

55-56

关于面向对象的类的形式化研究

Formalization Research about the Class in Oriented-Object Field

李晓燕 林云 蔡霞

(华中师范大学计算机科学系 武汉430079)

TP312

Abstract This paper introduces the notions of atom class and composite class. Furthermore, it gives a strict formalized description of class and its inheritance. Obviously, this kind of formalization underlies the construction of object-oriented model.

Keywords Atom class, Composite class, Inheritance, Formalization

在面向对象技术中,面向对象程序设计语言可以完全抛弃传统程序设计语言中的类型概念,即使它的最基本数据也可用类来描述。类及其继承性是面向对象程序设计语言区别传统程序设计语言的本质特性,本文将讨论类及其继承性的形式化描述及其数学属性。

1. 原子类与复合类

定义1 原子类,指不能再分解的类。每一个原子类都包含一个值集和一个隐含的行为集,值集是具有相同性质的原子数据之集合,而原子数据则是由字母、数字和其它特殊符号构成的字符串,其中由数字、小数点和正负号构成的字符串称之为“数”,原子数据 true 和 false 称之为“逻辑值”,原子类包括:类 INT、类 REAL、类 CHAR 和类 BOOL。

原子类 INT、类 REAL 的值集分别是整数集和实数集,它们隐含的行为集是分别施加于整数集和实数集之上的四则运算。类 CHAR 的值集是由基本符号构成的符号表,隐含对字符的连接等操作。类 BOOL 的值集包括 true 和 false 两个值,隐含布尔运算。

定义2 复合类是一个二元组,记为:

$$C = (D, F)$$

其中: D 是数据集:

$$D = [t_1 d_1; t_2 d_2; \dots; t_n d_n] (n > 0)$$

这里 $t_i (i = 1, 2, \dots, n)$ 是原子类或复合类; $d_i (i = 1, 2, \dots, n)$ 是类的载体; 允许 $t_i = t_j (i \neq j)$, 但 $d_i \neq d_j (i \neq j)$ 。

F 是行为集:

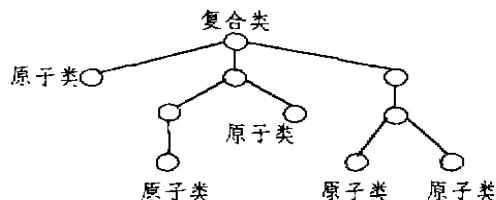
$$F = \{f | f: t_1 \times t_2 \times \dots \times t_n \rightarrow C\}$$

例如: $\text{Class employee} = (D, F)$

其中: $D = [\text{INT num}; \text{CHAR sex}; \text{INT age}; \text{REAL salary}]$

$$F = \{\text{print}(\text{employee}) \rightarrow \text{employee}\}$$

由复合类定义可知,一个复合类的结构是一棵以原子类作为叶子结点的树,如下图所示:



定义3 类 C 的值域记为 $vfield(C)$, 递归定义为:

① 原子类的值域: 类 INT 的值域是整数集, 类 REAL 的值域是实数集, 类 CHAR 的值域是基本字符表, 类 BOOL 的值域 $vfield(BOOL) = \{true, false\}$;

② 复合类的值域: 设复合类 $C = (D, F)$, 其中:

$$D = [t_1 d_1; t_2 d_2; \dots; t_n d_n] (n > 0)$$

$$F = \{f | f: t_1 \times t_2 \times \dots \times t_n \rightarrow C\}$$

$$vfield(C) = \{[v_1 d_1; v_2 d_2; \dots; v_n d_n] | v_i \in vfield(t_i), i = 1, 2, \dots, n\}$$

定义4 类 C 的实例记为 $instance(C)$, 递归定义为:

① 对于原子类 C, $\forall x [class(x) = C \Rightarrow x \in instance(C)]$ ($class(x)$ 表示 x 的类);

