第 42 卷 第 6 期
2015 年 6 月

计 算 机 科 学
Computer Science

Vol. 42 No. 6
June 2015

归纳数据类型的范畴论方法

苗德成[1] 奚建清[2] 苏锦钿[2]
（韶关学院数学与信息科学学院 韶关 512005）[1] （华南理工大学计算机科学与工程学院 广州 510640）[2]

摘 要 归纳数据类型是类型论研究的重要分支，传统的数理逻辑或代数方法侧重于描述归纳数据类型的有限语法构造，在语义性质与归纳规则的分析与设计方面存在一定的不足。基于范畴论的方法，在集合范畴的框架内给出谓词的形式化定义，分析谓词范畴与代数范畴的构成与性质，并探讨集合范畴上自函子到谓词范畴上自函子的提升，最后利用伴随函子及其伴随性质深入分析了归纳数据类型具有普适意义的归纳规则。

关键词 归纳数据类型，范畴论，谓词范畴，提升，伴随函子

中图法分类号 TP301.2 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.6.002

Category Theoretical Method of Inductive Data Types

MIAO De-cheng[1] XI Jian-qing[2] SU Jin-dian[2]
(School of Math and Information, Shaoguan University, Shaoguan 512005, China)[1]
(School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China)[2]

Abstract Inductive data types are important research branch of type theory, and traditional methods including mathematical logic and algebra focus on describing finite syntax construction for inductive data types, resulting in some deficiencies for analyzing and designing semantics properties and inductive rules. This paper provided formal definition of predicate in the framework of set category by category theoretical methods, analyzed the construction and properties of predicate category and algebra category, and probed lifting of endofunctors from set category to predicate category, and at last deeply researched universal inductive rules of inductive data types by adjoint functor and its adjoint properties.

Keywords Inductive data types, Category theory, Predicate category, Lift, Adjoint functor

1 引言

程序语言的数据类型可以递归定义。递归数据类型是类型表达式的同构解，而作为类型同构表达式最小解的归纳数据类型(Inductive Data Types)是一类重要的递归数据类型，在程序语言中有许多应用，如描述流、表、树及带环无限循环图等复杂的数据结构，支持协同进程演算及控制过程等。

传统归纳数据类型的研究以数理逻辑或代数方法为主[1-3]，侧重于描述归纳数据类型的有限语法构造，将归纳数据类型及操作封装在 Σ-代数结构内，但在语义性质与归纳规则描述等方面仍存在一定的不足。范畴论在形式语言建模理论、形式语义学和程序设计方法等计算机科学基础研究领域有着广泛的应用，为理解计算机体系结构、软件规范和程序间联系提供一种通用的理论工具、思维方法和研究手段。基于范畴论的观点，归纳数据类型抽象为函子的初始代数，由初始代数的初始性定义归纳数据类型一个递归的折叠操作 *fold*[4]，满足一系列代数运算，在程序计算逻辑与语义转换等研究中具有重要应用[5-7]。

本文的主要工作是应用范畴论方法研究归纳数据类型的语义性质及归纳规则。第 2 节首先在集合范畴的框架内给出谓词的形式化定义，对谓词范畴和代数范畴进行描述与证明，并分析相应的折叠操作；第 3 节探讨集合范畴到谓词范畴自函子的提升；第 4 节通过对代数范畴间一对伴随函子的研究深入分析归纳数据类型具有普适意义的归纳规则；第 5 节分析当前归纳数据类型研究领域的一些相关工作；最后总结全文并给出后续研究工作。

2 谓词范畴与代数范畴

折叠操作 *fold* 对归纳数据类型的递归操作建立语义模型，谓词描述归纳数据类型的语义性质，下面给出谓词的形式化定义。本文假定读者具备函子、自然转换与伴随等概念的范畴论基础，记 $Obj\ \mathbb{C}$ 为范畴 $\mathbb{C}$ 的对象集，$Mor\ \mathbb{C}$ 为范畴 $\mathbb{C}$ 的态射集。

