

继承的一个数学模型

瞿裕忠* 王志坚** 徐家福

(南京大学计算机软件研究所, 南京 210093)

摘要 提出了继承的一个数学模型, 引入图序类别型构用来表示面向对象语言中程序的代数结构, 将继承的形式语义刻划为图序类别型构集上的延拓函数. 通过该数学模型, 程序之代数结构的性质能反映语言的语法约束, 延拓函数的性质能揭示语言中继承的特点.

关键词 面向对象 继承 代数结构 图序类别型构 延拓

面向对象语言已成为大型软件开发的重要语言工具. 继承是面向对象语言的基本设施, 它使具有继承关系的不同类能共享特征(属性和程式统称为特征), 从而支持软件的易复用性和易扩充性. 在面向对象语言中, 继承的语义通常用自然语言进行描述. 为了深入理解和发 展面向对象语言与面向对象程序设计, 对继承进行形式语义刻划非常必要. Cardelli^[1,2]通过公式间序关系的推导规则来刻划继承, 这一逻辑方法受限于推导规则的强约束. Cook 等人^[3]将类解释为对象的生成函数, 将继承解释为生成函数的复合机制, 这一方法侧重于程式的具体实现, 特别是伪变量 self 的语义. Goguen 等人^[4]在多类别代数的基础上发展了有序类别代数, 以解释抽象数据类型间的子类型关系, 这一方法局限于具有较强限制的继承. 相关工作还有文献[5,6]. 综观已有的工作, 继承的形式语义刻划有待进一步研究. 本文旨在给出继承的一个数学模型, 该数学模型为刻划继承提供一个形式化方法, 并为继承的改进和完善打下一定的理论基础.

1 基本概念

本文将采用类别化集合这一概念. 给定一个“类别集” S , 一个 S -类别化集合(S -sorted set) A 是这样的一簇集合 $\{A_s | s \in S\}$, 其中 A_s 对于任意的 $s \in S$ 均为集合. 一个多类别型构(many-sorted signature)是一个二元组 (S, Σ) , 其中 S 为类别集, Σ 为形如 $\{\Sigma_{w,c} | w \in S^*, c \in S\}$ 的 $S^* \times S$ -类别化集合, 通常称 $\Sigma_{w,c}$ 中的元素为操作符. 在有些情况下, 类别集 S 上具有偏序结构. S 上的一个偏序结构是指 S 上的一个自反、传递和反对称的二元关系. 一个有序类别型构(order-sorted signature)是一个三元组 $(S, \leq, \Sigma)$, 其中 $(S, \leq)$ 是偏序集, (S, Σ) 是多类别型构, 且操作符满足下列单调性条件^[4]:

1994-12-07 收稿, 1995-07-18 收修改稿

* 现工作单位: 东南大学计算机系, 南京 210096

** 现工作单位: 河海大学计算机系, 南京 210098

若 $\sigma \in \Sigma_{s_1, w_1, c_1} \cap \Sigma_{s_2, w_2, c_2}$ 且 $w_1 \leq w_2$, 则 $c_1 \leq c_2$,

其中, 关系符 $\leq$ 在等长的 S^* 上的含义是 S 上偏序关系 $\leq$ 的自然推广:

$$s_1 \cdots s_n \leq t_1 \cdots t_n \text{ 当且仅当 } \forall 1 \leq i \leq n, s_i \leq t_i.$$

本文将使用有向无环图这一概念. 一个有向无环图 (directed acyclic graph) 是一个二元组 (S, R) , 其中 S 为有限集, R 为 S 上的一个二元关系, 且 $\bar{R}$ (R 的自反、传递闭包) 是反对称的, 且不存在 $s \in S$ 使 $s R s$. 其中

$\bar{R} \equiv \{(s, t) | \exists n \geq 1, \exists s_1, \dots, s_n, s_1 = s \wedge s_n = t \wedge \forall 1 \leq i < n, s_i R s_{i+1}\}$. 在一个有向无环图 (S, R) 中, 设 $s \neq t$, 且 $s \bar{R} t$, 则从 s 到 t 的一条路径是下列集合中的一个元素:

$$\text{Paths}(s, t) \equiv \{\gamma \in S^* | \gamma(1) = s \wedge \gamma(\text{len} \gamma) = t \wedge \exists \forall 1 \leq i < \text{len} \gamma, \gamma(i) R \gamma(i+1)\},$$

其中 $\text{len} \gamma$ 代表序列 γ 的长度.

