

NF² 关系模型与属性继承*

李天柱

(河北大学计算中心, 保定 071002)

摘要 将面向对象模型与面向值的模型相结合, 可兼得二者的优点. 本文基于具有组合元组标识的 NF² 关系模型给出了强调嵌套结构特征的关系运算投影 Π^S 和自然连接 \sim^S , 并在此基础上研究 NF² 关系数据模型中的类、子类及其间的属性继承的语义及形式化定义; 引入了 S-子类的概念; 研究了公共子类及多属性继承的特征. 所有的讨论基于组合元组标识, 具有较强的面向对象模型的特征.

关键词 NF² 关系模型, 关系运算, 类、子类, 属性继承.

类、子类及其间的继承特性是面向对象数据库(O-O 数据库)的重要特征, 但缺乏严格的确切的描述与定义. 有的文章也讨论了 NF² 关系模型上的继承特性^[1], 但语义不清晰, 不足以反映 O-O 模型中所描述的类、子类及其间的继承特性.

从使得面向对象模型与面向值的模型相结合出发, 本文研究了 NF² 关系模型中类、子类及其间属性继承的语义和形式化定义. 有的概念略不同于通常 O-O 模型中的相应概念, 但却更清晰, 具有更确切的语义背景.

1 关于 NF² 关系模式

我们采用基于文献[2]的关系模式的描述方法.

定义 1. 设有规则 $R, R=(R_1, \dots, R_n)$, 其中 R, R_1, \dots, R_n 叫做名字, $R_i (i=1, \dots, n)$ 叫做 R 的属性, 基中一些 $R_i (i \leq i \leq n)$ 仍可为另一规则的左部, 但不出现名字的递归. 则这样由规则 R 的有限递归地得出的一组规则, 叫做 NF² 深关系模式 R , 简称为 NF² 关系模式 R , 记做 $R=(R_1, \dots, R_n)$, 简记做关系模式 R .

在上述规则组中, 不出现在任何规则左部的属性叫做零阶属性; 否则, 叫做高阶属性. 高阶属性具有双重性, 它既是所在规则的一个属性(当它在一规则的右端时), 又是以其自身为根的规则集(以它的左端的规则开始的一组规则)所构成的 NF² 关系模式的名字.

令定义 1 中的每一规则含有二个特殊属性(零阶属性), 一个为元组标识 id , 另一个为对象标识 R° . id 和 R° 构成组合元组标识. 含有 id 和 R° 的元组叫做有标识元组, 不含 id 和

* 本文 1992-07-21 收到, 1993-06-21 定稿

本课题由国家自然科学基金和河北省自然科学基金资助. 作者李天柱, 1941 年生, 教授, 主要研究领域为数据库, 知识库及信息系统.

本文通讯联系人: 李天柱, 保定 071002, 河北大学计算中心

R^0 的元组叫做无标识无组或值元组. 又令每一关系有一个关系标识 ID . id 、 ID 和 R^0 均由系统管理, 服从“唯一标识假设”, R^0 为该元组所描述的实体(entity)在计算机内的逻辑标识, 类似于文献[3]中的代号(surrogate), 是一个特殊的 Key, 在任一 NF² 关系中无重值(高阶属性值也是一个 NF² 关系), 但在不同的 NF² 关系之间, R^0 可能有重值*.

由于组合元组标识的引入, 元组的相等可分为同一(identical)、浅相等(shallow-equal)、深相等(deep-equal)^[4], 我们分别用 $=$ 、 $\overset{d}{=}$ 、 $\overset{a}{=}$ 来表示.

本文中, 以 E_R 或 $R_1R_2\cdots R_n$ 表示 NF² 关系模式 $R=(R_1, \dots, R_n)$ 的属性集(不含 id 和 R^0). 若 $Q=R_{i_1}R_{i_2}\cdots R_{i_p}$ 为 E_R 的一个子集, 令 $t[Q]$ 表示元组 t 在属性集 Q 上的分量. $t[Q]$ 本身没有标识, 但 $t[Q]$ 所含元素可能是一高阶属性值, 此高阶属性值含有标识. 对不同的 $t[Q]$, 可应用浅相等和深相等的概念.

对 NF² 关系, 也引入上述三种相等. 两个 NF² 关系同一意为所有有标识元组全相同. 其它相等可依元组相等类型依次类推**.