定义 1 设 Set 为集合范畴，$\forall X\in Obj\ Set$，X 上的一个谓词是一个二元组 $\langle X,P\rangle$，即 $P:X\to Set$。对 $\forall x\in X$，Px 构成一个集合，称集合 X 为谓词 $\langle X,P\rangle$ 的定义域。

谓词 $\langle X,P\rangle$ 到 $\langle X',P'\rangle$ 的态射是一个序对 $(f,f^{\sim}):\langle X,P\rangle\to\langle X',P'\rangle$，其中 $f:X\to X'$ 是相应谓词定义域上的函数，而 $f^{\sim}:\forall x\in X(Px\to P'(fx))$ 将描述 χ 语义性质的 Px 映射为 $P'(fx)$。

到稿日期：2014-06-04 返修日期：2014-09-20 本文受国家自然科学基金项目(61103038)，广东省自然科学基金项目(S2013010015944，2012A010701011)，韶关市科技计划项目(2013CX/K61)资助。

苗德成(1979－)，男，博士，副教授，主要研究方向为形式语言建模理论、范畴论方法，E-mail：tony10860@126.com；**奚建清**(1962－)，男，博士，教授，主要研究方向为数据库与网络计算、软件体系结构；**苏锦钿**(1980－)，男，博士，副教授，主要研究方向为形式化方法、双代数。

定理 1 以谓词为对象，以谓词态射为态射构成谓词范畴$\mathbb{P}$。

证明：设 $dom: Mor\ \mathbb{P} \to Obj\ \mathbb{P}$ 为域函数，$cod: Mor\ \mathbb{P} \to Obj\ \mathbb{P}$ 为共域函数，$\circ: Mor\ \mathbb{P} \times Mor\ \mathbb{P} \to Mor\ \mathbb{P}$ 为函数间的复合关系。下面证明系统$\mathbb{P}=(Obj\ \mathbb{P}, Mor\ \mathbb{P}, dom, cod, \circ)$是一个范畴。

设$\langle X,P\rangle, \langle X',P'\rangle, \langle X'',P''\rangle \in Obj\ \mathbb{P}$，$(f,f^{\sim}): \langle X,P\rangle \to \langle X',P'\rangle \in Mor\ \mathbb{P}$，$(g,g^{\sim}): \langle X',P'\rangle \to \langle X'',P''\rangle \in Mor\ \mathbb{P}$，则$(g,g^{\sim})\circ(f,f^{\sim}): \langle X,P\rangle \to \langle X'',P''\rangle \in Mor\ \mathbb{P}$，所以 $dom((g,g^{\sim})\circ(f,f^{\sim}))=\langle X,P\rangle=dom(f,f^{\sim})$，且 $cod((g,g^{\sim})\circ(f,f^{\sim}))=\langle X'',P''\rangle=cod(g,g^{\sim})$，满足匹配条件。

令$(h,h^{\sim}): \langle X'',P''\rangle \to \langle X''',P'''\rangle \in Mor\ \mathbb{P}$，则$(h,h^{\sim})\circ(g,g^{\sim}): \langle X',P'\rangle \to \langle X''',P'''\rangle \in Mor\ \mathbb{P}$，有$(h,h^{\sim})\circ((g,g^{\sim})\circ(f,f^{\sim})): \langle X,P\rangle \to \langle X''',P'''\rangle$，而$((h,h^{\sim})\circ(g,g^{\sim}))\circ(f,f^{\sim}): \langle X,P\rangle \to \langle X''',P'''\rangle$，所以$(h,h^{\sim})\circ((g,g^{\sim})\circ(f,f^{\sim}))=((h,h^{\sim})\circ(g,g^{\sim}))\circ(f,f^{\sim})$，满足结合律条件。

