第9卷第3期
1998年3月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 9, No. 3
Mar. 1998

归纳类型的构造集语义*

傅育熙

(上海交通大学计算机科学与工程系 上海 200030)

摘要 有归纳类型的马丁洛夫类型理论在经典集合论中有一简单的模型. 构造演算不存在经典集合论模型,但在构造集合论中有模型. 本文刻画了构造演算中归纳类型的构造集语义.

关键词 类型理论,归纳类型,ω-集合.

中图法分类号 TP301

类型理论已有100年的历史. 为了避免悖论,罗素在定义公理集合论时首次引入类型系统. 但在随后的几十年中,类型理论没有多少发展. 计算机科学的诞生使类型理论成为研究热点. 这一理论的特点是用统一的框架处理逻辑和计算,因而可将已有的数理逻辑学中的方法和结论应用于程序理论的研究. 类型理论在逻辑和计算两个方面都有许多未解决的问题,面向对象语言的类型系统就是其中一例.

类型理论的语义学是类型理论的一个重要研究分支. 马丁洛夫类型理论[1]的模型论比较简单. 由于马丁洛夫类型系统中的类型是值谓的(Predicative),故其中的任一类型可解释为集合. 对于非值谓的(Impredicative)类型系统F[2],也称为多态类型系统(Polymorphism),情况要复杂得多. 研究者曾一度试图为F找一个经典集合论模型,但这类模型被证明是不可能的. Reynolds在他那篇著名的文章[3]里证明:无法将非值谓类型解释为经典集合. 随后Pitts撰文[4]指出:Reynolds证明中的关键一步使用了经典逻辑的推理方法,该推理在构造逻辑中不成立. 在此基础上,Pitts给出了系统F在构造集合论中的模型. 具体地说,此模型是某类Grothendieck toposes中的内模型,因为所有Grothendieck toposes均为构造集合论模型,故Pitts模型可看成是多态类型系统在构造集合论中的模型. Hyland的Effective Topos[5]是此类模型之一,事实上,多态类型系统在Effective Topos的一个子范畴中已可解释,即有名的Mod模型.[6] Coquand和Huet对多态类型系统进行了扩充,提出了构造演算(Calculus of Constructions).[7]从数学的角度看,构造演算是一个含有内逻辑的类型系统,该内逻辑实质上就是多态类型系统. 从语用的角度而言,构造演算中的非逻辑类型为构造集合,反映在语义上,构造演算在Effective Topos的一个子范畴ω-Set内有模型. ω-Set有一子范畴Per,构造演算的内逻辑命题即被解释为此子范畴中的对象. 此子范畴与Mod范畴是等价的,故构造演算与多态类型系统在Effective Topos中的解释是一致的.

归纳类型是类型理论中最重要的一个类型. 原则上说,马丁洛夫类型理论中的所有类型都是归纳类型. 以后的工作推广了马丁洛夫归纳类型,引入了广义归纳类型.[8,9]广义归纳类型的元素均是良序的,故具有广义归纳类型的马丁洛夫类型理论在经典集合论中也可被解释. 此解释的数学基础即集合论中的递归定理. 广义归纳类型随后被引入构造演算. Ore[10]首次试图给出具有广义归纳类型的构造演算在ω-Set中的解释,但该解释被发现是错误的.[11]作者在文献[12]中给出了广义归纳类型的Per范畴模型. 本文的目的是在文献[12]的基础上给出具有广义归纳类型的构造演算在ω-Set中的解释. 为此,只需说明如何将广义归纳类型解释为ω-集. 事实上,本文将只说明如何解释W-类型,广义归纳类型的解释可类似地得到.

1 ω-集

本节定义ω-集并概述其基本性质,详细描述可见文献[6,10~14]. 我们假定读者具备范畴论和递归论的基本知识. 在本文中,ω表示自然数集. 任意选定递归函数的一个哥德尔编码系统,对任两自然数m和n,用$m\cdot n$表示将第m个递归函数作用到自然数n上的结果,注意$m\cdot n$可能没有定义. 用记号$m\cdot n\downarrow$表示$m\cdot n$有定义. 我们用克林(Kleene)记号$\Lambda m.f(m)$表示递归函数f的哥德尔编码. 设$\langle_,_\rangle:\omega\times\omega\to\omega$是一有效的配对函数,$(_)_0,(_)_1:\omega\to\omega$为

* 本文研究得到国家自然科学基金资助. 作者傅育熙,1962年生,博士,副教授,主要研究领域为类型理论,语义学,并行理论.
本文通讯联系人:傅育熙,上海200030,上海交通大学计算机科学与工程系
本文1997-03-26收到原稿,1997-04-26收到修改稿

相应的第1和第2分量函数,即对所有 n,有 $n_0,n_1\in\omega$ 和 $\langle n_0,n_1\rangle=n$,$\langle n_0,n_1\rangle_0=n_0$ 和 $\langle n_0,n_1\rangle_1=n_1$.

