

形式语言基于 Monads 的语义计算模型

苗德成¹ 奚建清² 苏锦钿³

(韶关学院信息科学与工程学院 韶关 512005)¹ (华南理工大学软件学院 广州 510640)²

(华南理工大学计算机科学与工程学院 广州 510640)³

摘要 传统形式语言的语义建模方法在语义解释与规则描述等语义计算方面存在不足,应用范畴论方法的 Monads 对形式语言的语义计算进行了研究。基于 Monads 构造 Kleisli 范畴,在 Kleisli 范畴的形式化框架内建立语义计算模型,并对该模型进行了应用。与传统语义建模方法相比,所提语义计算模型具有普适性,其语义解释与规则描述的能力更强。

关键词 语义计算, Monads, 伴随函子, 形式语言, Kleisli 范畴

中图法分类号 TP301.2 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.01.038

Semantics Computational Model of Formal Languages Based on Monads

MIAO De-cheng¹ XI Jian-qing² SU Jin-dian³

(School of Information Science and Engineering, Shaoguan University, Shaoguan 512005, China)¹

(School of Software, South China University of Technology, Guangzhou 510640, China)²

(School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China)³

Abstract Traditional semantics modelling methods of formal languages have some drawbacks to interpret semantics and describe rule, and this paper explored semantics computation of formal languages by monads which is categorical method. It firstly constructed Kleisli category by monads, presented a semantics computational model in the formal framework of Kleisli category, and then applied it by example. Compared with traditional semantics modelling methods, the semantics computational model presented by this paper is universal and has more strong abilities of semantics interpreting and rule describing.

Keywords Semantics computing, Monads, Ad joint functor, Formal languages, Kleisli category

1 引言

作为计算理论的核心和基础,形式语言的语义计算是程序设计方法学的重要研究课题^[1]。目前提出的几十种语义计算模型针对不同应用领域建模,侧重于不同抽象层次间具体语义计算技术的研究,有较强的局限性而不具备普适意义^[2-6],导致形式语言的理论研究和工程实践不完善,难以满足软件研发的实际需求。因此,完善和提升已有形式语言的语义计算模型解决问题的能力,研究具有普适性的新方法和新技术,成为形式语言理论新一轮的研究重点。

范畴论方法在数据库系统与程序设计方法学等计算机科学中有较为广泛的应用,为描述数据模型(Data Model)的形式化结构与程序设计规范提供了一种通用的思维方法和理论工具。本文在前人研究工作的基础上,应用范畴论方法的伴随函子与 Monads^[2]等工具,针对形式语言现有语义计算模型在语义解释与规则描述方面存在的问题,提出一种基于 Monads 的语义计算模型,对其进行了应用,并在 Kleisli 范畴的形式化框架内描述了具有普适意义的语义解释与语义规则。

2 语义计算模型研究现状

20 世纪 80 年代中后期, Hagino 开始运用范畴论方法研究形式语言的语义计算^[3],奠定了范畴语义计算模型的研究基础,但其语义解释与规则描述在继承与多态等方面存在一些不足。随后, Poll 对 Hagino 的工作进行了扩展,应用代数与共代数(Co-algebras)的对偶性质研究了子类型与继承^[4]。Nogueira 基于双代数(Bi-algebras)工具建立了语义计算模型并在多态编程中进行了应用^[5]。国内学者苏锦钿应用 λ -双代数探讨了形式语言语法构造与语义行为的关系^[6],并结合共模态(Co-monadic)共递归方法给出了强共归纳(Strong Co-inductive)数据类型的定义及其语义计算^[7],以上成果一定程度上解决了上述问题。

最近, Dorel 提出一个证明程序等价性的 4 规则逻辑演绎系统,对形式语言的语义计算进行了深入的研究,并证明了该系统的可靠性与弱完备性^[8]。Ghani 与 Revell 等学者提出 λ_1 -fibration 的概念,分别用基范畴(Base Category)与全范畴(Total Category)在卡式闭范畴的形式化框架内描述单位消

到稿日期:2015-12-06 返修日期:2016-04-03 本文受国家自然科学基金项目(61103039),广东省自然科学基金项目(S2013010015944),广东省高等学校优秀青年教师培养计划项目(YQ2014155)资助。

苗德成(1979—),男,博士,副教授,主要研究方向为形式语言理论与范畴论方法, E-mail: tony10860@126.com; 奚建清(1962—),男,博士,教授,博士生导师,主要研究方向为数据库与网络计算; 苏锦钿(1980—),男,博士,副教授,主要研究方向为双代数与共代数。

除语义和关系语义,归纳地构造了一种参数化的计量单位 (Unit of Measure)^[9],在 Fibrations 理论的层面对 Kennedy 的语义计算工作^[10]进行了拓展。