② 对于复合类 $C = (D, F)$, 其中:

$$D = [t_1 d_1; t_2 d_2; \dots; t_n d_n] (n > 0)$$

$$F = \{f | f: t_1 \times t_2 \times \dots \times t_n \rightarrow C\}$$

设 $C' \in instance(C)$, $C' = (D', F')$

$$\text{则 } D' = [x_1 d_1; x_2 d_2; \dots; x_n d_n] (n > 0)$$

$$F' = \{f | f: x_1 \times x_2 \times \dots \times x_n \rightarrow C'\}$$

其中, $x_i \in instance(t_i) (i = 1, 2, \dots, n)$

2. 类的偏序关系

定义5 设类 $C_1 = (D_1, F_1)$, $C_2 = (D_2, F_2)$ 。若 $\forall d [d \in D_1 \Rightarrow d \in D_2] \wedge \forall f [f \in F_1 \wedge f: \rightarrow C_1 \Rightarrow f \in F_2 \wedge f: \rightarrow C_2]$, 且不对 $f \in F_2$ 进行重载, 则称类 C_2 是由类 C_1 派生出的子类, 而称类 C_1 是类 C_2 的超类, 记为:

$$C_2 \leq C_1$$

例如, $\text{Class employee} = (D_1, F_1)$

$\text{Class manager} = (D_2, F_2)$

其中: $D_1 = [\text{INT num}; \text{CHAR sex}; \text{INT age}; \text{REAL salary}]$

$F_1 = \{\text{print}(\text{employee}) \rightarrow \text{employee}\}$

$D_2 = [\text{INT num}; \text{CHAR sex}; \text{INT age}; \text{REAL salary}; \text{INT level}]$

$F_2 = \{\text{print}(\text{manager}) \rightarrow \text{manager}\}$

显然有 $\text{Class manager} \leq \text{Class employee}$

定义6 单重继承, 指一个类不能有两个或两个以上的超类。

说明 本文仅限于讨论单重继承, 以下的继承均指单重继承。

定理1 子类与超类之间的关系是一个偏序关系 ($\leq$), 满足自反性、传递性和反对称性。

证明: 设类 $C_1 = (D_1, F_1)$, 类 $C_2 = (D_2, F_2)$, 类 $C_3 = (D_3, F_3)$ 。

(1) 自反性

$\forall d [d \in D_1 \Rightarrow d \in D_1] \wedge \forall f [f \in F_1 \wedge f: \rightarrow C_1 \Rightarrow f \in F_1 \wedge f: \rightarrow C_1]$

$\therefore$ 类 C_1 是自身的一个子集, 即 $C_1 \leq C_1$ 。

(2) 传递性, 设类 C_1 是类 C_2 的子类, 类 C_2 又是类 C_3 的子类, 即: $C_1 \leq C_2, C_2 \leq C_3$, 则

$\forall d [d \in D_2 \Rightarrow d \in D_1] \wedge \forall f [f \in F_2 \wedge f: \rightarrow C_2 \Rightarrow f \in F_1 \wedge f: \rightarrow C_1]$

$\forall d [d \in D_3 \Rightarrow d \in D_2] \wedge \forall f [f \in F_3 \wedge f: \rightarrow C_3 \Rightarrow f \in F_2 \wedge f: \rightarrow C_2]$

即 $\forall d [d \in D_3 \Rightarrow d \in D_1] \wedge \forall f [f \in F_3 \wedge f: \rightarrow C_3 \Rightarrow f \in F_1 \wedge f: \rightarrow C_1]$

$\forall d [d \in D_2 \Rightarrow d \in D_1] \wedge \forall f [f \in F_2 \wedge f: \rightarrow C_2 \Rightarrow f \in F_1 \wedge f: \rightarrow C_1]$

$\therefore \forall d [d \in D_3 \Rightarrow d \in D_1] \wedge \forall f [f \in F_3 \wedge f: \rightarrow C_3 \Rightarrow f \in F_1 \wedge f: \rightarrow C_1]$

故类 C_1 是类 C_3 的子类, 即 $C_1 \leq C_3$ 。

(3) 反对称性

设类 C_1 是类 C_2 的子类, 类 C_2 又是类 C_1 的子类, 即: $C_1 \leq C_2, C_2 \leq C_1$, 则

$\forall d [d \in D_2 \Rightarrow d \in D_1] \wedge \forall f [f \in F_2 \wedge f: \rightarrow C_2 \Rightarrow f \in F_1 \wedge f: \rightarrow C_1]$