定义. 一个 ω-集为一有序对 $(X,\rhd_X)$,其中集合 X 为该 ω-集的基集,二元关系 $\rhd_X\subseteq\omega\times X$ 是其实现关系,满足以下条件:对任一 $x\in X$,存在 $n\in\omega$ 使得 $n\rhd_X x$. 从 $(X,\rhd_X)$ 到 $(Y,\rhd_Y)$ 的态射为可实现函数 $f:X\to Y$. 这里,函数 f 是可实现的,当且仅当存在某一 $n\in\omega$,使得若 $p\rhd_X x$ 则 $n\cdot p\rhd_Y f(x)$. 所有 ω-集及其间的态射构成范畴 ω-Set.

ω-集由其实现关系完全决定. 若 A 为 ω-集,用 $|A|$ 表示它的基集,$\rhd_A$ 表示它的实现关系. 范畴 ω-Set 是局部笛卡尔闭的,证明见文献[5]. 现将该范畴中的一些基本构造描述如下:

- $(\varnothing,\varnothing)$ 是初始对象,这里 $\varnothing$ 表示空集. 显然从 $(\varnothing,\varnothing)$ 到任意 ω-集有唯一一个态射.
- 若 $A=(|A|,\rhd_A)$ 和 $B=(|B|,\rhd_B)$ 是 ω-集,则其和 $A+B$ 的实现关系定义如下:

$$\rhd_{A+B}\equiv\{\langle m,\langle n,x\rangle\rangle\mid(m)_0=n=0\wedge(m)_1\rhd_A x\vee(m)_0=n=1\wedge(m)_1\rhd_B x\}$$

ω-集 $(\{\varnothing\},\omega\times\{\varnothing\})$ 是 ω-Set 的终极对象. 显然,从任意 ω-集到 $(\{\varnothing\},\omega\times\{\varnothing\})$ 有唯一一个态射.

- 若 $A=(|A|,\rhd_A)$ 和 $B=(|B|,\rhd_B)$ 是 ω-集,则其积 $A\times B$ 的实现关系定义为:

$$\rhd_{A\times B}\equiv\{\langle m,\langle a,b\rangle\rangle\mid(m)_0\rhd_A a\wedge(m)_1\rhd_B b\}$$

- ω-集 $A=(|A|,>_X)$ 和 $B=(|B|,\rhd_B)$ 的幂 B^A 的实现关系定义为:

$$\rhd_{B^A}\equiv\{\langle m,f\rangle\ \ f:|A|\to|B|\wedge\forall n,a.\,(n\rhd_A a\Rightarrow m.n\rhd_B f(a))\}$$

- 自然数对象 N 的实现关系定义为:$N\equiv\{\langle n,n\rangle\mid n\in\omega\}$.

以上只枚举了 ω-Set 中的一些简单构造,它的范畴特性在 Hyland 的文章[5]中有详细讨论.

2 构造演算和归纳类型

构造演算有许多变种,本节给出的变种基于如下的认识:所谓构造演算即在马丁洛夫理论中加入一内逻辑. 该逻辑可看成是 Church 的高阶逻辑[15]在类型理论中的对应物. 构造演算通常由两部分组成:外围语言定义语言的基本框架与逻辑核定义内逻辑.