现有研究成果主要倾向于形式语言的文法分析与语法构造,如文献^[11]的分类范畴文法虽然较好地解决了复杂文法描述问题,并证明了形式语言的层次结构,但其文法迭代类型的语义计算还需进一步研究;文献^[12]针对结构分析与设计语言 AADL(Architecture Analysis & Design Language)构造了实时规约语言,支持模拟与模型检测,但重写逻辑语义框架在描述多通信模型与对象结构等方面存在局限性,尤其是在实时模型检测时违背自动机理论的判定规则。在形式语言的语义计算方面,当前仍有许多尚未解决的问题,如语义解释与规则描述等,特别是语义规则多以自动生成为主,缺乏坚实的数学基础和精确的形式化描述。

针对以上问题,本文应用范畴论方法,提出了一种基于 Monads 的语义计算模型。本文的组织结构如下:首先,引入范畴间的伴随函子 $F \dashv G$ 与 Monads M 等本文研究所必需的基本概念;其次,由 M 构造 Kleisli 范畴 K ,进而建立语义计算模型 M_K ;再次,对 M_K 进行了应用,并在 Kleisli 范畴 K 内给出具有普适性的语义解释与规则描述;最后,总结全文并指出后续研究工作。

3 伴随函子与 Monads

本文假定读者具备函子、自然转换与图表交换等范畴论基础。若范畴全体对象与态射都构成集合,则该范畴是小范畴 (Small Category)^[13],本文研究对象均基于小范畴。记 $Obj\ C$ 为范畴 C 的对象集, $Mor\ C$ 为范畴 C 的态射集, id 为单位态射, Id 为单位函子。

函子的伴随性质 (Ad joint Properties) 在计算机科学的许多具体应用中广泛存在,为研究不同范畴中不同对象间的联系提供了一种有力的形式化工具。虽然伴随函子的应用较多,但目前尚未形成一个统一的定义。本文从形式语言的语义计算角度引入伴随函子的定义。

定义 1 设 $F:C \rightarrow D$ 与 $G:D \rightarrow C$ 是范畴 C, D 间的一对函子。存在两个自然转换 $\eta: Id_C \rightarrow GF, \epsilon: FG \rightarrow Id_D$, 满足图表交换 $G\epsilon \circ \eta G = Id_G, \epsilon F \circ F\eta = Id_F$ 。称 F 与 G 是一对伴随函子, F 是 G 的左伴随, G 是 F 的右伴随, 记为 $F \dashv G; C \rightarrow D$, 简记为 $F \dashv G$, 其中自然转换 η 为 $F \dashv G$ 的单位, ϵ 为 $F \dashv G$ 的共单位。

由定义 1 知,伴随函子的伴随性质可由单位与共单位确定。由伴随性质可进一步得到 C 上的自函子 $T=GF; C \rightarrow C$, 则单位 $\eta: Id_C \rightarrow T$, 有自然转换 $\mu=G\epsilon F; GFGF \rightarrow GF$, 即 $\mu: T^2 \rightarrow T$ 。以上构造可得到 Monads 这种具有伴随性质的数学结构,它是泛代数 (Universal Algebra) 对伴随概念的一种抽象。

定义 2 范畴 C 上的一个 Monad $M=(T, \eta, \mu)$ 由 3 个要素组成,自函子 $T:C \rightarrow C$ 与两个自然转换 $\eta: Id_C \rightarrow T, \mu: T^2 \rightarrow T$ 满足图表交换。

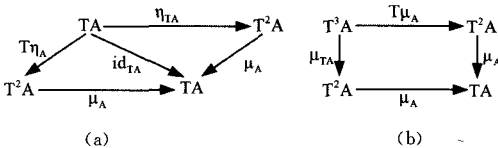


图 1 Monads 结构的图表交换

图 1(b)中的图表交换构造了 T 上满足结合律的二元操作 $\mu: T^2 \rightarrow T$, 而图 1(a)中的图表交换则构造了操作 μ 关于自函子 T 的一个单位元 $\eta: Id_C \rightarrow T$ 。伴随函子 $F \dashv G$ 的伴随性质在一定程度上可由范畴 C 上的 Monad M 确定。

例 1 将 Monad $M=(T, \eta, \mu)$ 应用到带计算副作用 (Side Effects) 的形式语言中, 令 $f: E_A \rightarrow A$ 为表达式求值的语义函数, 其中 E_A 为含数据类型 A 的变量的表达式。定义自函子 $T: A \rightarrow A \times S, S$ 为实际存储位置, 若计算结果存在, 则 S 非空。 $\exists a \in A, \eta_A(a) = a' \times s, \mu_A(a) = a'' \times s', a' = f(a), a'' = f(a'), s, s' \in S$ 。