本文对零阶属性值的相等一律用 $=$ 表示.

2 投影 Π^S 及自然连接 \bowtie^S

为了讨论类、子类及继承, 与文献[2]不同, 我们给出考虑保持嵌套结构的 NF² 关系运算投影 Π^S 及自然连接 \bowtie^S .

定义 2. 设 r 为 NF² 关系模式 $R=(R_1, \dots, R_n)$ 下的一个 NF² 关系, Q 为 R 的属性集 E_R 的一个子集, 则 r 在 Q 上的投影为关系模式 P 下的 NF² 关系:

$$\textcircled{1} P=(R_{i_1}, \dots, R_{i_p}), R_{i_k} \in \{R_1, \dots, R_n\}, i_1 \leq i_k \leq i_p, Q=R_{i_1} \cdots R_{i_p};$$

$$\textcircled{2} \Pi_Q^S(r) ::= \{t_i \mid \exists t \in r, t_i[Q]=t[Q] \wedge t_i[R^0]=t[R^0] \wedge t_i[id]=new\}.$$

这样的投影叫做 r 在 Q 上的浅投影, 简称投影; 若将②中的 $t_i[Q]=t[Q]$ 改为 $t_i[Q] \overset{d}{=} t[Q]$, 则这样的投影叫做 r 在 Q 上的深投影; 若将②中的 $t_i[id]=new$ 改为 $t_i[id]=t[id]$, 则这样的投影叫做同一投影. 根据标识唯一性假设, 同一投影不能作永久存储. 事实上, 同一投影是一个视图(View), 即一个虚关系. 其中 $t_i[id]=new$ 表示 t_i 的元组标识为新的.

上面定义的 r 在 Q 上的投影保持了 NF² 关系 r 原来的嵌套结构特征, 运算结果不会产生重复的有标识元组(尽管可能存在重复的无标识元组).

定义 3. 设关系模式 R_1 和 R_2 的属性集(不含 id 和 R^0)分别为 E_1 和 E_2 , 其公共属性集为 X , 且令 $A=E_1-X, B=E_2-X$, 则 R_1 下的 NF² 关系 r_1 和 R_2 下的 NF² 关系 r_2 之间的自然连接 $r_1 \overset{S}{\bowtie} r_2$ 对应的关系模式和关系为:

$$\textcircled{1} R=(A, X, B),$$

$$\textcircled{2} r_1 \overset{S}{\bowtie} r_2 ::= \{t \mid \exists u \in r_1, \exists v \in r_2; t[X]=u[X]=v[X] \wedge t[A]=u[A] \wedge t[B]=v[B] \wedge (t[R^0]=u[R^0]=v[R^0] \vee u[R^0] \neq v[R^0] \wedge t[R^0]=new) \wedge t[id]=new\}.^{***}$$

* 我们将另文专门讨论对象标识(包括这儿的组合元组标识)的语义及构成.

** NF² 关系与 O-O 模型中的集合不完全相同, 其相等概念也略有不同.

*** $u[R^0]=v[R^0]$ 时, R^0 应包含在 X 中.

本文所感兴趣的是 $u[R^{\circ}] = v[R^{\circ}]$ 的情况, $u[R^{\circ}] \neq v[R^{\circ}]$ 的情况较复杂.

象投影运算一样,修正②中一些条件,可将自然连接分为浅自然连接(简称自然连接)和深自然连接(但我们不定义同一自然连接).

文献[2]中定义自然连接考虑 NF^2 关系自然连接与对应平面化关系的自然连接的等价性,基于公共属性集上的属性值的扩展交不空(文献[5]亦然);而本文给出的自然连接^{①②}着眼于保持 NF^2 关系的嵌套结构,基于公共属性集上的属性值相等(浅相等或深相等),保持了高阶属性值的完整性.

3 类、子类及继承特性

从 O—O 数据模型观点看,类(class)是同类对象的集合,它们有共同的属性和运算.