记 id 为单位态射，谓词$\langle X,P\rangle$存在唯一单位谓词态射 $id_{\langle X,P\rangle} \in Mor\ \mathbb{P}$，使得 $dom(id_{\langle X,P\rangle})=cod(id_{\langle X,P\rangle})=\langle X,P\rangle$，且对任意谓词态射$(f,f^{\sim}) \in Mor\ \mathbb{P}$，若 $dom(f,f^{\sim})=\langle X,P\rangle$，则$(f,f^{\sim})\circ id_{\langle X,P\rangle}=(f,f^{\sim})$；若 $cod(f,f^{\sim})=\langle X,P\rangle$，则 $id_{\langle X,P\rangle}\circ(f,f^{\sim})=(f,f^{\sim})$，满足等式射存在条件。

由范畴的定义[8,9]知，系统$\mathbb{P}=(Obj\ \mathbb{P}, Mor\ \mathbb{P}, dom, cod, \circ)$是一个范畴，即以谓词为对象，以谓词态射为态射构成谓词范畴$\mathbb{P}$。证毕。

迭代(Iteration)操作为归纳数据类型的递归计算提供一种自然与规范的描述方式，使程序具有简洁性和可读性等良好性质，并便于代码重用。从范畴论的角度分析，迭代操作源自归纳数据类型的初始代数语义，而归纳数据类型则被视为初始代数的载体(Carriers)。

定义 2 设 F 为范畴$\mathbb{C}$上的一个自函子，对$\forall X \in Obj\ \mathbb{C}$，存在一个 F-代数 $h: FX \to X \in Mor\ \mathbb{C}$，称 X 为 F-代数 h 的载体。

F-代数 h 与 $h': FX' \to X'$的 F-代数态射是其载体间的一个映射 $f: X \to X'$，且满足图表交换 $f\circ h=h'\circ Ff$。

定理 2 以 F-代数为对象，以 F-代数态射为态射构成代数范畴 Alg_F。

证明与定理 1 证明类似，略。

Alg_F 中若存在初始代数 $in: F(\mu F) \to \mu F$，则 in 是唯一同构的，并且其载体 μF 是函子 F 的最小不动点(Least Fixed Point)[10]。初始代数 in 的初始性确保 in 到任意 F-代数 $h: FX \to X$ 存在一个唯一的 F-代数态射 $fold\ h: \mu F \to X$，如图 1 所示。折叠操作 $fold$ 源自初始代数语义，可对任意归纳数据类型建模，如取 $F(\mu F)$中任一实例 m，则有：$fold\ h\ (in\ m)=h(F(fold\ h)m)$成立。

$$\begin{tikzcd}
F(\mu F) \arrow[r, "F(fold\ h)"] \arrow[d, "in"'] & FX \arrow[d, "h"] \\
\mu F \arrow[r, "fold\ h"'] & X
\end{tikzcd}$$

图 1 初始代数及其初始性

例 1 记自然数类型为 Nat，令谓词 $P: Nat \to Set$ 表示自然数类型的语义性质，将$\forall n \in Nat$ 映射为n 所满足的语义性质，即以 Pn 描述自然数n 的语义集。设 $Succ$ 为自然数的后继函数，则有 Nat 的归纳规则：

$Ind_{Nat}: \forall(P: Nat \to Set)P0 \to (\forall n \in Nat)$
$(Pn \to P(Succ\ n)) \to (\forall n \in Nat)Pn$

化简前束范式(Prenex Normal Form)为：

$Ind_{Nat}': \exists(P: Nat \to Set)(\forall n \in Nat)$
$((P0 \to (Pn \to P(Succ\ n))) \to Pn)$

例 1 的自然数类型归纳规则 Ind_{Nat} 应用集值函数(Set-valued Functions)Pn 描述 n 的语义性质，为自然数这种归纳数据类型的递归计算提供了一种简洁的描述方式，特别是在函数式程序语言(如 Haskell)中，Ind_{Nat} 所生成的代码片段具有易读、易写与易理解等良好特点。但 Ind_{Nat} 中$\forall$与$\exists$等高阶对象(Higher-ordered Objects)难以在 Set 中得到合理的解释；同时，更为重要的是，谓词 $P: Nat \to Set$ 的共域是 Set 而非其对象，导致 Set 上的自函子F 无法对P 进行直接计算，即对归纳数据类型 X 的任意实例 x，将折叠操作 $fold$ 应用到以 Px 为载体的F-代数上，无法得到 μF 的归纳规则。因此，我们充分借鉴文献[11]的函子化提升(Lifting Functors)思想，将 Set 上的自函子F 提升为谓词范畴$\mathbb{P}$上的函子$F^{\perp}$。

