,取决于 CON(t)。

结束语 通过放宽关系模型中关系集合所包含的元组的属性个数必须一致的限定以及在嵌套关系定义的层次上集合与元组必须严格交替出现的限制,我们将 OO 技术中的继承与封装机制引入其中,定义了 OODB 模型。针对该模型定义了 17 种代数操作。操作的语义以谓词演算的形式给出。由于操作是从集合对象到集合对象的,因此操作具有封闭性。由于查询操作不会产生无限集合对象以及无穷验证,因此操作是安全的。

在嵌套关系模型中,Unnest 操作要求嵌套的关系模式中不得有相同的属性名。否此,嵌套关系模型的查询代数只能支持对复合为层次结构的对象的查询;而 OODB 的代数操作通过将具有同名属性的对象嵌入到某一属性中,实现了具有同名属性的元组对象之间能相互连接而不会产生同名属性冲突的目的,因而 OODB 的代数操作能有效地支持对复合为有向有环图结构的对象的查询。

另外,代数操作尽可能地保留被操作对象的概念及其聚合关系。通过指明所关注类所具有的类型结构,实现了在异质集合对象上进行同质操作为目的。同时,也使得操作产生的集合对象的基类与被操

(下转第 66 页)

计算机科学 1997 Vol. 24 №. 2

# 基于稳健误差估计器的快速 BP 算法

刘光远　邱玉辉　虞厥邦

（西南师范大学计科系　重庆 630715）（成都电子科技大学光电系）

**摘　要**　标准的或许多改进的 BP 算法大都采用均方误差估计器，致使学习易于陷入局部极小、收敛速度慢和对初始权敏感等。本文针对这些缺点提出了较好的算法，仿真结果证实了该算法的正确性。

**关键词**　标准 BP 算法，误差估计器，稳健 BP 算法，XOR 问题

## 1　引言

自从 Rumelhart 等科学家于 1986 年提出并系统论述了反向传播（Backpropagation，简称 BP）算法以来，该算法已在模式识别、信号处理、自适应控制、机器人、机器学习等众多领域得到了广泛的应用。近十年来许多研究人员对该网络进行了深入的研究，其研究重点集中在诸如训练时间长、易于陷入局部极小、对初始权敏感、训练存在发散或麻痹现象等 BP 网络的不足点上，这些工作都是卓有成效的。

众所周知，标准 BP 算法所定义的误差估计器是一个二次式函数，但这并不是唯一的选择。1988 年 Baum 和 Wileze 等人提出了一个误差估计器，可以克服麻痹现象，但可能导致过调或振荡。1989 年，Falman 提出了一种折衷方案，其效果比前者好。然而，这些工作并未很好地解决 BP 网对初始权敏感及收敛速度慢等问题。针对这些方法的局限性，本文利用拉格朗日乘数法提出了一种稳健 BP 算法。

## 2　稳健 BP 算法

事实上，对于 BP 算法，所选用的误差（函数）估计器即使不是二次式，只要该误差估计器在目标函数等于实际输出时能达到最小值就满足要求。最优化理论中的拉格朗日乘数法给予我们启示，即只要找到一项适当的约束，求解某误差函数之极小使之能满足要求，则我们的问题即可得到解决。

（上接第 39 页）

作集合对象的基类之间存在继承关系，为操作产生的对象的基类指明了适当的位置。

### 参考文献

[1] Won Kim，面向对象数据库系统：诺言·现实·前景(1)，计算机科学，21(4)1994

[2] Won Kim，面向对象数据库系统：诺言·现实·前景(2)，计算机科学，21(5)1994

[3] M. Mosaffari and Y. Tanaka, ODM: An object-oriented data model, New Generation Computing, 7(1) 1989

[4] Jan Van Den Bussche and Andreas Heuer, Using SQL with object-oriented database, Info. Sys., 18(7)1993

[5] M. A. Roth, H. F. Korth and A. Siberschatz, Extended algebra and calculus for nested relational databases, ACM Trans. Database Systems, 13(4) 1988

[6] D. Van Gucht and P. C. Fischer, Multilevel nested relational structures, J. Comput. Sys. Sci, 36(1) 1988

[7] G. M. Shaw and S. B. Zdonik, An object-oriented query algebra, Proc. 2nd Int. Workshop Database Programming Language, 1989

[8] G. M. Shaw and S. B. Zdonik, A query algebra for object-oriented database, Proc. 6th Int. Conf. Data Engineering, 1990

刘光远　博士生，主要研究领域为神经网络。