子类及其超类(subclass and superclass)是以不同抽象层次描述事物间的“是一个”(is a)联系的.“是一个”关系是一个二元关系,具有自反性和传递性.因此,子类和其超类间的关系也应是一个二元关系,且具有自反性和传递性.类比子类的抽象层次更高一些.从类的抽象层次看,子类中的对象也应是其超类中的对象,子类中的对象具有其超类中的对象所具有的属性和运算,也可具有它自己所独有的属性和运算.因此,从数据结构看,超类及其子类是基于类所具有的属性和类中的对象集这两方面来划分的*.如学生是人的子类,研究生是学生的子类.一个研究生,如张三,既是研究生类中的一个成员对象,也是学生类中的一个成员对象,进而也是人这个类中的一个成员对象.而作为一个研究生,他不仅应具有学生和人所应具有的属性,而且还可有其特有的属性,如导师,研究方向等.这就是通常所说的描述“是一个”联系的类和子类的含义.

引入继承的概念是为了便于信息共享和减少信息冗余(数据信息及其上的操作信息).为此,可以令子类只表达它含有的对象所特有的属性和运算,而从其超类得到这些对象所应具有的,与其超类中的对象相同的那些属性和运算.这应该就是具有继承特性的类和子类的概念的含义.我们称这样的子类为简化的子类,简称 S -子类.因此,在继承概念下对超类和子类(指 S -子类)与原本对超类及子类的描述是有区别的,前者是后者的变形.这里对类及子类的描述与 SDM 语义模型中对类及子类的描述是一致的^[6],不过更清晰**.

基于以上对类、子类及继承的概念的描述,下面给出在 NF^2 关系模型下的类、子类及继承的形式化描述及其性质.

定义 4. 设 $R = (R_1, \dots, R_n)$ 为 NF^2 关系模式, r 为 R 下的任一 NF^2 关系, α 为 r 上的运算的集合,则三元组 $\langle R, r, \alpha \rangle$ 叫做类,简记为类 R .

本文以大写字母表示类名,以相应的小写字母表示该类中的 NF^2 关系.

在数据库理论和技术中,主要考虑大量数据的管理问题,因此数据及其结构就显得特别重要.文献[7]指出,在 O—O 模型中,子类对其超类的继承有 4 种.实质上为 2 种:1 种是基于类中的属性,1 种是基于类中对象上的运算.本文主要基于类中的对象集及其所具有的属性来讨论类、子类及其间的继承特性,对其上的运算所扮演的角色不做讨论,在类的表示中,

* 有些研究者将操作(方法)看做特殊属性.
** 通常人们所说的子类,有时指这儿的子类,有时指这儿的 S -子类.

今后也将运算集符号 α 省去.

在下面的讨论中,对投影和自然连接运算,若无特殊说明,是指浅投影和浅自然连接.

定义 5. 设有类 $\langle R, r \rangle$ 和类 $\langle G, g \rangle$,

$$\textcircled{1} R = (R_1, \dots, R_n), G = (R_1, \dots, R_n, R_{n+1}, \dots, R_m), m \geq 0;$$

$$\textcircled{2} g = r \bowtie_{Q}^S g$$

其中 $Q = R^\circ E_R$, 则类 G 叫做类 R 的子类,类 R 叫做类 G 的超类,并记为 $R > G$. 若 $m = 0$, 则类 G 叫做类 R 的平凡子类. 由于 R° 是实体标识,且具有唯一性,故只要取 $Q = R^\circ$ 即可.

这个定义完整地描述了子类与其超类之间的“是一个”的联系,

这儿定义的子类和超类是同层次上的概念,且保持了 NF² 关系嵌套结构及内容的完整性. 这不同于文献[5]中子对象的概念,也不完全同于次型的概念.

我们不加证明而指出子类的若干性质:

(1) 超类——子类关系是自反的,传递的.

(2) $E_R \subseteq E_G$, 这叫做型包含.

(3) 令 $S(R^\circ \langle r \rangle)$ 和 $S(R^\circ \langle g \rangle)$ 分别表示关系 r 和 g 中的 R° 取值集合,则 $S(R^\circ \langle r \rangle) \supseteq S(R^\circ \langle g \rangle)$, 这叫做对象集包含.

上述(2)和(3)就是通常所述子类——超类之间的“是一个”的语义.