3 函子提升

定义 3 从谓词范畴$\mathbb{P}$到集合范畴 Set 的函子 $U: \mathbb{P} \to Set$ 称为遗忘函子，将$\mathbb{P}$中的每一个谓词$\langle X,P\rangle$映射为其定义域 X，每一个谓词态射$(f,f^{\sim})$映射为定义域上的函数 f。

定义 4 设 F 为 Set 上的自函子，函子 F 从 Set 到谓词范畴$\mathbb{P}$的提升$F^{\perp}: \mathbb{P} \to \mathbb{P}$，并沿着遗忘函子 U 的方向满足图表交换 $FU=UF^{\perp}$。

由定义 4 知，如果 P 是 X 上的一个谓词，则 $F^{\perp}P$ 是 FX 上的一个谓词，即 $F^{\perp}P: FX \to Set$。对任意谓词态射 $f: P \to P'$，$F^{\perp}f$ 为$F^{\perp}$-代数上的谓词态射：$(FX, F^{\perp}P) \to (FX', F^{\perp}P')$，记为$(q,q^{\sim})$，其中 $q=FX \to FX'=F(Uf)$，$q^{\sim}: \forall y \in FX\ (F^{\perp}Py \to F^{\perp}P'(qy))$，如图 2 所示。

$$\begin{tikzcd}
FX \arrow[rr, "q"] \arrow[dr, "F^{\perp}P"'] & & FX' \arrow[dl, "F^{\perp}P'"] \\
& Set &
\end{tikzcd}$$

图 2 $F^{\perp}$-代数上的谓词态射

记谓词 $P: X \to Set$ 的内涵(Comprehension)为$\{P\}$，$\{P\}=\coprod_{x \in X}Px$ 是一个序对(x,p)，其中 $p \in Px$。为进一步描述 $F^{\perp}$ 的抽象本质，引入内涵函子[11]的概念。

定义 5 内涵函子$\{-\}: \mathbb{P} \to Set$，将任意谓词 P 映射为$\{P\}$，谓词态射$(f,f^{\sim}): P \to P'$映射为$\{(f,f^{\sim})\}: \{P\} \to \{P'\}$，即$\{(f,f^{\sim})\}(x,p)=(fx, P'(fx))$。

内涵函子与遗忘函子间的自然转换 $\pi: \{-\} \to U$ 满足 $\pi_P(x,p)=x$，有 $F^{\perp}P=(F\pi_P)^{-1}$，对于谓词定义域上的函数 $f: X \to X'$，其逆 f^{-1}是谓词 $P': X' \to Set$。尽管自然转换π将 F 与其提升 $F^{\perp}$联系起来，但这种关系并不直观。下面研究以$\{P\}$为载体的 F-代数与以 P 为载体的$F^{\perp}$-代数间直观的互推导关系，进而描述归纳数据类型具有普适意义的归纳规则。

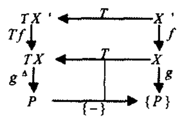
4 归纳数据类型的归纳规则

定义 6 记终对象[12]为 1，函子 $T: Set \to \mathbb{P}$ 将 Set 中每个集合 X 映射为 1，对 Set 上每一个自函子 F 及其提升 $F^{\perp}$，沿着 T 的方向满足图表交换，即 $F^{\perp}(TX)=T(FX)$，称 T 为真值函子。