上下文 $\Rightarrow()$ valid

若 $\Gamma\Rightarrow A$ type, x 不是 Γ 中的变量 则 $\Gamma,x:A$ valid

假设 若 $\Gamma,x:A,\triangle$ valid 则 $\Gamma,x:A,\triangle\Rightarrow x:A$

传递 若 $\Gamma\Rightarrow A$ type, $\Gamma\Rightarrow a:A$, $\Gamma\Rightarrow A=B$ 则 $\Gamma\Rightarrow a:B$

空类型 若 Γ valid 则 $\Gamma\Rightarrow 0$ type

若 $\Gamma\Rightarrow r:0$, $\Gamma\Rightarrow S$ type 则 $\Gamma\Rightarrow ?_S(r):S$

单元类型 若 Γ valid 则 $\Gamma\Rightarrow 1$ type

若 Γ valid 则 $\Gamma\Rightarrow{}^*:1$

若 $\Gamma,x:1\Rightarrow C$ type, $\Gamma\Rightarrow c:C[^*]$ 则 $\Gamma\Rightarrow D[c]:\Pi x:1.C$

和类型 若 $\Gamma\Rightarrow A$ type, $\Gamma\Rightarrow B$ type 则 $\Gamma\Rightarrow A+B$ type

若 $\Gamma\Rightarrow B$ type, $\Gamma\Rightarrow a:A$ 则 $\Gamma\Rightarrow\mathrm{inl}[a]:A+B$

若 $\Gamma\Rightarrow A$ type, $\Gamma\Rightarrow b:B$ 则 $\Gamma\Rightarrow\mathrm{inr}[b]:A+B$

若 $\Gamma,z:A+B\Rightarrow C$ type, $\Gamma\Rightarrow d:\Pi x:A.C[\mathrm{inl}[x]]$, $\Gamma\Rightarrow e:\Pi x:B.C[\mathrm{inr}[y]]$ 则 $\Gamma\Rightarrow\mathrm{case}[d,e]:\Pi z:A+B.C$

广义积类型 若 $\Gamma\Rightarrow A$ type, $\Gamma,x:A\Rightarrow B$ type 则 $\Gamma\Rightarrow\Pi x:A.B$

若 $\Gamma,x:A\Rightarrow b:B$ 则 $\Gamma\Rightarrow\lambda x:A.b:\Pi x:A.B$

若 $\Gamma\Rightarrow f:\Pi x:A.B$, $\Gamma\Rightarrow a:A$ 则 $\Gamma\Rightarrow fa:B[a]$

广义和类型 若 $\Gamma\Rightarrow A$ type, $\Gamma,x:A\Rightarrow B$ type 则 $\Gamma\Rightarrow\Sigma x:A.B$

若 $\Gamma\Rightarrow a:A$, $\Gamma\Rightarrow b:B[a]$ 则 $\Gamma\Rightarrow\langle a,b\rangle:\Sigma x:A.B$

若 $\Gamma\to c:\Sigma x:A.B$ 则 $\pi_0[c]:A$

若 $\Gamma\to c:\Sigma x:A.B$ 则 $\pi_1[c]:B[a]$

自然数类型 若 Γ valid 则 $\Gamma\Rightarrow N$ type

若 Γ valid 则 $\Gamma\Rightarrow 0:N$

若 $\Gamma\Rightarrow n:N$ 则 $\Gamma\Rightarrow\mathrm{succ}[n]:N$

若 $\Gamma,z:N\Rightarrow C$ type, $\Gamma\Rightarrow c:C[0]$, $\Gamma\Rightarrow e:\Pi x:N.\Pi y:C[x].C[\mathrm{succ}[x]]$ 则 $\Gamma\Rightarrow\mathrm{rec}[d,e]:\Pi z:N..C$

归纳类型 若 $\Gamma\Rightarrow A$ type，$\Gamma,x:A\Rightarrow B$ type 则 $\Gamma\Rightarrow Wx:A.B$

若 $\Gamma\Rightarrow a:A$，$\Gamma\Rightarrow b:B[a]\rightarrow Wx:A.B$ 则 $\Gamma\Rightarrow \mathrm{sup}[a,b]:Wx:A.B$

若 $\Gamma,z:Wx:A.B\Rightarrow C$ type，

$\Gamma\Rightarrow d:\Pi x:A.\Pi y:(B\rightarrow Wx:A.B).\Pi h:(\Pi b:B.C[y(b)]).C[\mathrm{sup}[a,b]]$

则 $\Gamma\rightarrow T[d]:\Pi z:(Wx:A.B).C$

若 $\Gamma,z:Wx:A.B\Rightarrow C$ type，

$\Gamma\Rightarrow d:\Pi x:A.\Pi y:(B\rightarrow Wx:A.B).\Pi h:(\Pi b:B.C[y(b)]).C[\mathrm{sup}[a,b]]$，

$\Gamma\Rightarrow a:A$，$\Gamma\Rightarrow b:B[a]\rightarrow Wx:A.B$ 则 $\Gamma\Rightarrow T[d](\mathrm{sup}[a,b])=d(a,b,\lambda x:B.T[d](b(x))):C[\mathrm{sup}[a,b]]$

上述有关 W-类型的第3和第4条规则分别为归纳和等式规则. W-类型是标准的归纳类型，广义归纳类型是对 W-类型的推广. 以上这些规则定义了外围语言，下面几条规则定义构造演算的逻辑核.