为了定义继承的形式语义, 下面给出一些新的概念及其直观含义, 其中包括图序类别型构、程序的代数结构和图序类别型构的延拓等.

定义 1 (图序类别型构) 一个图序类别型构 (graph-sorted signature) 是一个三元组 (S, R, Σ) , 其中 (S, R) 是有向无环图, Σ 是形如 $\{\Sigma_{s, w, c} | s, c \in S, w \in S^*\}$ 的 $(S \times S^*) \times S$ -类别化集合, 且操作符满足下列单调性条件:

若 $\sigma \in \Sigma_{s_1, w_1, c_1} \cap \Sigma_{s_2, w_2, c_2}$ 且 $s_1 \bar{R} s_2$, 则 $c_1 \bar{R} c_2$.

注

(1) 本文假定类别集 S 均是有限集;

(2) 为简洁起见, 本文假定: 若 $\sigma \in \Sigma_{s, w, c} \cap \Sigma_{s, w', c'}$, 则 $w = w'$ 且 $c = c'$, 其直观含义是同一类中的特征不允许一名多用;

(3) GSS 也用来表示图序类别型构的全体;

(4) 对于一个图序类别型构 (S, R, Σ) , $(S, \bar{R}, \Sigma)$ 是一种特殊的有序类别型构.

在纯面向对象语言中, 程序通常是一组结构化的类, 从程序正文中可以抽取出下列信息: 程序中涉及的类名集、类间的直接继承关系、每个类中说明 (含重说明) 的特征名及其类型. 这些信息分别由下列集合表示: 一个类别集 S , S 上的一个有向无环图结构 R , 一个形如 $\{\Sigma_{s, w, c} | s, c \in S, w \in S^*\}$ 的 $(S \times S^*) \times S$ -类别化集合. 三元组 (S, R, Σ) 称为相应程序的代数结构, (S, R) 称为相应程序的继承图. 本文引入图序类别型构这一概念就是用来表示面向对象语言中程序的代数结构. 从面向对象语言的程序正文中抽取其代数结构的形式方法较简单, 因篇幅所限, 在此从略. 直观地看, 如果 (S, R, Σ) 是某一程序的代数结构, 则 $s \in S$ 表示 s 是相应程序中涉及的类名 (系统固有类或定义类), $s R t$ 表示相应程序中类 s 是类 t 的直接衍类, $\sigma \in \Sigma_{s, w, c}$ 表示 σ 在类 s 中被说明成型构为 $\langle w, c \rangle$ 的特征 (即参数类型序列为 w , 结果类型为 c), 而单调性条件反映了这样的事实, 即程序中衍类对基类的特征重说明要求结果类型比原来的要缩小. 有些面向对象语言对特征重说明的参数类型也有所限制, 如 Eiffel 语言^[7] 要求参数类型也要缩小, 而 Fun 语言^[2] 要求参数类型放大, 为此分别定义协变图序类别型构和反变图序类别型构如下:

定义 2 (协变图序类别型构) 一个图序类别型构 (S, R, Σ) 称为是协变图序类别型构 (covariantly graph-sorted signature), 当且仅当操作符满足下列协变单调性条件:

若 $\sigma \in \Sigma_{s_1, w_1, c_1} \cap \Sigma_{s_2, w_2, c_2}$ 且 $s_1 \bar{R} s_2$, 则 $w_1 \bar{R} w_2$ 且 $c_1 \bar{R} c_2$.

定义 3 (反变图序类别型构) 一个图序类别型构 (S, R, Σ) 称为是反变图序类别型构 (contravariantly graph-sorted signature), 当且仅当操作符满足下列反变单调性条件:

若 $\sigma \in \Sigma_{s_1, w_1, c_1} \cap \Sigma_{s_2, w_2, c_2}$ 且 $s_1 \bar{R} s_2$, 则 $w_2 \bar{R} w_1$ 且 $c_1 \bar{R} c_2$.