π_{TX}是终对象为 1 上的自同构 id_1，$F\pi_{TX}$ 也是一个自同构，$(F\pi_{TX})^{-1}$将$\forall x \in FX$ 映射为一个单点集$\{1\}$，即 $F^{\perp}(TX)=(F\pi_{TX})^{-1}=\{1\}=T(FX)$。真值函子 T 是遗忘函子 U 的

右伴随[11]，同时也是内涵函子$\{-\}$的左伴随[13]，即$U \dashv T \dashv \{-\}$。对$\forall X \in Obj\ Set, P \in Obj\ \mathbb{P}$，记$Mor\ \mathbb{P}$中从$TX$到$P$的所有态射构成集合$\mathbb{P}(TX,P)$，$Mor\ Set$中从$X$到$\{P\}$的所有态射构成集合$Set(X,\{P\})$。

设函数$(-)^{\triangle}: Set(X,\{P\}) \to \mathbb{P}(TX,P)$，$(-)^{\triangledown}: \mathbb{P}(TX,P) \to Set(X,\{P\})$，则$(-)^{\triangle}$与$(-)^{\triangledown}$是$Set(X,\{P\})$与$\mathbb{P}(TX,P)$间的自然同构[12]。取$f: X' \to X \in Mor\ Set, g: \mathbb{X} \to \{P\} \in Mor\ Set$，如图3所示。由$(-)^{\triangle}$的自然性得到函数$(-)^{\triangle}$的复合性质，即$(g \circ f)^{\triangle} = g^{\triangle} \circ Tf$。同理，取$f: P \to P' \in Mor\ \mathbb{P}, g: TX \to P \in Mor\ \mathbb{P}$，如图4所示，由$(-)^{\triangledown}$的自然性得到函数$(-)^{\triangledown}$的复合性质，即$(f \circ g)^{\triangledown} = \{f\} \circ g^{\triangledown}$。



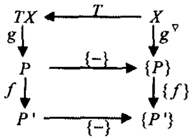


图3　函数$(-)^{\triangle}$的复合性质　　图4　函数$(-)^{\triangledown}$的复合性质

定理3　设Φ为F-代数范畴Alg_F到$F^{\perp}$-代数范畴$Alg_{F^{\perp}}$的函子，即$\Phi: Alg_F \to Alg_{F^{\perp}}$，对任意$F$-代数$h: FX \to X$，则有$\Phi h: F^{\perp}(TX) \to TX$。

证明：对F-代数$h': FX' \to X' \in Obj\ Alg_F$，由定义6知$F^{\perp}(TX') = T(FX')$，令$\Phi h = Th$，图5中$k: X \to X'$为$F$-代数$h: FX \to X$与$h': FX' \to X'$的$F$-代数态射，由图5可知$\Phi k = Tk: TX \to TX'$为$F^{\perp}$-代数$\Phi h: F^{\perp}(TX) \to TX$与$\Phi h': F^{\perp}(TX') \to TX'$的$F^{\perp}$-代数态射。同理可证，若$id_h$为$F$-代数$h: FX \to X$的单位态射，则有$\Phi(id_h) = id_{\Phi h}$；对另一$F$-代数$g$与$h$的复合$g \circ h$，有$\Phi(g \circ h) = \Phi g \circ \Phi h$。故$\Phi$将$F$-代数范畴$Alg_F$中的对象映射为$F^{\perp}$-代数范畴$Alg_{F^{\perp}}$中的对象，即对任意$F$-代数$h: FX \to X$，有$\Phi h: F^{\perp}(TX) \to TX$。证毕。

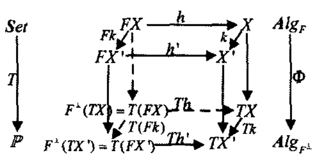


图5　F-代数范畴到$F^{\perp}$-代数范畴的函子Φ

定理3利用真值函子T将Set中的对象与态射映射为$\mathbb{P}$中相应的对象与态射，并基于定义6的图表交换通过函子Φ建立F-代数范畴Alg_F到$F^{\perp}$-代数范畴$Alg_{F^{\perp}}$的联系，下面利用函子Φ的右伴随Ψ研究$Alg_{F^{\perp}}$到Alg_F的联系。