命题 若 Γ valid 则 $\Gamma\Rightarrow$Prop type

若 $\Gamma\Rightarrow P$:Prop 则 $\Gamma\Rightarrow P$ type

若 $\Gamma,x:A\Rightarrow P$:Prop 则 $\Gamma\Rightarrow\forall x:A.P$:Prop

若 $\Gamma,x:A\Rightarrow P$:Prop,$\Gamma,x:A\Rightarrow p:P$ 则 $\Gamma\Rightarrow\wedge x:A.p:\forall x:A.P$

若 $\Gamma\Rightarrow a:A,\Gamma\Rightarrow p:\forall x:A.P$ 则 $\Gamma\Rightarrow p(a):P[a]$

3 类型的构造集语义

范畴 ω-Set 是一个很好的计算模型. 下面总结 ω-集作为指称语义模型时常用的一些构造. 设 Γ 为一个 ω-集，A，A'：$|\Gamma|\rightarrow\omega$-Set 为两映射. 可定义 ω-集 $\sigma(\Gamma,A)$，$\pi(\Gamma,A)$ 及映射 $A+A',A\times A'$：$|\Gamma\rightarrow\omega$-Set. 若 B：$\sigma(\Gamma,A)|\rightarrow\omega$-Set 为另一映射，可定义另外两映射 $\sigma_\Gamma(A,B)$，$\pi_\Gamma(A,B)$：$|\Gamma|\rightarrow\omega$-Set. 注意，在定义映射 f：$|\Gamma|\rightarrow\omega$-Set 时，只需说明对任一 $\alpha\in|\Gamma|$如何定义 $f(\alpha)$. 这些构造如下：

- $\sigma(\Gamma,A)$. 基集为$\{(\gamma,a)|\gamma\in|\Gamma|\wedge a\in|A(\gamma)|\}$；实现关系为$\langle m,n\rangle\rhd_{\sigma(\Gamma,A)}(\gamma,a)\Leftrightarrow m\rhd_\Gamma\gamma\wedge n\rhd_{A(\gamma)}a$.
- $\sigma_\Gamma(A,B)(\alpha)$. 基集为$\{(a,b)|a\in|A(\alpha)|\wedge b\in|E(\alpha,a)|\}$；实现关系为$\langle m,n\rangle\rhd_{\sigma\Gamma(A,B)(\alpha)}(a,b)\Leftrightarrow m\rhd_{A(\alpha)}a\wedge n\rhd_{B(\alpha,a)}b$.
- $\pi(\Gamma,A)$. 基集为$\{f:|\Gamma|\rightarrow\bigcup_{\gamma\in|\Gamma|}|A(\gamma)||\forall\gamma\in|\Gamma|.f(\gamma)\in|A(\gamma)|\wedge\exists n\in\omega.n\rhd_{\pi(\Gamma,A)}f\}$；实现关系为 $n>_{\pi(\Gamma,A)}f\Leftrightarrow\forall\gamma\in|\Gamma|\forall p\in\omega.p\rhd\Gamma\gamma\Rightarrow n.p\rhd_{A(\gamma)}f(\gamma)$.
- $\pi_\Gamma(A,B)(\alpha)$. 基集为$\{f:|A(\alpha)|\rightarrow\bigcup_{a\in|A(\alpha)|}|B(\alpha,a)||\forall a\in|A(\alpha)|.f(a)\in|B(\alpha,a)|\wedge\exists n\in\omega.n\rhd_{\pi\Gamma(A,B)(\alpha)}f\}$；实现关系为 $n\rhd_{\pi\Gamma(A,B)(\alpha)}f\Leftrightarrow\forall a\in|A(\alpha)|.\forall P\in\omega.p\rhd_{A(\alpha)}a\Rightarrow n.p\rhd_{B(\alpha,a)}f(\alpha)$.
- $(A+_\Gamma B)(\alpha)$. 基集为$\{(0,a)|a\in|A(\alpha)|\}\cup\{(1,b)|b\in|B(\alpha)|\}$；实现关系为$\langle j,n\rangle\rhd_{(A+\Gamma B)(\alpha)}(j,c)\Leftrightarrow(i=j=0\wedge n\rhd_{A(\alpha)}c)\vee(i=j=1\wedge n\rhd_{B(\alpha)}c)$.