形式语言的编译器在将源代码转换为目标代码的过程中会进行优化,导致表达式中部分变量的计算顺序发生变化,从而产生副作用。对于有副作用的形式语言,其执行顺序直接影响程序的执行结果。为保证程序运行结果的确定性,部分命令式形式语言 (如 C++) 引入顺序点 (Sequence Point) 的概念缓解副作用问题,但任意两个顺序点之间代码的执行顺序仍是不确定的。例 1 在 Monads 的形式化框架内将形式语言的副作用限定在自函子 T 对变量的作用,并以 S 存储变量的计算结果,其简洁、统一的描述在一定程度上解决了可读性差的问题。

4 Kleisli 范畴与语义计算模型

定义 2 解释了 Monad 的数学性质,但不能直接应用于形式语言的语义计算。由 Monad M 可构造 Kleisli 范畴 K , K 与自由 M -代数的全子范畴是等价的。Kleisli 范畴在理论计算机科学中的应用较为广泛,下面引入 Kleisli 范畴的定义,并在 Kleisli 范畴内抽象地描述形式语言具有普适意义的语义计算。

定义 3 设 $M=(T, \eta, \mu)$ 是范畴 C 上的一个 Monad, 基于 M 构造 Kleisli 范畴 K 。 $Obj\ K = Obj\ C, \forall f: A \rightarrow B \in Mor\ K, \exists f': A \rightarrow TB \in Mor\ C$ 。Kleisli 范畴 K 的复合运算为 $\exists g: B \rightarrow C \in Mor\ K, g \circ f \in Mor\ K, \mu_C \circ Tg' \circ f' \in Mor\ C$ 。Kleisli 范畴 K 中对象 A 上的单位态射 $id_A = \eta_A: A \rightarrow TA \in Mor\ K$ 。

范畴 C 与 Kleisli 范畴 K 之间可构造一对伴随函子 $F \dashv G; C \rightarrow K$ 。令 $GA=TA, FA=A$, 由图 2 知 $Gf = \mu_B \circ Tf'$ 。类似地, 若 $g: A \rightarrow B \in Mor\ C$, 则有 $Fg = Tg \circ \eta_A$ 。图 2 中的图表交换 $T=G \circ F$ 将自函子 T 与伴随函子 $F \dashv G$ 有机融合起来, 这为描述形式语言的语义计算提供了较大便利。

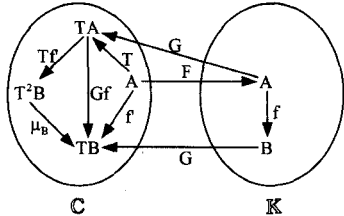


图 2 伴随函子 $F \dashv G$

对于定义 3 中 Kleisli 范畴的任一态射 $f: A \rightarrow B$, 范畴 C 中对应存在一个态射 $f': A \rightarrow TB$ 描述 f 的语义计算。记 f'_C 为 f' 的计算态射, 且 $f'_C: TA \rightarrow TB$ 。由 Monads 与计算态射进一步建立形式语言的语义计算模型 M_K 。

范畴 C 上的语义计算模型 $M_K=(T, \eta, (-)_C)$, 对 $\forall A \in$

$Obj\ K, \eta_A: A \rightarrow TA$ 是一个单态射 (Mono-morphism)。对 $\forall f: A \rightarrow B \in Mor\ K, Tf = (\eta_B \circ f)_C, \mu_A = (id_{TA})_C$, 且有以下等式成立: $(\eta_A)_C = id_{TA}, f'_C \circ \eta_A = f'$ 。对 C 中另一态射 $g': B \rightarrow TC$, 有 $g'_C \circ f'_C = (g'_C \circ f')_C$ 。

M_K 中的单态射 η_A 是一个从数据类型到语义计算的包含态射, 而 f'_C 则是 f 在语义计算层面上的扩展, 即先将 Kleisli 范畴 K 中从数据类型到数据类型的态射 f 映射为范畴 C 中从数据类型到语义计算的态射 f' , 再应用 M_K 将 f' 映射为从语义计算到语义计算的计算态射 f'_C , 描述具体的语义计算效果。

定理 1 以形式语言中的数据类型为对象, 以函数为态射, 构成一个范畴 FL 。

证明: 令 $dom, cod: Mor\ FL \rightarrow Obj\ FL$ 分别为域与共域函数, 下面证明 FL 是一范畴。

1) 匹配条件。设 $\forall A, B, C \in Obj\ FL$, 存在两个态射 $f: A \rightarrow B, g: B \rightarrow C \in Mor\ FL$, 则态射间的复合运算 $g \circ f: A \rightarrow C \in Mor\ FL$, 所以满足范畴的匹配条件, 即 $dom(g \circ f) = dom(f) = A, cod(g \circ f) = cod(g) = C$ 。