定义 6. 设类 $\langle G, g \rangle$ 是类 $\langle R, r \rangle$ 的子类,且 $R = (R_1, \dots, R_n), G = (R_1, \dots, R_n, R_{n+1}, \dots, R_m), E_m = R_{n+1} \cdots R_m$. 若类 $\langle G', g' \rangle$ 满足

$$\textcircled{1} G' = (R_{n+1}, \dots, R_m);$$

$$\textcircled{2} g' = \prod_{E_m}^S (g)$$

则类 $\langle G', g' \rangle$ 叫做类 R 的关于子类 G 的简化的子类,简称为 S -子类,并以 $R \gg G'$ 记之. 平凡子类的对应 S -子类只含 id 和 R° .

本文中以带“ $'$ ”的字母表示相应子类的 S -子类及其中的 NF² 关系.

显然, S -子类有如下性质:

$$\textcircled{1} E_R \cap E_{G'} = \emptyset;$$

$$\textcircled{2} S(R^\circ \langle g' \rangle) = S(R^\circ \langle g \rangle), S(R^\circ \langle g' \rangle) \subseteq S(R^\circ \langle r \rangle).$$

定义 7. 设类 $\langle G, g \rangle$ 是类 $\langle R, r \rangle$ 的子类,且 $R = (R_1, \dots, R_n), G = (R_1, \dots, R_n, R_{n+1}, \dots, R_m)$. 类 $\langle G', g' \rangle$ 是类 R 的关于子类 G 的 S -子类 ($G' = (R_{n+1}, \dots, R_m)$). 则 S -子类 G' 对其超类 R 的属性继承是指类间的映射 $\rho: G' \rightarrow G_1$.

$$\textcircled{1} \text{ 模式级 } \rho: G' \rightarrow G_1 \mid E_{G_1} = E_R \cup E_{G'};$$

$$\textcircled{2} \text{ 关系级 } \rho: g' \rightarrow g_1 = r \bowtie_{R'}^S (g');$$

$$\textcircled{3} \text{ 元组级 } \rho: u \rightarrow v \mid (u \in g', \exists t \in r: u[R^\circ] = t[R^\circ]); v \in g_1 \wedge v[E_R] = t[E_R] \wedge v[E_{G'}] = u[E_{G'}] \wedge v[R^\circ] = u[R^\circ] \wedge v[id] = \text{new}.$$

上面映射规则(2)和(3)实质上是一致的.

属性继承有如下性质:

(1) $E_{G_1} = E_G, g_1 = g, S(R^\circ \langle g_1 \rangle) = S(R^\circ \langle g \rangle) = S(R^\circ \langle g' \rangle)$. 这就是说,在浅相等意义下,通过属性继承可由 S -子类恢复对应子类.

(2)属性继承特性包括 S -子类对其超类的型和对象值的继承. 这就是 S -子类对其超类属性继承的语义.

这 2 个性质描述了 S -子类与对应子类及超类之间的关系.

4 S -子类约束

子类定义和 S -子类定义规定了超类、子类、 S -子类之间的联系,即规定了 S -子类所受的约束.因本模型中,直接存储的是基类和 S -子类(对应子类由属性继承来生成),故直接描述 S -子类(相对于其超类)所应满足的约束是必要的,以避免异常更新.

设 $R \gg G'_1 \gg \dots \gg G'_n$ 是某子类序列对应的 S -子类序列,类 R 是一个基类.对任意的 $i, 0 \leq i \leq n$ (G'_0 代表 R), G'_i 应满足下列约束:

- ① $E_R \cap E_{G'_1} \cap \dots \cap E_{G'_n} = \emptyset$;
- ② $S(R^\circ \langle g'_i \rangle) \supseteq S(R^\circ \langle g'_{i+1} \rangle)$.

约束①是为了尽可能地减少信息冗余;约束②保证可由 S -子类通过属性继承生成对应的类 R 的子类.对类 R 或类 G'_i 更新时,这些约束应保持成立.不难由此给出具体的更新规则,由于篇幅所限,不再详述.

5 公共子类和多继承

有时,一个类同时是几个类的子类.如承担研究课题的研究生(简称 RA),既是在校研究生,又是承担研究任务的研究人员. RA 类同时是研究生类和研究人员类的子类. RA 类应同时具有研究生类和研究人员类所具有的属性,如,研究生类的所学“课程”、“导师”,研究人员的所承担“课题”等.在子类和 S -子类定义中应有标识能表达一个 RA 学生既是一个学生(研究生)又是一个研究人员,即研究生和研究人员的同一个人的情况.相应地,属性继承中就存在多继承.本文引入的特殊属性 R° 可充当这个标识.