在 ω-集范畴中解释类型系统时，上下文 Γ 被解释为 ω-集【Γ】，判断 $\Gamma\Rightarrow A$:Type 被解释为一从$|\Gamma|$到 ω-Set 中对象的映射，这里$|\Gamma|$即$|$【Γ】$|$. 空上下文解释为 ω-Set 中的初始对象$(\{\varnothing\},\omega\times\{\varnothing\})$，非空上下文 $\Gamma,x:A$ 解释为 σ(【Γ】,【A】)，这里【A】：$|\Gamma|\rightarrow\omega$-Set 是 $\Gamma\Rightarrow A$:Type 的解释. 下面给出构造演算中除 W-类型和逻辑核之外的类型在 ω-Set 中的解释.

$$[\![0]\!]\equiv(\varnothing,\varnothing)$$

$$[\![1]\!]\equiv(\{\varnothing\},\omega\times\{\varnothing\})$$

$$[\![A+B]\!]\equiv[\![A]\!]+_{[\![\Gamma]\!]}[\![B]\!]$$

$$[\![\Pi x:A.B]\!]\equiv\pi_{[\![\Gamma]\!]}([\![A]\!],[\![B]\!])$$

$$[\![\Sigma x:A.B]\!]\equiv\sigma_{[\![\Gamma]\!]}([\![A]\!],[\![B]\!])$$

$$[\![N]\!]\equiv(\omega,\{\langle n,n\rangle|n\in\omega\})$$

项 $\Gamma\Rightarrow a:A$ 解释为一态射【a】:【Γ】$\rightarrow\sigma$(【Γ】,【A】)，该态射满足条件:【a】;$\pi_1=Id_{[\![\Gamma]\!]}$，其中 $\pi_1:\sigma$(【Γ】,【A】)$\rightarrow$【Γ】为第一分量函数. 由于篇幅限制，我们只讨论这些. 有兴趣的读者可参阅文献[13]. W-类型的语义在下节讨论.

4 W-类型的 ω-集语义

设 A:Type 和 $x:A\Rightarrow B$:Type.【A】为 A 的解释，将【A】的基集简记为$|A|$. 类型 B 的解释为映射【B】：$|A|\omega$-Set. 首

先构造一个 ω-集序列如下：

- $\omega^0 = (\varnothing, \varnothing)$.
- $\omega^{\alpha+1} \equiv (|\omega^{\alpha+1}|, \rhd_{\alpha+1})$，此处 $\rhd_{\alpha+1} \equiv \{\langle\langle m,n\rangle,\langle x,y\rangle\rangle \mid m [\![A]\!] x \wedge n\{[\![B]\!](x) \to \omega^\alpha\} y\}$ 且 $|\omega^{\alpha+1}|$ 由 $\rhd_{\alpha+1}$ 的第 2 分量组成.
- $\omega^\gamma = (\bigcup_{\alpha\in\gamma} |\omega_\alpha|, \bigcup_{\alpha\in\gamma} \rhd_\alpha)$，这里 γ 为极限序数.

显然，$\omega^0 \subset \omega^1 \subset \ldots \subset \omega^\alpha \subset \ldots$. 基于简单的基数限制[9]，上述序列必须稳定于某一序数 θ. 将 $[\![W\ x:A.B]\!]$ 定义为 ω-集 ω^θ. 对于 $a:A$ 和 $b:B[a] \to W\ x:A.B$，有 $[\![a]\!] \in |A|$ 和 $[\![b]\!] \in |[\![B]\!]([\![a]\!]) \to \omega^\theta|$. 故 $[\![\sup[a,b]]\!] \equiv \langle [\![a]\!], [\![b]\!] \rangle \in |\omega^{\theta+1}| = |\omega^\theta|$.