2) 结合律条件。取另一态射 $h: C \rightarrow D \in Mor\ FL, h \circ g: B \rightarrow D \in Mor\ FL, h \circ (g \circ f): A \rightarrow D \in Mor\ FL$, 则有态射的结合运算 $(h \circ g) \circ f: A \rightarrow D \in Mor\ FL$, 即满足范畴的结合律条件 $h \circ (g \circ f) = (h \circ g) \circ f$ 。

3) 单位态射存在条件。记 $\exists!$ 为表示唯一存在的摹状词, 对 $\forall A \in Obj\ FL, \exists! id_A \in Mor\ FL$, 域与共域函数对单位态射的作用 $dom(id_A) = cod(id_A) = A$, 且对 $\forall f \in Mor\ FL$, 若 $dom(f) = A$, 则 $f \circ id_A = f$; 若 $cod(f) = A$, 则 $id_A \circ f = f$, 满足范畴的单位态射存在条件。

由范畴的定义^[14]可知 FL 是一范畴。证毕。

定理 1 为形式语言的语义解释与规则描述提供了数学基础, 将定理 1 应用到语义计算模型 M_K , 即以数据类型 A 或数据类型的语义计算 TA 为对象, 函数为态射, $(\eta_A)_C$ 为单位态射, 并定义结合运算 $g'_C \circ f'_C = (g'_C \circ f')_C$, 则形式语言在 M_K 框架内构成 Kleisli 范畴 K 。

例 2 若 FL 是形式语言构成的范畴, 自函子 T 定义为共变幂域 (Covariant Power Set) 函子, 记 $\mathcal{P}(A)$ 为集合 A 的幂集, 1 为终对象, $T(A) = \mathcal{P}(A), \eta_A: A \rightarrow TA$ 。对数据类型 A 中的任一元素 $a, \eta_A(a)$ 为终对象 1 构成的单点集 $\{1\}$, $(\eta_A)_C$ 为终对象的单位态射, 即 $(\eta_A)_C = id_1$, 而 μ_A 为 A 中相应元素构成的并集, 即 $\mu_A = \bigcup_{a \in A} \{a\}$ 。

5 内核小语言及其语义计算

采用 Backus-Naur 范式定义一个小型类 Java 语言, 称之为内核小语言 KSL。下面在 M_K 框架内描述 KSL 的语义解释与语义规则。

5.1 KSL 的语法构造

p : 程序 e : 表达式 s : 语句 dec : 声明
 t : 数据类型 var : 变量 cnt : 常量
 op : 操作 una : 一元操作 bin : 二元操作
 $p ::= \text{begin } del; s \text{ end}$
 $del ::= var; t | cnt; t$
 $t ::= int | bool | char$

$s ::= s_1; s_2 \mid \text{if } e \text{ then } s_1 \text{ else } s_2 \mid fi \mid \text{while } e \text{ do } s \text{ od} \mid var = e \mid skip \mid e$
 $e ::= var \mid cnt \mid ope \mid (e)$
 $op ::= una \mid bin$

为简化问题陈述, 本文约定表达式 e 中仅有一个自由变元, 语法域中加粗部分是 KSL 编译器内部解析的关键字。KSL 的数据类型只有整型 int 、布尔型 $bool$ 与字符型 $char$, 根据需求可适当增加其它数据类型。语句包括顺序、选择、循环、赋值与空 5 种基本操作, 一元操作 una 包括分别针对 int , $bool$ 与 $char$ 的取负、非与后继等, 二元操作 bin 包括加法、析取与合并等。下面在语义计算模型 M_K 的形式化框架内给出 KSL 的语义解释。

5.2 KSL 的语义解释

引入 Kleisli 范畴 K 中初始对象 0 与终对象 1 , 语义蕴涵符号 $\Rightarrow$ 与语法推导关系 $\models$, 并约定 $\models$ 的优先级低于 $\Rightarrow$ 。同时, 为与 KSL 中的赋值操作 $=$ 相区别, 引入符号 $\overset{\Delta}{=}$ 表示语义解释。KSL 中的基本数据类型 $\llbracket int \rrbracket, \llbracket bool \rrbracket, \llbracket char \rrbracket \in Obj\ K$, 并定义数据类型 t 在 M_K 自函子 T 的作用 $\llbracket Tt \rrbracket \overset{\Delta}{=} T\llbracket t \rrbracket$, 且有 $\llbracket t_1 \Rightarrow t_2 \rrbracket \overset{\Delta}{=} [\llbracket t_1 \rrbracket \rightarrow \llbracket t_2 \rrbracket] \in Mor\ K$ 。

$$\llbracket var; t \rrbracket \overset{\Delta}{=} 0 \rightarrow \llbracket t \rrbracket.$$

$$\llbracket cnt; t \rrbracket \overset{\Delta}{=} \llbracket t \rrbracket \rightarrow 1.$$