在大多数强类型面向对象语言中, 对于程序的代数结构 (S, R, Σ) , Σ 中的操作符通常满足单调性条件, 有的甚至满足协变单调性条件或反变单调性条件, 如分别在 Eiffel 语言 (剔除特征重命名因素) 和 Fun 语言中.

定义 4 (图序类别型构中类别上的操作符) 设 (S, R, Σ) 是一个图序类别型构, 且 $s \in S$, 如果存在 $w \in S^*$ 和 $c \in S$ 使得 $\sigma \in \Sigma_{s, w, c}$, 则称 σ 为 (S, R, Σ) 中类别 s 上的操作符, 简称为类别 s 上的操作符. 本文用 Σ_s 表示 (S, R, Σ) 中类别 s 上的操作符全体, 即

$$\Sigma_s = \{\sigma \mid \exists w \in S^*, \exists c \in S, \sigma \in \Sigma_{s, w, c}\}.$$

直观地看, $\sigma \in \Sigma_s$ 表示 σ 是在类 s 中说明的特征. 这样, Σ 中的操作符全体按其所在的类别被划分为形如 $\{\Sigma_s \mid s \in S\}$ 的一组集合. 由此可见, 区别于有序类别型构, 图序类别型构中的操作符总是在某一 (或某些) 类别上. 换言之, 程序中的特征必须在类中说明. 同一操作符出现在多个类别上体现了操作符的多态性. 另外, 操作符满足的单调性条件反映了多数面向对象语言对特征重说明的语法约束. 当然, 这一单调性条件与有序类别型构中的单调性条件也有所区别. 以下给出图序类别型构的延拓概念.

定义 5 (图序类别型构的延拓) 设 (S, R, Σ) 和 (S, R, Σ') 是有同一有向无环图结构的两个图序类别型构, 如果 $\Sigma_{s, w, c} \subseteq \Sigma'_{s, w, c}$ 对一切的 $w \in S^*, s, c \in S$ 均成立, 则称 (S, R, Σ') 是 (S, R, Σ) 的延拓 (Extension).

2 继承之形式语义的定义方法

本节给出继承之形式语义的一种定义方法. 将图序类别型构集上的延拓函数作为继承的形式语义, 并研究程序之代数结构和延拓函数的性质.

继承是面向对象语言的语言机制. 通过语法子句, 可知一个类是否可以有多个直接基类, 从而将继承区分为单继承 (如 Smalltalk-80^[9]) 和多继承. 多继承通常被划分为 3 种^[9, 10]: 图式继承 (如 Extended Smalltalk-80^[11] 和 Eiffel)、线性继承 (如 Flavors^[12])、和树式继承 (如 C++^[13]). 关于各种继承的自然语言描述可参见有关文献[7, 8, 11 ~ 14]. 简单地说, 继承使一个类可以拥有不在该类说明的特征. 这样, 一个类的特征包括两部分: 在该类中说明的 (最新引入或重说明) 和直接继承的. 以下考虑将继承的语义形式化.

上节引入了图序类别型构来表示程序的代数结构, 对一个程序 p , 它的代数结构可用一个图序类别型构 (S, R, Σ) 来表示, 其中 S 表示程序 p 中定义的和系统固有类的类名集; R 表示衍类与直接基类间的关系 ($\bar{R}$ 表示衍类与基类间的关系); Σ 是形如 $\{\Sigma_{s, w, c} \mid s, c \in S, w \in S^*\}$ 的 $(S \times S^*) \times S$ 一类别化集合, 使得 $\sigma \in \Sigma_{s, w, c}$ 表示 σ 在类 s 中说明成型构为 $\langle w, c \rangle$ 的特征. 继承机制使得基于层次结构 R 的不同类共享特征, 从而, 除了在该类说明的特征外, 一个类还可以拥有不在该类说明的特征, 因此 Σ 有一个延拓 Σ^e , 使 $\sigma \in \Sigma^e_{s, w, c}$ 表示类 s 拥有名为 σ 、型构为 $\langle w, c \rangle$ 的特征. 显然, Σ^e 是按某一确定的方法由 (S, R, Σ) 构造而得. 这一构造方法代表了继承之所在, 并且可由代数结构上的一个形如 $\text{Ext}: \text{GSS} \rightarrow \text{GSS}$ 的函数来表示, 该函数称为图序类别型构集上的一个延拓函数. 因此, 本文用图序类别型构集上的延拓函数来表示继承

的形式语义. 值得注意的是: 对于不同的语言, 其程序的代数结构通常具有不同的性质 (主要由于解决特征名冲突的方法和特征重说明的语法约束不同), 且表示继承之形式语义的延拓函数随着给定语言之继承方式的不同而不同.