我们还需解释 $W\ x:A.B$ 的归纳规则和等价规则. 根据 $W\ x:A.B$ 中元素的构造过程，可定义一系列函数，所有这些函数可由一个统一的递归函数实现. 考虑下面的递归函数：

$$e \equiv \Lambda f \Lambda m.\, n_d((m)_0, (m)_1, \Lambda x.\, f((m)_1(x)))$$

其中 n_d 实现 $[\![d]\!]$，d 是 W-归纳规则中出现的那个 d. 根据递归定理，存在递归函数 $R(e)$，使

$$R(e) \cong \Lambda m.\, n_d((m)_0, (m)_1, \Lambda x.\, R(e)((m)_1(x)))$$

这里 $f \cong g$ 定义为 $\forall n.\, (f.n\downarrow \Leftrightarrow g.n\downarrow) \wedge (f.n\downarrow \Rightarrow f.n = g.n)$. 下面定义所需的函数序列.

- $F_0 \equiv (\varnothing, \varnothing)$. F_0 是定义域为 $|\omega^0|$ 的函数图. 显然，此函数为 $R(e)$ 所实现.
- $F_{\alpha+1} = F_\alpha \cup \{\langle\langle x,y\rangle, [\![d]\!](x,y,F_\alpha \circ y)\rangle \mid \langle x,y\rangle \in |\omega^{\alpha+1}| - |\omega^\alpha|\}$. 根据 $\omega^{\alpha+1}$ 的定义，$y \in |[\![B]\!](x) \to \omega^\alpha|$. F_α 是定义域为 ω^α 的函数，故 F_α 可由 $R(e)$ 所实现. 显然 $F_\alpha \circ y$ 是良定义的，因而 $F_{\alpha+1}$ 也是良定义的，并为定义域是 $\omega^{\alpha+1}$ 的函数图. 设 $\langle n_x, n_y\rangle \rhd \omega^{\alpha+1} \langle x,y\rangle$，则

$$R(e) \cdot \langle n_x, n_y\rangle \cong n_d(n_x, n_y, \Lambda x.\, R(e)(n_y(x)))$$

从 $n \rhd_{[\![B]\!](x)} b$ 可得 $n_y.n \rhd \omega^\alpha y(b)$. 根据归纳法假定，$R(e)$ 实现 F_α. 所以 $R(e) \cdot (n_y.n)$ 有定义，而 $\Lambda x.\, R(e)(n_y(x))$ 实现 $F_\alpha \circ y$. 所以 $n_d(n_x, n_y, \Lambda x.\, R(e)(n_y(x)))$ 有定义且实现 $[\![d]\!](x,y,F_\alpha \circ y)$. 因而 $R(e) \cdot \langle n_x, n_y\rangle$ 有定义且实现 $F_{\alpha+1}(\langle x,y\rangle)$. 结论：$F_{\alpha+1}$ 可由 $R(e)$ 实现.

- 设 γ 为极限序数，定义 $F_\gamma \equiv \bigcup_{\alpha\in\gamma} F_\alpha$. 由于对每一个 $\alpha \in \gamma$，$R(e)$ 实现 F_α，故 $R(e)$ 也实现 F_γ.

根据构造，$F_0 \subset F_1 \subset F_2 \ldots F_\theta = F_{\theta+1}$. 设 $\langle x,y\rangle \in |\omega^\theta|$ 并假定 α 是满足 $\langle x,y\rangle \in |\omega^\alpha|$ 的最小序数. 显然，α 是一后继序数，所以

$$\begin{aligned} F_\theta(\langle x,y\rangle) &= F_\alpha(\langle x,y\rangle) \\ &= [\![d]\!](x,y,F_{\alpha-1} \circ y) \\ &= [\![d]\!](x,y,F_\theta \circ y) \end{aligned}$$

这些等式表明，若将 $T[d]$ 解释为 F_θ，则等式规则成立.

在上述解释中，未考虑上下文. 由于递归定理的有效性，故在考虑上下文的情况下，上述构造也成立. 本节给出了 W-类型的 ω-集语义，广义归纳类型的 ω-集语义也可类似地给出.

5 构造演算的 ω-模型

ω-Set 有一个满的子范畴，即所有朴素集(Modest Set)构成的范畴 Mod. 一个朴素集是一个 ω-集 $(X, \rhd_X)$，其中 $\rhd_X$ 是一个映上函数 $f:\omega \to X$ 的图. Mod 与另外一个范畴等价.