$$\llbracket (e) \rrbracket \overset{\Delta}{=} \llbracket e \rrbracket.$$

记 $f^{(n)}$ 为 f 的第 n 次计算态射, 当 $n=1$ 时, $f^{(1)}$ 简记为 f_C , 并令 g_1 与 g_2 分别为语句 s_1 与 s_2 在 K 中的对应态射。

$$\llbracket s_1; s_2 \rrbracket \overset{\Delta}{=} (g_2)_C \circ g_1.$$

$$\llbracket \text{if } e \text{ then } s_1 \text{ else } s_2 \mid fi \rrbracket \overset{\Delta}{=} \begin{cases} g_1, & \text{if } \llbracket e \rrbracket \overset{\Delta}{=} \text{true} \\ g_2, & \text{else } \llbracket e \rrbracket \overset{\Delta}{=} \text{false} \end{cases}$$

$$\llbracket \text{while } e \text{ do } s \text{ od} \rrbracket \overset{\Delta}{=} \begin{cases} \llbracket skip \rrbracket, & \text{if } \llbracket e \rrbracket \text{ 首次为 false,} \\ g_C^{(n)} \circ g_C^{(n-1)} \circ \dots \circ g_C \circ g, & \text{else if } \llbracket e \rrbracket \\ \text{连续 } n \text{ 次为 true,} \\ g_C^{(n-1)} \circ g_C^{(n-2)} \circ \dots \circ g_C \circ g, & \text{else } \llbracket e \rrbracket \\ \text{连续 } n-1 \text{ 次为 true, 第 } n \text{ 次为 false} \end{cases}$$

$\llbracket var = e \rrbracket \overset{\Delta}{=} \eta_{t_1}$, 变元 var 与表达式 e 的数据类型满足匹配条件, 即 $var, e: t$ 。

$\llbracket skip \rrbracket \overset{\Delta}{=} \perp$, 用底元素 $\perp$ 表示空语句的不确定性语义计算。

5.3 KSL 的语义规则

对表达式 e 中的自由变元 x , 定义一元操作 $una: t_2 \Rightarrow t_3$ 的语法规则为:

$$Rule_1: x: t_1 \Rightarrow e: t_2 \mid = x: t_1 \Rightarrow una(e): t_3$$

其中, 以 Kleisli 范畴 K 中的态射 g_1 解释语法推导关系 $\models$ 前件的语义 $\llbracket x: t_1 \Rightarrow e: t_2 \rrbracket \overset{\Delta}{=} g_1: \llbracket t_1 \rrbracket \rightarrow T\llbracket t_2 \rrbracket$, 以态射 g_2 解释一元操作语义 $\llbracket una: t_2 \Rightarrow t_3 \rrbracket \overset{\Delta}{=} g_2: \llbracket t_2 \rrbracket \rightarrow T\llbracket t_3 \rrbracket$, 则有 $\llbracket Rule_1 \rrbracket \overset{\Delta}{=} (g_2)_C \circ g_1$ 。

定义二元操作 $bin: t_1 \times t_2 \Rightarrow t_3$ 的语法规则为:

$Rule_2: x; t \Rightarrow e_1; t_1 \times e_2; t_2 \mid = x; t \Rightarrow bin(e_1; t_1 \times e_2; t_2); t_3$
 则前件语义 $\llbracket x; t \Rightarrow e_1; t_1 \times e_2; t_2 \rrbracket \stackrel{\Delta}{=} g_1: \llbracket t_1 \rrbracket \rightarrow T\llbracket t_1 \times t_2 \rrbracket$, 二元
 操作语义为 $\llbracket bin; t_1 \times t_2 \Rightarrow t_3 \rrbracket \stackrel{\Delta}{=} g_2: \llbracket t_1 \times t_2 \rrbracket \rightarrow T\llbracket t_3 \rrbracket$, 则有
 $\llbracket Rule_2 \rrbracket \stackrel{\Delta}{=} (g_2) \circ g_1$.

置换规则为:
 $Rule_3: x; t_1 \Rightarrow e; t_2 \mid = x; t_1 \Rightarrow e[x=y]; Tt_2$
 其中, $e[x=y]$ 表示用另一自由变元 y 置换表达式 e 中自由变
 元 x 的所有出现, 前件语义 $\llbracket x; t_1 \Rightarrow e; t_2 \rrbracket \stackrel{\Delta}{=} g: \llbracket t_1 \rrbracket \rightarrow T\llbracket t_2 \rrbracket$, 置
 换表达式语义 $\llbracket e[x=y]; Tt_2 \rrbracket \stackrel{\Delta}{=} \mu_{\llbracket t_2 \rrbracket}^{-1}: T\llbracket t_2 \rrbracket \rightarrow T^2\llbracket t_2 \rrbracket$, 则有
 $\llbracket Rule_3 \rrbracket \stackrel{\Delta}{=} \mu_{\llbracket t_2 \rrbracket}^{-1} \circ g$, 其中 μ^{-1} 为 μ 的逆运算.