定理4　存在函子Φ的右伴随Ψ（见图6），即$\Phi \dashv \Psi$，对任意$F^{\perp}$-代数$j: F^{\perp}P \to P$，则有$\Psi j: F\{P\} \to \{P\}$。

证明：设$unit$为伴随函子$T \dashv \{-\}$的单位，则$unit^{\triangle}$为其共单位。对$\forall P \in Obj\ \mathbb{P}$，有$unit_P^{\triangle}: T\{P\} \to P$，由定义6知$T(F\{P\}) = F^{\perp}(T\{P\})$，根据函数$(-)^{\triangle}$的定义及复合性质，有$(j \circ F^{\perp} unit_P^{\triangle})^{\triangledown} = \{j\} \circ (F^{\perp} unit_P^{\triangle})^{\triangledown}$。

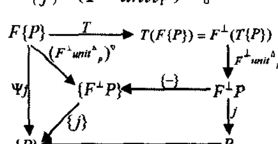


图6　函子Φ的右伴随Ψ

同时，从图6左部的三角交换图表可知$\Psi j = \{j\} \circ (F^{\perp} unit_P^{\triangle})^{\triangledown}$，即$\Psi j = (j \circ F^{\perp} unit_P^{\triangle})^{\triangledown}$。故$\Psi$将$F^{\perp}$-代数中对象$j$映射为$F$-代数中对象$\Psi j$。同理可证，$\Psi$将$F^{\perp}$-代数态射、单位态射及态射复合分别映射为$F$-代数中的态射、单位态射及态射复合，即对任意$F^{\perp}$-代数$j: F^{\perp}P \to P$，有$\Psi j: F\{P\} \to \{P\}$。证毕。

定理4利用函子Φ的右伴随Ψ建立起$F^{\perp}$-代数范畴$Alg_{F^{\perp}}$到F-代数范畴Alg_F的联系，在定理4的证明过程中由定义6可进一步推得$F^{\perp} unit_P^{\triangle}$与$(F^{\perp} unit_P^{\triangle})^{\triangledown}$是同构的，相应于定理3令$\Phi h = Th$，可令$\Psi h = \{h\}$，这为定理4的证明与应用提供了极大的便利。定理4建立了以$\{P\}$为载体的F-代数与以P为载体的$F^{\perp}$-代数间直观的互推导关系，为归纳数据类型提供了一种基于初始代数载体μF的归纳规则的建模方法，如图7所示。

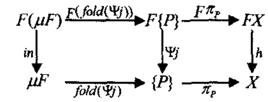


图7　归纳数据类型的归纳规则

对$\forall j: F^{\perp}P \to P \in Obj\ Alg_{F^{\perp}}$，应用定理4有$\Psi j: F\{P\} \to \{P\}$，从图7知$\pi_P \circ fold(\Psi j): \mu F \to X$，进而得到归纳数据类型$X$的通用归纳规则的前束范式：

$$Ind_{Gen}: \forall (F: Set \to Set)(P: X \to Set)$$
$$(j: F^{\perp}P \to P)(x \in \mu F)P((\pi_P \circ fold(\Psi j))x)$$

特殊情况下，若$X = \mu F, h = in$，则由初始F-代数in的初始性可得归纳数据类型μF的通用归纳规则：

$$Ind_{Gen}': \forall (F: Set \to Set)(P: \mu F \to Set)$$
$$(j: F^{\perp}P \to P)(x \in \mu F)P((fold\ j)x)$$

5　相关研究

归纳数据类型是类型理论研究中的一个重要分支，20世纪70年代，Martin-Löf归纳数据类型理论[14]取得了一系列重要研究成果，奠定了归纳数据类型的重要研究基础。但在多态类型系统研究方面仍存在一定的不足，如无法在经典集合论模型中给出合理的解释[15]，经典逻辑的推理在构造逻辑中也不成立[16]等。众多学者的共同努力推动了归纳数据类型理论的发展，如Pitts模型[16]、Effective Topos[17]与Mod模型[18]等在一定程度上解决了上述问题。随后，广义归纳类型成为归纳数据类型研究的重点，并被引入到构造演算的(Calculus of Constructions)[19,20]研究中。文献[21]建立了广义归纳类型的Per范畴模型，并给出其构造演算在Effective Topos子范畴ωSet中的解释[22]。