下面研究图式继承语言中程序之代数结构的性质和图式继承的形式语义. 在图式继承语言的程序中, 对一个衍类而言, 如果不同直接基类拥有的同名特征源自不同的基类, 则需在该衍类中重说明. 因此程序的代数结构应为强正规图序类别型构, 其定义如下:

定义 6 (强正规图序类别型构) 设 (S, R, Σ) 是图序类别型构, 称 (S, R, Σ) 是强正规图序类别型构, 当且仅当下列强正规性条件 (strong Normality condition) 成立:

对任给的 $s, t \in S (s \neq t)$, 若 $u \in \Psi(s, t)$, 则 $\Sigma_s \cap \Sigma_t \subseteq \Sigma_u$,

其中 $\Psi(s, t) \equiv \{u \in S | u \bar{R} s \wedge u \bar{R} t \wedge u \neq s \wedge u \neq t \wedge$

$\exists \gamma \in \text{Paths}(u, s), \exists \gamma' \in \text{Paths}(u, t), \text{mg } \gamma \cap \text{mg } \gamma' = \{u\}\}$,

这里 $\text{mg } \gamma$ 代表序列 γ 中元素的集合.

直观地, 如果从衍类 u 到两个不同的基类 s 和 t 有两条互不交叉的继承路径 γ 和 γ' , 那么基类 s 和 t 中说明的同名特征在衍类 u 中须重说明. 这一强正规性条件可用于解决同名继承特征的冲突. 为了深入研究图式继承语言中程序之代数结构的性质, 现引入正规图序类别型构概念如下:

定义 7 (正规图序类别型构) 设 (S, R, Σ) 是图序类别型构, 则称 (S, R, Σ) 是正规图序类别型构, 当且仅当下列正规性条件 (Normality condition) 成立:

对任给的 σ 和 s , 如果 $\{t | s \bar{R} t \wedge \sigma \in \Sigma_t\}$ 非空则有最小元.

命题 1 强正规图序类别型构一定是正规图序类别型构.

证 设 (S, R, Σ) 是一个强正规图序类别型构, 对任给的 σ 和 s , 设 $\{t | s \bar{R} t \wedge \sigma \in \Sigma_t\}$ 非空, 下面用反证法证明 $\{t | s \bar{R} t \wedge \sigma \in \Sigma_t\}$ 有最小元.

假设不然, 由 $(S, \bar{R})$ 的有限性得: 存在 $t_1, t_2 (t_1 \neq t_2)$ 是该集合的极小元, 显然 $s \neq t_1, s \neq t_2$, 从而存在 γ_1, γ_2 分别是 s 到 t_1 和 t_2 的两条路径, 可分两种情形: (i) γ_1 和 γ_2 不交叉, 从而 $s \in \Psi(t_1, t_2)$, 但 $\sigma \in \Sigma_s$. 显然不成立, 与强正规性矛盾; (ii) γ_1 和 γ_2 相互交叉, 令这些相交结点中的最大元为 d (其存在性在于这些相交结点同在一条路径上), 则 $d \in \Psi(t_1, t_2)$. 但是由 t_1 和 t_2 的极小性得 $\sigma \in \Sigma_d$ 不成立, 与强正规性矛盾. 无论哪种情形均推出矛盾, 从而命题得证.

图式继承语言一般通过语法约束使程序的代数结构具有强正规性, 从而也具有正规性. 但是, 正规性并不保证强正规性. 例如满足下列性质的图序类别型构 $(\{s, t, u\}, \{(s, t), (u, t), (u, s)\}, \Sigma)$ 是正规图序类别型构, 但并不是强正规图序类别型构:

$$\Sigma_s = \{\sigma_1, \sigma_2\}, \Sigma_t = \{\sigma_1, \sigma_3\}, \Sigma_u = \emptyset.$$

下面定义图序类别型构集上的延拓函数 GExt 来表示图式继承的形式语义.