等式规则为:
 $Rule_4: x; t_1 \Rightarrow e_1; t_2, x; t_1 \Rightarrow e_2; t_2 \mid = x; t_1 \Rightarrow e_1 = e_2; t_2$
 $Rule_4$ 推导关系前件中的 $\wedge$ 操作在逻辑上表示与关系,
 同时约定其优先级高于 $\mid =$ 而低于 $\Rightarrow$. 以态射 g_1 解释等式规
 则前件, 操作前面部分的语义 $\llbracket x; t_1 \Rightarrow e_1; t_2 \rrbracket \stackrel{\Delta}{=} g_1: \llbracket t_1 \rrbracket \rightarrow T\llbracket t_2 \rrbracket$, 以态射 g_2 解释等式规则前件, 操作后面部分的语义为
 $\llbracket x; t_1 \Rightarrow e_2; t_2 \rrbracket \stackrel{\Delta}{=} g_2: \llbracket t_1 \rrbracket \rightarrow T\llbracket t_2 \rrbracket$, 则 $\llbracket Rule_4 \rrbracket \stackrel{\Delta}{=} g_1 = g_2$.

表达式存在的规则为:
 $Rule_5: x; t_1 \Rightarrow e; t_2 \mid = x; t_1 \Rightarrow \exists e; t_2, Rule_5$ 中语法推导关
 系的 后件表示自由变元 x 生成的表达式 e 存在, 且数据类型
 为 t_2 . 前件语义 $\llbracket x; t_1 \Rightarrow e; t_2 \rrbracket \stackrel{\Delta}{=} g: \llbracket t_1 \rrbracket \rightarrow T\llbracket t_2 \rrbracket, \exists ! f: \llbracket t_1 \rrbracket \rightarrow$
 $\llbracket t_2 \rrbracket$, 有 $\llbracket Rule_5 \rrbracket \stackrel{\Delta}{=} g = \eta_{\llbracket t_2 \rrbracket} \circ f$, 即态射 g 通过态射 f 由单位态
 射 $\eta_{\llbracket t_2 \rrbracket}$ 唯一分解.

$Rule_5$ 适用于 KSL 中较为复杂的语法现象, 如强制类型
 转换, 即若引入浮点型 float 至 KSL 中, $x; int, e$ 为 $x/3$, 其中/
 是除法运算, 则存在 $e = x/3; float$.

应用文献[15]提出的简单重用、扩张重用与选择重用等
 方法, 由 KSL 生成形式语言 L_i 及其语言族 $\{L_i\}$, 为系统开发
 人员提供便利, 根据个人偏好和实际需求选择适合软件系统
 开发特定阶段的程序语言来解决特定问题.

6 相关工作比较

形式语义学 (Formal Semantics) 中的操作语义、指称语义
 与公理语义是形式语言语义计算的 3 种传统建模方法. 其
 中, 操作语义应用部分函数 (Partial Functions) 将每一个不含
 自由变元的语法项映射为可能的结果值, 基于语法项生成同
 余关系, 其核心在于解决两个语法项在操作等价性上的证明;
 指称语义先建立一个语义模型, 然后在 该模型内给出程序的
 语义解释, 其关键在于证明两个语法项在语义模型中指称同
 一个对象; 而公理语义方法则给出一类可能的语义模型, 其难
 点是证明两个语法项在所有可能模型中映射为同一个对象有
 时可归结为 NP-完全问题.

以上 3 种传统建模方法依赖于操作语义、指称语义和公
 理语义等特定的语义计算环境, 缺乏通用的建模概念, 如序贯
 程序语言 SPL/3^[14] 的指称语义受限于 call-by-reference 的函
 数调用假定, 参数传递仅限于变量而无法扩展为一般的表达
 式, 导致其语义解释不具备普适性; 公理语义方法, 如一阶谓
 词语言 $\mathcal{P}$ ^[16], 语法推导关系与规则描述仅适用于简单的形式

语言类型, 谓词演算转换为程序的语义规则多为自动生成, 缺
 乏精确的形式化描述. 而本文在 KSL 中定义的语法推导关
 系与规则描述普遍适用于命令式、函数式与不确定性等形式
 语言类型. 在建模工具的普适性方面, 本文基于 Monads 的
 语义计算模型 M_K 与传统方法的比较如表 1 所列.