现有研究大多侧重于应用数理逻辑或代数中的方法来研究归纳数据类型的有限语法构造，如文献[23]基于代数函子分析归纳数据类型的构造，用统一的形式化框架描述语义关系及性质。但对许多归纳数据类型来说，如流、表、树与堆栈等，在程序逻辑和语义计算方面仍存在许多尚未解决的问题，如语义性质与归纳规则的分析与设计等。

相对于传统的归纳数据类型研究方法，本文应用范畴论方法在归纳数据类型的语法构造、语义性质与归纳规则的分析与设计方面具有独特的优势，主要体现在以下3个方面：首先，范畴论简洁的描述性和灵活的扩展性可对归纳数据类型进行精确分析，构造伴随函子等工具设计归纳数据类型的归纳规则，使其描述具有普适性、语义计算自动化；其次是范畴论高度的抽象性在谓词范畴与代数范畴内给出归纳数据类型的语义解释，不再依赖于传统代数或数理逻辑方法的特定约

束，从而增强了归纳数据类型的内聚性，提高了程序语言的独立性；再次是范畴论严密性和一致性的语义计算特点适于在程序语言建模过程中进行严密推理，特别是程序语言建模初期归纳数据类型的分析与构造，极大地减少了软件开发前期出错的情况，为后期确认测试与系统维护等工作提供可靠依据。

结束语 应用范畴论研究归纳数据类型，为程序语言的程序逻辑与语义计算提供了一种简洁、统一的描述方式，同时也增强了程序语言对归纳数据类型语义行为的处理与证明能力。本文分析了谓词范畴与代数范畴的构成与性质，探讨了集合范畴到谓词范畴的函子提升，并利用伴随函子及其伴随性质建立了具有普适意义的归纳数据类型归纳规则。

下一步工作将初步探讨归纳数据类型及归纳规则构成形式系统(Formal Systems)的可靠性、完备性与一致性等元性质，并进一步将我们的研究成果扩展到共归纳数据类型(Coinductive Data Types)及共归纳规则的研究中，应用范畴论的对偶原理(Dual Principles)及适当的分配律(Distributive Laws)深入分析归纳数据类型与共归纳数据类型之间的融合与计算。另外，应用 Fibrations 理论[24-27]将归纳数据类型与共归纳数据类型的研究成果从基本范畴(如集合范畴、谓词范畴和代数范畴等)推广到 2-范畴，深入探讨程序逻辑与语义计算在高阶范畴中的数学解释和范畴语义，也是我们下一步研究的重点。

参考文献

[1] Rutten J J M M, Turi D. Initial algebra and final coalgebra semantics for concurrency[M]// A Decade of Concurrency Reflections and Perspectives. 1993:530-582

[2] Rutten J J M M. Universal coalgebra: a theory of systems[J]. Theoretical Computer Science, 2000, 249(1): 3-80

[3] 苗德成，奚建清，贾连印，等. 一种形式语言代数模型[J]. 华南理工大学学报：自然科学版，2011，39(10)：74-78

Miao D C, Xi J Q, Jia L Y, et al. A Formal Language Algebraic Model[J]. Journal of South China University of Technology (Natural Science Edition), 2011, 39(10): 74-78

[4] Bird R. Introduction to functional programming using Haskell (2nd Edition)[M]. Prentice-Hall, UK, 1998: 15-47

[5] Bird R S, Moor O D. Algebra of Programming [M]. Prentice-Hall, UK, 1997: 107-156

[6] Hutton G. Fold and unfold for program semantics[C]// Proc. of the 3rd ACM SIGPLAN Int. Conf. on Functional Programming, 1998. NewYork: ACM, 1998: 280-288