定义 8 (延拓函数 GExt) 函数 $\text{GExt}: \text{GSS} \rightarrow \text{GSS}$ 定义如下:

$$\text{GExt}((S, R, \Sigma)) \equiv (S, R, \Sigma^G),$$

Σ^G 形如 $\{\Sigma_{s,w,c}^G | s, c \in S, w \in S^*\}$ 的 $(S \times S^*) \times S$ -类别化集合, 其中 $\sigma \in \Sigma_{s,w,c}^G$ 当且仅当下列条件均满足:

(1) $\{t | s \bar{R} t \wedge \sigma \in \Sigma_t\}$ 是非空集, 且有最小元 (记为 $1(s, \sigma)$),

(2) $\sigma \in \Sigma_{1(s, \sigma), w, c}$

注 (1) $(S, \bar{R})$ 的有限性能保证 $\Sigma_{s,w,c}^G$ 的良定性, Σ 中操作符满足的单调性条件能保证 Σ^G 中的操作符满足单调性条件 (其证明类似于命题 2 的证明), 所以函数 GExt 也是良定的.

(2) 从上述定义易知: $\Sigma_{s,w,c} \subseteq \Sigma_{s,w,c}^G$, 因此 (S, R, Σ^G) 是 (S, R, Σ) 的延拓, 定义中采用的函数名 GExt 隐含这一函数是延拓函数 (Extension function), 且这一函数是为了刻画图式继承 (Graph-Oriented Inheritance) 的, “G” 取自图式继承的第一个英文字母.

命题 2 设 (S, R, Σ) 是图序类别型构, 则

(1) 如果 (S, R, Σ) 是协变图序类别型构, 那么 (S, R, Σ^G) 也是协变图序类别型构;

(2) 如果 (S, R, Σ) 是反变图序类别型构, 那么 (S, R, Σ^G) 也是反变图序类别型构.

证 先证 $1(s, \sigma)$ 关于 s 的单调性, 即对任意的 $s_1 \bar{R} s_2$, 若 $1(s_1, \sigma)$ 和 $1(s_2, \sigma)$ 均存在, 则 $1(s_1, \sigma) \bar{R} 1(s_2, \sigma)$. 因为 $1(s_2, \sigma)$ 是 $\{u | s_2 \bar{R} u \wedge \sigma \in \Sigma_u\}$ 中的元素, 而 $s_1 \bar{R} s_2$, 所以 $1(s_2, \sigma)$ 属于 $\{u | s_1 \bar{R} u \wedge \sigma \in \Sigma_u\}$, 由 $1(s_1, \sigma)$ 的最小性得 $1(s_1, \sigma) \bar{R} 1(s_2, \sigma)$.

下面证明结论 (1). 对任给的 $\sigma \in \Sigma_{s_1, w_1, c_1}^G \cap \Sigma_{s_2, w_2, c_2}^G$, 且 $s_1 \bar{R} s_2$, 现将 $1(s_1, \sigma)$ 和 $1(s_2, \sigma)$ 分别简记为 t_1 和 t_2 , 则由 $1(s, \sigma)$ 关于 s 的单调性得 $t_1 \bar{R} t_2$. 另外由定义 8 得 $\sigma \in \Sigma_{t_1, w_1, c_1}$ 以及 $\sigma \in \Sigma_{t_2, w_2, c_2}$, 由 (S, R, Σ) 的协变单调性得出 $w_1 \bar{R} w_2$ 和 $c_1 \bar{R} c_2$. 从而结论 (1) 得证.

结论 (2) 的证明类似于 (1) 的证明, 从略.

命题 3 设 $(S, R, \Sigma) \in \text{GSS}$, 若它具有正规性, 则

(1) 对任意的 $s, c \in S, w \in S^*$,

$$\Sigma_{s,w,c}^G = \{\sigma | (\exists t, s \bar{R} t \wedge \sigma \in \Sigma_t) \wedge \sigma \in \Sigma_{1(s,\sigma),w,c}\},$$

其中 $1(s, \sigma)$ 代表 $\{t | s \bar{R} t \wedge \sigma \in \Sigma_t\}$ 中的最小元;

(2) 对任意的 $s \in S, \Sigma_s^G = \{\sigma \in \Sigma_t | s \bar{R} t\}$;

(3) 对任意的 $s \in S, \Sigma_s^G = \Sigma_s \cup \{\sigma \in \Sigma_t^G | s \bar{R} t\}$;

(4) 对任意的 $s \bar{R} t, \Sigma_t^G \subseteq \Sigma_s^G$.