定义. 一个 ω 上的 per A 是 ω 上的一个对称传递二元关系. $|A|$ 表示 $\{n \in \omega \mid nAn\}$. 从一个 per A 到另一个 per B 的态射 f 是满足 $\forall m, m' \in \omega.\, mAm' \Rightarrow n.mBn.m'$ 的函数 $f:|A| \to |B|$.

设 Per 是所有 ω 上的 per 构成的范畴. 易证 Per 和 Mod 等价，Per 的范畴性质比 ω-Set 更好[5]，是许多构造逻辑的模型，下面的定理便是一例.[6]

定理. Mod 和 Per 是多态类型系统的模型.

尽管 Per 和 Mod 是等价的，但它们的大小完全不一样. Mod 的所有对象构成一个类，而 Per 的所有对象构成一个集合. 不仅如此，Per 还是 ω-Set 的一个内范畴，且此内范畴在 Effective Topos 中是内完全的，它是我们在 topos 中找到的第 1 个具有此性质的内范畴. 正是这种非常强的性质使我们有[6]：

定理. Per 是构造演算内逻辑的模型.

因此推出本文的主要结论.

定理. ω-Set 是具有归纳类型的构造演算的模型.

证明：ω-Set 是构造演算的模型.[13] 上节已指出如何在 ω-Set 中解释归纳类型. 证毕

6 结 论

本文在构造演算的 ω-集模型中刻画了归纳类型的语义.本文工作的意义并不局限于类型理论,它对逻辑框架中编码的语义[16]及 CZF 的模型论都有应用价值.

参考文献

1 Nordstrom B, Peterson K, Smith J. Programming in Martin-Lof's type theory—an introduction. Vol. 7 of International Series of Monographs on Computer Science, London: Oxford University Press, 1990

2 Girard J. The system F of variable types, fifteen years later. Theoretical Computer Science, 1987,45:1~102

3 Reynolds J. Polymorphism is not set theoretic. In: Kahn G *et al* eds. Semantics of Data Types, Lecture Notes in Computer Science 173. Berlin: Springer-Verlag, 1984. 145~156

4 Pitts A. Polymorphism is set theoretic, constructively. In: Pitt D *et al* eds., Category Theory and Computer Science, Lecture Notes in Computer Science 283. Berlin: Springer-Verlag, 1987. 12~39

5 Hyland M. The effective topos. In: Troelstra A, van Dalen D eds, The L. E. J. Brouwer Centenary Symposium, North-Holland, 1982. 165~216

6 Longo G, Moggi E. Constructive natural deduction and its 'Modest' interpretation. In: Semantics of Natural and Computer Languages. Massachusatts: M. I. T. Press, 1990

7 Coquand T, Huet G. The calculus of constructions. Information and Computation, 1988,76(2):95~120

8 Coquand T, Paulin-Mohring C. Inductively defined types. In: COLOG'88, Lecture Notes in Computer Science 417. Berlin: Springer-Verlag, 1990. 50~66

9 Dybier P. Inductive sets and families in Martin-Lof type theory and their set theoretical semantics. In: Huet G, Plotkin G eds. Proceedings of the First Workshop on Logical Frameworks. Cambridge: Cambridge University Press, 1991

10 Ore C. The extended calculus of constructions (ECC) with inductive types. Information and Computation, 1992,99:231~264

11 Fu Y. Some semantic issues in type theory[Ph. D. Thesis]. Department of Computer Science, University of Manchester, 1992

12 Fu Y. Recursive models of general inductive types. Fundamenta Informatica, 1996,26:115~131

13 Luo Z. ECC an extended calculus of construction. In: Proceedings of the 4th Symposium on Logic in Computer Science. los Alamitos, Califonia: IEEE Computer Science Press, 1989

14 Fu Y. Understanding inductive types in constructions. Technical Report, Department of Computer Science, University of Manchester, 1992

15 Church A. A formulation of the simple theory of types. Journal of Symbolic Logic, 1940,5(1):56~68

16 Fu Y. Categorical properties of logical frameworks. Mathematical Structures in Computer Science, 1997,7(1):1~47

Constructive Semantics of Inductive Types

FU Yu-xi

(Department of Computer Science and Engineering Shanghai Jiaotong University Shanghai 200030)

Abstract Inductive types in Martin-Lof type theory have simple interpretations in classical set theory. The calculus of constructions however does not enjoy such a model. This paper shows how inductive types in the calculus of constructions can be interpreted in the category of omega sets.

Key words Type theory, inductive types, ω-set.

Class number TP301