首先,扩展子类的定义.

定义 8. 设有类 $\langle G_1, g_1 \rangle$ 和类 $\langle G_2, g_2 \rangle$,若类 $\langle G_3, g_3 \rangle$ 满足

- ① $E_{G_3} \supseteq (E_{G_1} \cup E_{G_2})$;
- ② $g_1 \cdot \overset{S}{E}_1 g_3 = g_2 \cdot \overset{S}{E}_2 g_3 = g_3$, 其中 $E_1 = R^\circ E_{G_1}, E_2 = R^\circ E_{G_2}$;

则类 $\langle G_3, g_3 \rangle$ 叫做类 $\langle G_1, g_1 \rangle, \langle G_2, g_2 \rangle$ 的公共子类.

显然,若类 G_3 是类 G_1, G_2 的公共子类,则类 G_3 是类 G_1 的子类,也是类 G_2 的子类.而且, $S(R^\circ \langle g_1 \rangle) \supseteq S(R^\circ \langle g_3 \rangle), S(R^\circ \langle g_2 \rangle) \supseteq S(R^\circ \langle g_3 \rangle)$, 即 $S(R^\circ \langle g_3 \rangle) \subseteq ((S(R^\circ \langle g_1 \rangle) \cap S(R^\circ \langle g_2 \rangle)))$.

定义 9. 设类 $\langle G_3, g_3 \rangle$ 是类 $\langle G_1, g_1 \rangle$ 和 $\langle G_2, g_2 \rangle$ 的公共子类,类 $\langle G'_3, g'_3 \rangle$ 满足

- ① $E_{G'_3} = E_{G_3} - (E_{G_1} \cup E_{G_2})$,
- ② $g'_3 = \prod_{Q \in G_3}^S g_3$, 其中 $Q = E_{G'_3}$

则类 G'_3 叫做类 G_1, G_2 的公共子类 G_3 的公共 S -子类.

显然, $S(R^\circ \langle g'_3 \rangle) = S(R^\circ \langle g_3 \rangle)$.

公共子类 G_3 同时是类 G_1 的子类和类 G_2 的子类,但其对应的 S -子类却不同于公共 S

—子类 G'_3 , 若 $E_G = \emptyset$, 则类 G'_3 中只含 R° 和 id , 且类 G'_3 仍不能省。

定义 10. 设 $\langle G'_s, g'_s \rangle$ 是类 $\langle G_1, g_1 \rangle, \dots, \langle G_n, g_n \rangle$ 的关于 $G_i (i=1, \dots, n)$ 的公共子类 $\langle G_s, g_s \rangle$ 的公共 S -子类, 则类 G'_s 对类 G_1, \dots, G_n 的多属性继承是指类间映射 $\rho: G'_s \rightarrow G_p$;

① 模式级 $\rho: G'_s \rightarrow G_p | E_{G_p} = E_1 \cup \dots \cup E_m \cup E_{G'_s}$, 其中 $E_i = E_{G_i}, i=1, \dots, m$;

② 关系级 $\rho: g'_s \rightarrow g_p = g_{n_1} \times_{R^\circ} \dots \times_{R^\circ} g_{n_m} \times_{R^\circ} g'_s$;

③ 元组级 $\rho: u \rightarrow v | ((u \in g'_s, \exists t_{n_1} \in g_{n_1}, \dots, \exists t_{n_m} \in g_{n_m}) : u[R^\circ] = t_{n_1}[R^\circ] = \dots = t_{n_m}[R^\circ]) : v \in g_p \wedge v[E_1] = t_{n_1}[E_1] \wedge \dots \wedge v[E_m] = t_{n_m}[E_m] \wedge v[E_{G'_s}] = u[E_{G'_s}] \wedge v[R^\circ] = u[R^\circ] \wedge v[id] = new$;

若 $\{n_1, \dots, n_m\} = \{1, \dots, n\}$, 则上述多属性继承叫做全多属性继承; 若 $n_m = 1$, 则上述多属性继承叫做择一多属性继承; 其它情况叫做部分多属性继承。这比一般 $O-O$ 模型中的多属性继承要灵活。