表 1 M_K 与传统方法在普适性上的比较			
M_K	操作语义	指称语义	公理语义
普适性	是	否	否

相对于 λ -演算与幂域 (Power Domain) 理论等其他的语
 义计算建模方法, 本文的 M_K 具有同样的表达能力, 但在语义
 解释与规则描述方面比前者更强, 如表 2 所列. 例如, 对任意
 数据类型 A, B, λ -演算以从 A 到 B 的全函数 (Total Func-
 tions) 解释指数对象类型 (Exponential Object Type) B^A 的语
 义, 即有 $\llbracket B^A \rrbracket \stackrel{\Delta}{=} \{f: A \rightarrow B \mid \forall a \in A, f(a) \in B \text{ 且 } f \text{ 计算终止}\}$,
 难以描述非终止性与不确定性程序的语义行为; 集合论方法
 用完全偏序 (Complete Partial Order) $A \cup \{\perp\}$ 解释 A 的不确
 定性语义, 即 $\llbracket A \rrbracket \stackrel{\Delta}{=} A \cup \{\perp\}$; 而幂域理论应用幂集 $\mathcal{P}(A)$ 解
 释 A 的不确定性语义, 即 $\llbracket A \rrbracket = \mathcal{P}(A)$.

表 2 M_K 与其他方法在表达能力上的比较				
	M_K	λ -演算	集合论	幂域理论
语义解释	强	弱	弱	弱
规则描述	强	弱	弱	弱

本文不拘泥于特定的语义计算环境, 在统一的范畴论框
 架内用 $MonadM_K$ 的自函子 T 解释 A 的语义, 即 $\llbracket A \rrbracket \stackrel{\Delta}{=} TA$,
 其基本要求是程序须构成一个范畴, 定理 1 完成这一证明, 并
 且在 Kleisli 范畴 K 内给出了 KSL 简洁与精确的语义解释与
 规则描述.

结束语 本文应用范畴论方法建立了一种基于 Monads
 的语义计算模型, 为形式语言的语义解释与规则描述提供了一
 种简洁、统一的描述方式, 在 Kleisli 范畴内给出 KSL 具有
 普适意义的语义解释与规则描述. 下一步工作将证明 M_K 与
 KSL 操作语义、指称语义与公理语义等传统语义模型的等价
 性, 从而论证 M_K 的正确性和灵活性; 初步探讨由 KSL 生成
 的程序设计语言 L_i 及其语言族 $\{L_i\}$ 构成形式系统 (Formal
 Systems) 的可靠性、完备性与一致性等元性质, 并进一步将本
 文的研究成果从基本范畴 (如集合范畴、代数范畴等) 推广到
 2-范畴, 深入探讨形式语言的语义计算在高阶范畴中的数学
 结构和范畴语义.

参 考 文 献

[1] CHEN Y Q. Formal Languages and Automata[M]. Beijing: Chi-
na Machine Press, 2008. (in Chinese)
陈有祺. 形式语言与自动机[M]. 北京: 机械工业出版社, 2008.

[2] HE W. Category theory [M]. Beijing: Science Press, 2006. (in
Chinese)
贺伟. 范畴论[M]. 北京: 科学出版社, 2006.

[3] HAGINO T. A Categorical programming language[D]. Edin-
burgh, UK: Laboratory for Foundations of Computer Science,
Dept. of Computer Science, University of Edinburgh, 1987.