$\forall d [d \in D_1 \Rightarrow d \in D_2] \wedge \forall f [f \in F_1 \wedge f: \rightarrow C_1 \Rightarrow f \in F_2 \wedge f: \rightarrow C_2]$

$$\rightarrow C_2]$$

$$\therefore \forall d [d \in D_1 \Rightarrow d \in D_2] \wedge \forall d [d \in D_2 \Rightarrow d \in D_1]$$

$$\therefore D_1 = D_2$$

又 $\forall f [f \in F_1 \wedge f: \rightarrow C_1 \Rightarrow f \in F_2 \wedge f: \rightarrow C_2]$, 且不对 $f \in F_2$ 进行重载

$\forall f [f \in F_2 \wedge f: \rightarrow C_2 \Rightarrow f \in F_1 \wedge f: \rightarrow C_1]$, 且不对 $f \in F_2$ 进行重载

$$\therefore F_1 = F_2$$

$$\text{故 } C_1 = C_2.$$

3. 类集的非全序格

定义7 底类记为: $\perp$, 定义如下:

设类 $C = (D, F)$, 若 $\exists d \in D$ 无定义, 则称类 C 为底类, 即 $C = \perp$ 。

定义8 顶类记 T : 定义为: 设类 $C = (D, F)$, 若 $\forall C' = (D', F')$, $[[\forall d [d \in D \Rightarrow d \in D'] \wedge \forall f [f \in F \wedge f: \rightarrow C \Rightarrow f \in F' \wedge f: \rightarrow C']]]$, 则称类 C 为顶类, 即 $C = T$ 。

定义9 设类 $C_1 = (D_1, F_1)$, 类 $C_2 = (D_2, F_2)$, 类 C_1 与 C_2 的并记为 $C_1 \cup C_2$:

①若 $C_1 \leq C_2$ 或 $C_1 \leq C \dots \leq C_2$, 则令 $C_2 = C_1 \cup C_2$;

②若 $C_2 \leq C_1$ 或 $C_2 \leq C \dots \leq C_1$, 则令 $C_1 = C_1 \cup C_2$;

③若 $C_1 \leq C' \dots \leq C$, 且 $C_2 \leq C' \dots \leq C$, 则令 $C = C_1 \cup C_2$;

④否则令 $T = C_1 \cup C_2$ 。

定义10 设类 $C_1 = (D_1, F_1)$, 类 $C_2 = (D_2, F_2)$, 类 C_1 与 C_2 的交记为: $C_1 \square C_2$:

①若 $C_1 \leq C_2$ 或 $C_1 \leq C \dots \leq C_2$, 则令 $C_1 = C_1 \cup C_2$;

②若 $C_2 \leq C_1$ 或 $C_2 \leq C \dots \leq C_1$, 则令 $C_2 = C_1 \cup C_2$;

③否则令 $\perp = C_1 \cup C_2$ 。

定理2 类集 S 对于偏序关系 ($\leq$) 是一个偏序集合 ($S, \leq$), 构成一个非全序格。

证: 事实上, 类集 S 中的每一对元素 (C_1, C_2), 都有一个下确界 $\text{INF}(\{C_1, C_2\}) = C_1 \square C_2$, 和一个上确界 $\text{SUP}(\{C_1, C_2\}) = C_1 \cup C_2$ 。而底类 $\perp$ 和顶类 T , 则可分别作为类集 S 的下确界和上确界。

参考文献

- 1 李晓燕. 电脑常用高级语言辞典. 北京: 气象出版社, 1989
- 2 李晓燕. 程序设计语言的对象的结构模型. 华中师范大学学报(自然科学版), 1996, 30: 143~147
- 3 蔡希尧, 陈平. 面向对象技术. 西安: 西安电子科技大学出版社, 1995
- 4 陈其明. 数据库对象和型的广义结构模型. 计算机学报, 1989, 8: 562~577
- 5 Backus J W. Can Programmung be liberated from the Von Neumann style? A functional style and its algebra of programs. CACM, 1978, 21(8): 613~641