类似于普通属性继承, $g_p = g_s$ 成立。

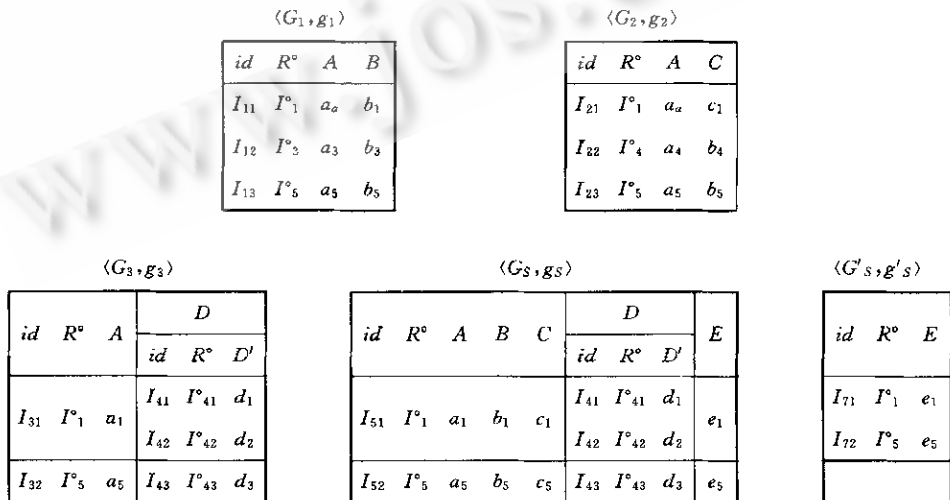
若公共 S -子类的若干超类之间有相同属性, (由 S -子类约束①, S -子类与其超类之间不会有相同属性名), 则在属性继承中会出现属性名冲突。象 $O-O$ 模型一样, 可采用重新命名的方法消除属性名重名冲突^[6]。但不能简单地应用“最左原则”或者“用户指定原则”于此, 因为不消除属性重名就意味着在一个依多属性继承由 S -子类恢复生成的子类中可能有重名属性, 且取值方式不同, 这会引起混乱。

公共 S -子类约束基本同第 4 节所述, 但条件更强些。对定义 10 中所用记号, 公共 S -子类 $\langle G'_s, g'_s \rangle$ 应满足以下约束:

① $E_{G'_s} \cap E_{G_i} = \emptyset, i=1, \dots, n$;

② $S(R^\circ \langle g_i \rangle) \supseteq S(R^\circ \langle g'_s \rangle), i=1, \dots, n$ 。

图 1 给出了一个公共子类, 公共 S -子类和多属性继承的例子。从这个例子我们还可以看出, $\langle G_s, g_s \rangle$ 是 $\langle G_i, g_i \rangle (i=1, 2, 3)$ 的子类, 但 $\langle G'_s, g'_s \rangle$ 不是 $\langle G_i, g_i \rangle$ 关于子类 $\langle G_s, g_s \rangle$ 的 S -子类。



<i>id</i>	R°	<i>A</i>	<i>B</i>	<i>C</i>	<i>E</i>
I_{91}	I°_1	a_1	b_1	c_1	e_1
I_{92}	I°_5	a_5	b_5	c_5	e_5

<i>id</i>	R°	<i>B</i>	<i>C</i>	<i>E</i>
I_{81}	I°_1	b_1	c_1	e_1
I_{82}	I°_5	b_5	c_5	e_5

图1 $\langle G_s, g_s \rangle$ 是 $\langle G_1, g_1 \rangle, \langle G_2, g_2 \rangle, \langle G_3, g_3 \rangle$ 的公共子类, $\langle G'_s, g'_s \rangle$ 是对应的公共S-子类, $\langle G'_s, g'_s \rangle$ 对 $\langle G_1, g_1 \rangle, \langle G_2, g_2 \rangle$ 的部分多属性继承生成 $\langle G_p, g_p \rangle$ (不继承 $\langle G_3, g_3 \rangle$); $\langle G'_{s3}, g'_{s3} \rangle$ 是对应于 $\langle G_s, g_s \rangle$ 的 $\langle G_3, g_3 \rangle$ 的S-子类.