注 (1) 对于正规图序类别型构, 如果 $\{t | s \bar{R} t \wedge \sigma \in \Sigma_t\}$ 非空则有最小元, 因此结论 (1) 易证. 不难从结论 (1) 推出结论 (2) 和 (3), 从结论 (2) 推出 (4). 因而详细证明从略. Σ_s 和 Σ_s^G 的直观含义如下: Σ_s 表示在类 s 中说明的特征名集, 而 Σ_s^G 表示类 s 所拥有的 (包含在类 s 中说明的以及直接继承来的) 特征名集.

(2) 上述命题表明: 图序类别型构 (S, R, Σ) 的正规性能保证延拓后的图序类别型构 (S, R, Σ^G) 具有良好的性质. 不难看出, 若图序类别型构 (S, R, Σ) 不具有正规性, 则不能保证命题中的结论成立.

命题 4 设 $(S, R, \Sigma) \in \text{GSS}$, 若它是强正规的, 则

$$\Sigma_{s,w,c}^G = \Sigma_{s,w,c} \cup \{\sigma | \sigma \notin \Sigma_s \wedge (\exists t, s \bar{R} t \wedge \sigma \in \Sigma_{t,w,c}^G)\},$$

对任意的 $s, c \in S, w \in S^*$ 均成立.

证 先证左边包含于右边. 对任给的 $\sigma \in \Sigma_{s,w,c}^G$, 由定义 8 得 $\{t | s \bar{R} t \wedge \sigma \in \Sigma_t\}$ 为非空集, 且有最小元 $1(s, \sigma)$, 且 $\sigma \in \Sigma_{1(s,\sigma),w,c}$. 分两种情形, (1) 若 $\sigma \in \Sigma_s$, 则 $1(s, \sigma) = s$, 从而 $\sigma \in \Sigma_{s,w,c}$; (2) 若 σ 不属于 Σ_s , 则 $1(s, \sigma)$ 不等于 s , 从而存在 t 使得 $s \bar{R} t$ 且 $t \bar{R} 1(s, \sigma)$, 容易推得 $1(t, \sigma)$ 等于 $1(s, \sigma)$, 据定义 8 得 $\sigma \in \Sigma_{t,w,c}^G$.

再证右边包含于左边. 显然 $\Sigma_{s,w,c}$ 包含于 $\Sigma_{s,w,c}^G$. 如果 σ 不属于 Σ_s , 且存在满足 $s \bar{R} t$ 的 t 使得 $\sigma \in \Sigma_{t,w,c}^G$, 须证 $\sigma \in \Sigma_{s,w,c}^G$. 由 $\sigma \in \Sigma_{t,w,c}^G$ 知 $\{u | t \bar{R} u \wedge \sigma \in \Sigma_u\}$ 为非空集, 且有最小元

$1(t, \sigma)$, 而且 $\sigma \in \Sigma_{1(t, \sigma), w, c}$. 下面用反证法证明 $1(s, \sigma)$ 等于 $1(t, \sigma)$, 假设不然, 由 σ 不属于 Σ_s 知 $1(s, \sigma)$ 不等于 s , 从而存在一条从 s 到 $1(s, \sigma)$ 且不经过 t 的路径 γ , 使得对于路径 γ 上除 $1(s, \sigma)$ 以外的所有结点 v 均有 σ 不属于 Σ_v . 按 $s R t$ 以及 $1(t, \sigma)$ 的定义知: 存在一条从 s 到 $1(t, \sigma)$ 且经过 t 的路径 γ' 使得对于路径 γ' 上除 $1(t, \sigma)$ 以外的所有结点 u 均有 σ 不属于 Σ_u , 当然 $1(t, \sigma)$ 有可能等于 t . 可分两种情形: (i) γ 和 γ' 不交叉, 而 σ 属于 $\Sigma_{1(s, \sigma)}$ 且 σ 属于 $\Sigma_{1(t, \sigma)}$, 从 (S, R, Σ) 的强正规性推出 $\sigma \in \Sigma_s$, 矛盾; (ii) γ 和 γ' 相互交叉, 令这些相交结点中的最大元为 d (其存在性在于这些相交结点同在同一条路径上), 则 (S, R, Σ) 的强正规性要求 $\sigma \in \Sigma_d$, 与 $1(s, \sigma)$ 的最小性矛盾. 两种情形均推出矛盾. 因此 $1(s, \sigma)$ 等于 $1(t, \sigma)$, 从而由定义 8 得 $\sigma \in \Sigma_{s, w, c}^G$. 命题得证.