[7] 苏锦钿，余珊珊. 抽象数据类型的双代数结构及其计算[J]. 计算机研究与发展，2012，49(8)：1787-1803

Su J D, Yu S S. Bialgebraic structures for abstract data types and their computations[J]. Journal of Computer Research and Development, 2012, 49(8): 1787-1803

[8] Barr M, Wells C. Category theory for computing science[M]. NewYork: Prentice-Hall, 1990: 14-16

[9] 屈延文. 形式语义学基础与形式说明[M]. 北京：科学出版社，2010：120-121

Qu Y W. Foundations of formal semantics and formal descriptions[M]. Beijing: Science Press, 2010: 120-121

[10] 陈意云. 计算机科学中的范畴论[M]. 合肥：中国科技大学出版社，1993：36-43

Chen Y Y. Category theory for computer science[M]. Hefei: University of Science and Technology of China Press, 1993: 36-43

[11] Hermida C, Jacobs B. Structural induction and coinduction in a fibrational setting[J]. Information and Computation, 1998, 145(2): 107-152

[12] 贺伟. 范畴论[M]. 北京：科学出版社，2006：15-16

He W. Category theory[M]. Beijing: Science Press, 2006: 15-16

[13] Ghani N, Johann P, Fumex C. Fibrational induction rules for initial algebras[C]// Proceedings, Computer Science Logic. 2010: 336-350

[14] Bengt N, Kent P, Jan M S. Martin-Löf 类型论程序设计导引[M]. 宋方敏，译. 南京：南京大学出版社，2002：1-15

Bengt N, Kent P, Jan M S. Programming in Martin-Löf type theory: an introduction[M]. Song F M ed. Nanjing: Nanjing University Press, 2002: 1-15

[15] Reynolds J. Polymorphism is not set theoretic [M]// Lecture Notes in Computer Science 173. Berlin: Springer-Verlag, 1984: 145-156

[16] Pitts A. Polymorphism is set theoretic, constructively[M]// Lecture Notes in Computer Science 283. Berlin: Springer-Verlag, 1987: 12-39

[17] Hyland M. The effective topos[M]// The L. E. J. Brouwer Centenary Symposium, 1982. North-Holland, 1982: 165-216

[18] Longo G, Moggi E. Constructive natural deduction and its' Modest' interpretation[M]// Semantics of Natural and Computer Languages. Massachusatts: M. I. T. Press, 1990: 96-116

[19] Coquand T, Paulin-Mohring C. Inductively defined types [M]// Lecture Notes in Computer Science 417. Berlin: Springer-Verlag, 1990: 50-66

[20] Dybier P. Inductive sets and families in Martin-Lof type theory and their set theoretical semantics[M]. Cambridge University Press, 1991: 3-27

[21] Fu Y X. Recursive models of general inductive types[J]. Fundamenta Informatica, 1996, 26: 115-131

[22] 傅育熙. 归纳类型的构造集语义[J]. 软件学报，1998，9(3)：236-240

Fu Y X. Constructive semantics of inductive types[J]. Journal of Software, 1998, 9(3): 236-240

[23] Turi D, Plotkin G D. Towards a Mathematical Operational Semantics[C]// Proc. of the 12th Symp. On Logic in Computer Science, 1997. Los Alamitos, CA: IEEE, Computer Science Press, 1997: 280-291

[24] Hermida C A. Fibrations, Logical Predicates and Indeterminates [D]. Edinburg: University of Edinburg, 1993

[25] Ghani N, Johann P, Fumex C. Indexed Induction and Coinduction, Fibrationally[J]. Logical Methods in Computer Science, 2013, 9(3-6): 1-31

[26] Ghani N, Johann P, Fumex C. Generic Fibrational Induction[J]. Logical Methods in Computer Science, 2012, 8(2): 1-27

[27] Fumex C. Induction and Coinduction Schemes in Category Theory[D]. Glasgow: University of Strathclyde, 2012