- [J]. 软件学报, 2008, 19(8): 2043-2053.
- [3] DING Guo-dong, BAI Shuo, WANG Bin. Local Co-occurrence Based Query Expansion for Information Retrieval[J]. Journal of Chinese Information, 2006, 20(3): 84-91. (in Chinese)
丁国栋, 白硕, 王斌. 一种基于局部共现的查询扩展方法[J]. 中文信息学报, 2006, 20(3): 84-91.
- [4] WU Qin, BAI Yu-zhao, LIANG Yong-zhen. A Local Query Expansion Method Based on Semantic Dictionary[J]. Journal of NanJing University(Nature Sciences), 2014, 50(4): 526-533. (in Chinese)
吴秦, 白玉昭, 梁永祯. 一种基于语义词典的局部查询扩展方法[J]. 南京大学学报(自然科学), 2014, 50(4): 526-533.
- [5] LATIRI C, HADDAD H, HAMROUNI T. Towards an Effective Automatic Query Expansion Process Using an Association Rule Mining Approach[J]. Journal of Intelligent Information Systems, 2012, 39(1): 209-247.
- [6] LIU Cai-hong, QI Rui-hua, LIU Qiang. Efficient Query Expansion Based on Positive and Negative Association[J]. China Science Paper, 2013, 8(1): 51-56. (in Chinese)
刘彩虹, 祁瑞华, 刘强. 一种正负关联规则的快速查询扩展算法[J]. 中国科技论文, 2013, 8(1): 51-56.
- [7] TANNEBAUM W, RAUBER A. Using query logs of USPTO patent examiners for automatic query expansion in patent searching[J]. Information Retrieval, 2014, 17(5/6): 452-470.
- [8] DING Xiao-yuan, GU Chun-hua, WANG Ming-yong. Query Expansion of Local Co-occurrence Based on Query Log[J]. Computer Application and Software, 2013, 30(12): 22-27. (in Chinese)
- 丁晓渊, 顾春华, 王明永. 基于查询日志的局部查询扩展[J]. 计算机应用与软件, 2013, 30(12): 22-27.
- [9] WANG Lu, WANG Guo-chun, GUI Jin-hua, et al. A Semantic Retrieval System Based on Ontology[J]. Journal of Changchuan University of Technology (Natural Science Edition), 2013, 34(6): 726-730. (in Chinese)
王璐, 王国春, 桂金花, 等. 本体语义检索系统[J]. 长春工业大学学报(自然科学版), 2013, 34(6): 726-730.
- [10] MENG Hong-wei, ZHANG Zhi-ping, ZHANG Xiao-dan. Research on Intelligent Information Retrieval Model Based on Domain Ontology[J]. Journal of Intelligence, 2013, 32(9): 180-184. (in Chinese)
孟红伟, 张志平, 张晓丹. 基于领域本体的文献智能检索模型研究[J]. 情报杂志, 2013, 32(9): 180-184.
- [11] CHAUHAN R, GOUDAR R, SHARMA R, et al. Domain Ontology based Semantic Search for Efficient Information Retrieval through Automatic Query Expansion[C]// International Conference on Intelligent Systems and Signal Processing. 2013: 397-402.
- [12] WANG Hong, FAN Hong-jie, LI Jian, et al. Research on the Method of Semantic Query Expansion in Civil Aviation Emergency Domain Ontology[J]. International Journal of Digital Content Technology and its Applications, 2014, 8(5): 128-135.
- [13] WANG Xu-yang, XIAO Bo. Query Expansion Method Based on Ontology and Local Context Analysis[J]. Computer Engineering, 2012, 38(7): 57-60. (in Chinese)
王旭阳, 萧波. 基于本体和局部上下文的查询扩展研究[J]. 计算机工程, 2012, 38(7): 57-60.

(上接第 202 页)

- [4] POLL E. Subtyping and inheritance for categorical data types [C]// Proc. of Theories of Types and Proofs, Kyoto, Japan, RIMS Lecture Notes 1023. 1998: 112-125.
- [5] NOGUEIRA P, MORENO-NAVARRO J. Bialgebra views: a way for polytypic programming to cohabit with data abstract[C]// Proceedings of the ACM SIGPLAN Workshop on Generic Programming, New York, 2008. NY: ACM, 2008: 61-73.
- [6] SU J D, YU S S. Coinductive data types and their applications in programming languages[J]. Computer Science, 2011, 38(11): 114-118. (in Chinese)
苏锦钿, 余珊珊. 程序语言中的共归纳数据类型及其应用[J]. 计算机科学, 2011, 38(11): 114-118.
- [7] SU J D, YU S S. Comonadic corecursions on strong coinductive data types[J]. Journal of South China University of Technology (Natural Science Edition), 2014, 42(1): 128-134. (in Chinese)
苏锦钿, 余珊珊. 强共归纳数据类型上的 Comonadic 共递归[J]. 华南理工大学学报(自然科学版), 2014, 42(1): 128-134.
- [8] DOREL L, VLAD R. Program equivalence by circular reasoning [J]. Formal Aspects of Computing, 2015, 27(4): 701-726.
- [9] GHANI N, REVELL T, ATKEY R, et al. Fibrational units of measure [EB/OL]. [2015-03-21]. <http://personal.cis.strath.ac.uk/neil.ghani/pub.htm>.
- [10] KENNEDY A J. Relational parametricity and units of measure [C]// Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '97). New York: ACM, 1997: 442-455.
- [11] DENIS B, ALEXANDER D, ANNIE F. Categorical grammars with iterated types form a strict hierarchy of k-valued languages [J]. Theoretical Computer Science, 2012, 450(13): 22-30.
- [12] USA; Department of Computer Science, University of Illinois at Urbana-Champaign[R]. Formal semantics and analysis of behavioral AADL models in real-time Maude, 2010.
- [13] BARR M, WELLS C. Category theory for computing science [M]. New York: Prentice-Hall, 1990.
- [14] QU Y W. Formal semantics foundation and formal description (2nd Version)[M]. Beijing: Science Press, 2010. (in Chinese)
屈延文. 形式语义学基础与形式说明(第二版)[M]. 北京: 科学出版社, 2010.
- [15] MIAO D C, XI J Q, JIA L Y, et al. Formal language algebraic model [J]. Journal of South China University of Technology (Natural Science Edition), 2011, 39(10): 74-78. (in Chinese)
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.
- [16] LI W. Mathematical Logic[M]. Beijing: Science Press, 2008. (in Chinese)
李未. 数理逻辑[M]. 北京: 科学出版社, 2008.