注 (1) 上述命题表明: 如果程序的代数结构满足强正规性, 那么继承的语义更容易被理解, 因为延拓函数能简单地被递归定义. 直观地看, 类 s 所拥有的型构为 $\langle w, c \rangle$ 的特征要么是在类 s 中说明的型构为 $\langle w, c \rangle$ 的特征, 要么是某一直接基类所拥有的型构为 $\langle w, c \rangle$ 的特征.

(2) 从上述命题的证明不难看出: 正规性约束并不能保证命题中结论成立. 但是图序类别型构的正规性约束比强正规性约束要弱, 且不失为一种更自然的语法约束.

对于类 s 中程式 σ , 它可能是源自基类的程式, 当程式 σ 被调用, 且目标对象为类 s 的某一对象时, 被执行的是类 s 的程式 σ 的实现版本. 因此, 我们要确定每个类所拥有特征的实现版本. 形式地, 对任意的 $s \in S$ 和 $\sigma \in \Sigma_s^G$, 应查找操作符 σ 之实现版本所在的类别, 该类别即为 $\{t | s \bar{R} t \wedge \sigma \in \Sigma_t\}$ 中的最小元 $1(s, \sigma)$. 如果 (S, R, Σ) 具有强正规性, 那么下列函数可用来确定类别 s 所拥有操作符 σ 的实现版本.

定义 9 (Σ^G 中操作符之实现版本的确定函数 IV) 对于强正规图序类别型构 (S, R, Σ) , IV: $\sum_{s \in S} \Sigma_s^G \rightarrow S$,

$$\begin{aligned} \text{IV}((s, \sigma)) &\equiv \text{if } \sigma \in \Sigma_s \\ &\quad \text{then } s \\ &\quad \text{else Let } (t_1, \dots, t_n) = \{t | s \bar{R} t \wedge \sigma \in \Sigma_t^G\} \\ &\quad \text{In } \text{IV}((t_1, \sigma)). \end{aligned}$$

注 直观地, $\text{IV}(s, \sigma)$ 代表类 s 所拥有特征 σ 的实现版本所在的类, IV 是递归定义的. 如果类 s 说明了特征 σ , 那么为 s , 否则, 找出类 s 的直接基类中拥有特征 σ 的那些类, 记为 $t_1, \dots, t_n$, 从中任取一个, 不妨取 t_1 , 递归地确定类 t_1 所拥有特征 σ 的实现版本所在的类, 该类即为类 s 所拥有特征 σ 的实现版本所在的类. IV 的良好性由 (S, R, Σ) 的强正规性得到保证. 事实上, 可以证明 $\text{IV}((s, \sigma))$ 就是 $\{t | s \bar{R} t \wedge \sigma \in \Sigma_t\}$ 中的最小元.

程序之代数结构的正规性值得注意, 它对于解决继承特征的名冲突和确定特征的实现版本是非常本质的. 图序类别型构的正规性类似于有序类别型构的正则性 (Regularity), 一个有序类别型构 $(S, \leq, \Sigma)$ 被称为是正则的, 当且仅当下列正则性条件成立^[6]:

对 $\sigma \in \Sigma_{w_1, c_1}$ 和 $w_0 \leq w_1$, $\{w | w_0 \leq w \wedge \exists c, \sigma \in \Sigma_{w, c}\}$ 有最小元.

