第一章 PCF语言模型(二)

BY ALAN

1 前言

前面我们简单的介绍了λ演算, 在这个形式系统除语法之外, 还有三个主要部分。它们分别是**公理语义、操作语义和指称语义**。

2 公理语义

公理语义是**推导表达式之间等式的形式系统**,在等式公理系统, 存在下面两点:

- 存在一个对约束变量改名的公理
- 一个使函数调用(application)与函数作为值传递的公理

2.1 详细阐述

为了表示上面的两个公理, 我们使用$[N/x]M$这种记法来表示在M中将表达式N代入x的结果, 注意这种代入的结果不可以将原来自由的变量改成约束的。

但是我们可以将M中的约束变量全部改名, 以保证它们与N中所有自由变量不同名。然后再以N取代所有x的出现。例如$(\lambda x.\lambda y.x)\,(\lambda x.y)$ 首先第一件事情是将$\lambda x.\lambda y.x$ 中y改名, 然后将$\lambda x.y$代入表达式替换掉表达式中的x。于是可以写成下面这样:

$$\lambda x{:}\,\sigma.M = \lambda y.\sigma.[N/x]M, y在M中是约束的$$

由于λ项$\lambda x{:}\,\sigma.M$定义了以x为实参、以M为值的函数, 我们就可以计算以N代入x在实参为N的值, 例如将函数$\lambda x{:}\,\mathrm{nat}.x+5$作用到实参3上面的结果。

$$(\lambda x{:}\,\mathrm{nat}.x+5)\,3 = [3/x]\,(x+5) = 3+5$$

更一般地, 我们将上面的东西成为β等价:

$$(\lambda x{:}\,\sigma.M)\,N = [N/x]M \quad \beta_{(\mathrm{eq})}$$

β等价告诉我们, **我们可以通过将函数体中形参代入实参来估值一个函数作用**。除了这些公理和少量其他的公理, 等式证明系统还包含一些规则, 比如对称$(a=b, b=a)$, 传递$(a=b, b=c, a=c)$, 还有一个同余规则, "相等的函数作用在相等的实参上产生相等的值", 它还可以形式化的写成:

$$\frac{M_1 = M_2, N_1 = N_2}{M_1 N_1 = M_2 N_2}$$

3 操作语义

λ表达式的规约规则提供了等式推理的一种有向形式。直观地说, 基本的规约规则描述了单个计算步骤，它们可被不断计算表达式(如果该表达式不存在最简形式的话, 则计算不会结束)。

这种符号的"估值过程"或"解释器"能给出λ演算的计算特性。尽管β等价是一个等式, **但是通常我们将它从左向右读成一个化简规则**。例如等式$(\lambda x{:}\,\mathrm{nat}.x+5)\,3 = 3+5$让我们能将函数调用简化成$3+5$。利用加法的性质, 还可以简化成8。

由于规约是不对称的, 所以我们不使用等号而改为箭头$\rightarrow$ 而双箭头$\twoheadrightarrow$则可以用于表示零个或多个规约步骤。

中心的规约规则β等价的有向形式, 成为β规约, 写成

$$(\lambda x{:}\,\sigma.M)\,N \rightarrow [N/x]M$$

注意规约只能发生于α等价之下的，β规约所得到的项可能会因为新的约束变量名的任意选择而变得不同，但是也仅仅只是变量名的不同，但是两者依然是α等价的。**代入中α转化与变量的改名都是与静态作用域的部分。**

整个规约系统结合β和其它一些基本的一步规约，这些规约依据的规则让我们可以对一个项中部分进行计算。

选择一个**不同的地方来进行归约**会得到相同的结果:

$(\lambda x{:}\,\sigma.M)\,((\lambda y{:}\,\tau.N)\,P) \twoheadrightarrow (\lambda x{:}\,\sigma.M)\,([P/y]N) \twoheadrightarrow [([P/y]N)/x]M$

$(\lambda x{:}\,\sigma.M)\,((\lambda y{:}\,\tau.N)\,P) \twoheadrightarrow (\lambda x{:}\,\sigma.M)\,([((\lambda y{:}\,\tau.N)\,P)/x]M) \twoheadrightarrow [([P/y]N)/x]M$

此外存在两个在纯类型化λ项上著名的规约性质，它们分别是Church-Rosser性质和强正则性质，第一个性质是说

强正则化性质指出: 无论采用什么样的方法来规约一个类型化的λ表达式，单步规约不可能无休止进行。我们最终可得到一个正则形，它是一个不可进一步规约的表达式。

在PCF中，我们将递归加入到类型λ演算中，这会使它能编写非终止算法，于是破坏了强正则化性质。当然，PCF规约依然是汇聚的。

4 指称语义

在类型化λ演算的指称语义中，每个类型表达式与一个集合相关联，成为该类型的值集合。比如有类型σ的一个项被解释为具有类型σ的值集的一个元素，该集合可以通过对项的进行规约来定义。

所以类型$\sigma \rightarrow \tau$的值集是一个函数集合，所以一个类型化的λ演算项$\lambda x{:}\,\sigma.M$被解释为一个数学函数。对于熟悉无类型的λ演算的人来说，无类型的λ演算其实是可以通过有类型的λ演算以一种有意义的方式进行派生出来。