6 后记

本文的讨论采用了在整个系统服从“唯一标识假设”的元组标识和关系标识. 组合元组标识中的 R° 起重要作用. 它近似于文献[3]中的代号(surrogate), 也近似于O-O模型中的对象标识. R° 是实体在计算机内的逻辑标识, 由系统管理, 便于系统使用, 但不便于用户使用. 为了便于用户指称实体, 还应给每一个实例一个用户标识, 即实体名或编号. 在特定应用范围, 实体名或编号应具有唯一性, 不重名, 并与 R° 构成1-1对应, 也可将用户标识与 R° 合并, 如按用户给定规则, 由系统生成 R° (这涉及到标识体系的语义及构成, 我们将另文专门讨论). 在一个用dBASE实现的人事管理系统中, 我们用实体编号(人员编号)来模拟 R° 的作用(没有系统管理的 R°), 从而模拟实现了本文所述的类、S-子类、属性继承及S-子类约束. 当然, NF^2 嵌套结构也是模拟的, 如图2-6所示.

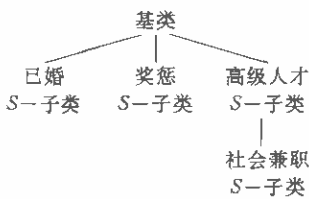


图2 人事管理类聚集结构

编号	姓名	职称	婚否
001	李四	无	未
002	王五	教授	婚
003	赵六	讲师	婚

图3 基类

编号	成员姓名	职称
002	钱玉	妻
	钱一	子
003	钱二	女
	刘玉	夫
	刘一	子

图4 已婚(家庭成员)S-子类

编号	姓名	职称	婚否
001	李四	无	未
002	王五	教授	婚

图5 基类中删除记录(编号=003)

编号	成员姓名	职称
002	钱玉	妻
	钱一	子
	钱二	女

图6 已婚S-子类中自动删除编号=003的记录但加不进编号=005的人的家庭成员

本文的讨论基本上都是基于浅相等的概念. 这和 starburst^[9]中的复杂对象视图(com-

plex object views)的思想是一致的,支持 NF^2 关系中被嵌套成分的共享.不难发现,本文的讨论在深相等的概念下仍成立.

本文基于具有组合元组标识的 NF^2 关系模型,讨论了类、子类、 S -子类、属性继承及相关概念,给出了它们的一种形式化定义,其语义清晰,能确切地反映现实世界,丰富了 $O-O$ 模型和 NF^2 关系模型理论,对面向值的模型和面向对象的模型相结合具有重要意义.

参考文献

- 1 Botcher S. Attribute inheritance implemented on top of a relational database. IEEE Int. conf. On Data Engineering, 1990.
- 2 Roth M A *et al.* Extended algebra and calculus for nested relational databases. ACM TODS, 1988, 13(4).
- 3 Codd E F. Extending the database relational model to capture more meaning. ACM TODS, 1979, 4(4).
- 4 Khoshafian S N, Copeland G P. Object identity. Proc. of Conf. on OOPSLA, 1986.
- 5 陈其明. 数据库对象和型的广义结构模型. 计算机学报, 1989, 12(8).
- 6 Hammer M, Mcload D. Database description with SDM: a semantic database model. ACM TODS, 1981, 6(3).
- 7 Atkinson M *et al.* The object-oriented database system manifesto. The 1st Int. Conf. on Deductive and Object-Oriented Database, 1989.
- 8 Banerjee J *et al.* Data model issues for object-oriented applications. ACM Transaction on Office Information Systems, 1987, 5(1).
- 9 Lohman G M *et al.* Extensions to starburst: objects, types, functions and rules. ACM Communication of the ACM, 1991, 34(10).

THE ATTRIBUTE INHERITANCE IN NF^2 RELATIONAL MODEL

Li Tianzhu

(Computer Center, Hebei University, Baoding 071002)

Abstract Combining object-oriented model with value-based model would help getting the merits of both models. In this paper, based on NF^2 relational model with composited tuple identity, the operations of projection \prod^S and natural join \bowtie^S , keeping the nested structure of relations, are given. From this, the authors have studied the semantics and formalized definition for class, subclass, attribute inheritance in NF^2 relational model, introducing a special attribute R^o and concept of S -subclass, discussed the common subclass and multiple inheritance. All of the discussions are based on composited tuple identity, possessing the features in $O-O$ model.

Key words NF^2 relational model, relational operation, class, subclass, inheritance.