直观地看, 有序类别代数理论将类别间的序关系解释为行为上保持一致的子类型关系, 对于多态操作符的语义解释须保持一致, 有序类别型构的正则性就是为了保证这样的一致性所需的语法约束. 然而在面向对象语言中, 类间的继承关系无需是行为上保持一致的子类型关系, 因而多态操作符的语义无须保持一致. 但是, 由于多继承的引入 (再加上动态定连), 对于多态操作符号语义解释可能会出现歧义性. 为了避免这样的歧义性, 程序之代数结构的正规

性不失为一种较自然的语法约束。

而程序之代数结构的强正规性约束比正规性约束要强,虽然它能使继承的语义更容易理解,但是这一约束限制了类库的层次结构,从而降低了类的复用性。本文认为:正规性约束相对于强正规性约束放宽了对类库层次结构的限制,而且也能保证类型系统具有较好的性质。

3 结语

本文提出了继承的一个数学模型,该数学模型包括程序的代数结构、图序类别型构、图序类别型构的延拓等概念,以及继承之形式语义的定义方法。通过这个数学模型,程序之代数结构的性质能反映语言的语法约束,延拓函数的性质能揭示语言中继承的特点,从而各种继承能得到恰当的形式刻画,本文着重研究了图式继承的形式语义以及相关性质。

文中提出的图序类别型构和正规图序类别型构等概念,推广了有序类别代数理论中的有序类别型构和正则有序类别型构等概念。本文的主要思想与有序类别代数理论的区别主要在于:有序类别代数将有序类别型构中类别间的偏序关系解释为行为上保持一致的子类型关系;而在本文提出的继承之数学模型中,程序之代数结构中类别集上的图结构用于图序类别型构的延拓,使得子类别上的操作符增加,从而继承的语义通过图序类别型构集上的延拓函数得到恰当的形式刻画。

通过这一数学模型,继承能深入地被理解,继承的设计和实现能得到改进。进一步的工作将研究更多的语言成分,如类的不变式,在这种情形下,程序的代数结构中需增加一组公理,延拓函数也要作相应的调整,其难点是不同基类中不变式的相容性问题。

致谢 南京大学软件研究所伊波、陈道蓄、吕建教授和费宗铭博士对本文工作进行过有益的讨论,并提出建议,作者深表谢意。

参 考 文 献

- 1 Cardelli L. A semantics of multiple inheritance. *Information and Computation*, 1988, 76: 138 ~ 164
- 2 Cardelli L, Wegner P. On understanding types, data, abstraction and polymorphism. *Computing Surveys*, 1985, 17(4): 471 ~ 522
- 3 Cook W, Palsberg J. A denotational semantics of inheritance and its correctness. *SIGPLAN Notices*, 1989, 24(10): 433 ~ 441
- 4 Goguen J A, Meseguer J. Order-sorted algebra I: equation deduction for multiple inheritance, overloading, exception and partial operations. *Theor Comp Science*, 1992, 105(2), 25 ~ 36
- 5 Jagannathan S, Agha G. A reflective model of inheritance. *Proc ECOOP'92 or LNCS 615*, 350 ~ 371
- 6 Walker D. π -Calculus semantics of object-oriented programming languages. *Proc Inter Conf TACS'91*, 532 ~ 547
- 7 Meyer B. *Eiffel: The Language*. Prentice-hall, 1992
- 8 Goldberg A, Robson D. *Smalltalk-80: The Language and Implementation*. Addison Wesley, 1983
- 9 Snyder A. Inheritance and development of encapsulated software components. In: Shriver B, Wegner P eds. *Research Directions in Object-Oriented Programming*. MIT Press, 1987, 165 ~ 188
- 10 Singh G B. Single versus multiple inheritance in object oriented programming. *Oops Messenger*, 1994, 5(1): 34 ~ 43
- 11 Borning A H, Ingalls H. Multiple inheritance in smalltalk-80. *Proc National Conf on Arti Intel AAAI'82*
- 12 Moon D A. Object oriented programming with flavors. *Proc ACM Conf on OOPSLA'86*, Published as *Sigplan Notices* sept, 1986, 21(11): 1 ~ 8
- 13 Stroustrup B. *The C++ programming language*. Addison Wesley, Menl Park (CA), 1991
- 14 徐家福,王志坚,翟成祥. 对象式程序设计语言. 南京:南京大学出版社, 1992