函数

前言 Prologue

— 老师, 什么是变量? 什么是未知数? 为什么变量可以换元? 我能只换一个元不换其他的吗?
— 不行, 他们是一个整体.
— 前面几个问题呢? 为什么?
— 这就要讲到我们眼里的数学算式到底是什意思了. 先从最简单的λ-演算讲起吧!

本文我们引申数学中的一些概念, 使得其更加通用.

1 对于函数的探讨(粗浅地)

在高中数学中, 我们学过了函数的概念.它看上去就是下面这样的:

定义 1. (函数的朴素定义)[1] 设A、B是非空的集合, 如果按照某个确定的对应关系f, 使对于集合A中的任意一个元素x, 在集合B中都有唯一确定的元素$f(x)$和它对应, 那么就称f: $A\to B$为从集合A到集合B的一个函数。A称为定义域, B称为陪域. 如果$a\in A, b\in B$,要想表达$f(a)=b$, 可以记为$a\mapsto b$.

但是在20世纪之前, 函数不长这个样子. 更普遍的观点为: "函数实际上是规则". 它把输入按照规则翻译为输出. 例如, 当时的人们看到"$f(x)=x^2$"的类似物, 更可能思考的是: "这个规则说, 输入一个数, 输出这个数的平方."

函数无处不在! 以我们每天都要接触的代数表达式为例, 代数表达式通常由三部分构成: 数字(1, 2,3,1.4,π, ...)、变量(x, y, z)、运算符($+, -, \times, \div, \sqrt{\cdot}, \ldots$). 我们写$x+y$实际上是指代的是$x+y$的**结果**, 而不是关心如何实行加法这个**过程**. 这样的记号一大好处是可以方便地把许多过程粘贴起来. 例如在做代数变形的时候读到$A=(x+y)\times z^2$, 我们不会再考虑"先把x和y加起来, 再把这个中间结果记做t; 对z平方, 中间结果记作w; 最后计算$t\times w$就是结果"这种琐碎的事实.

上世纪的人们实际上找到了一个非常精巧的表达方法来表示: "令f表示$x\mapsto x^2$这一对应关系, 然后我们考虑$A=f(5)$."这样一段话. 他们就会写:

$$A=(\lambda x.x^2)\,(5),$$

这个式子中, λ是一个特殊符号, 意思是: "我要定义一个函数". 紧跟着的x是一个"占位符",对这个虚拟的规则进行句点("."\)后面的操作, 就形成了我们的规则. 也就是说, 映射$f\mapsto f(x)$, 在这里就写作了$\lambda x.f(x)$.

倘若你把这个$\lambda x.x^2$视作函数的记号的话, 后面的(5)就不难理解了. 它表示把这个规则应用到5身上.这里的括号是展示优先级的. 同代数变形时候规则一样, 没有括号的时候从左往右读.

这个"占位符"实际上是一种特殊的变量, 称为**受约束的变量**(bound variable). 其被后面的表达式"拴住了" – 即, 你不能把上述表达式改为$(\lambda y.x^2)(5)$, 但是可以改为$(\lambda y.y^2)(5)$. 这里不做展开.

问题 1. 计算$(\lambda x.(x^3+2x+1))(1)$和$(\lambda y.(y^3+2y+1))(1)$. 他们的结果一样吗?

答案. 都是1+2+1=4.

这样的记号的好处之一是它可以很方便地表达复合函数. 回顾一下复合函数的概念.

定义 2. 设y是u的函数(即$y=f(u)$), u是x的函数(即$u=\phi(x)$). 如果$\phi(x)$的值全部或部分在$f(u)$的定义域内, 则y通过u成为x的函数, 记作$y=f(\phi(x))$, 称为由函数$y=f(u)$以及$u=\phi(x)$复合而成的复合函数。如果希望抛弃掉具体变量而研究对应关系本身的变化则可以写作$y=f\circ\phi$. 这表明对应规则y是由f和ϕ复合而来.

这样的记号是如何表示复合函数呢? 比如, 如果有一个规则f希望表达$f(f(x))$这个函数(或$f\circ f$). 一种简单的办法是:$\lambda x.f(f(x))$.

1. 定义中大家熟悉的部分, 就直接用灰色颜色的字体标识了. 这里的函数定义采用了大学课本的定义: 这是为了为后文从函数到函数的对应关系(函数)埋下伏笔.

前文说到，我们可以完全把这些函数视作了“变化规则”．我们当然可以概念上描述“变化规则”的“变化规则”．这样的内容一般称为“高阶函数”．如下的表达式表达了一个映射：$f\mapsto f\circ f$: $\lambda f.(\lambda x.f(f(x)))$. 也就是这里面的函数也成了占位符．要想求得确切的值，需要给出一个函数和一个自变量才可以．下面给一个例子．对于多重λ嵌套的情况，我们先保证书写的时候每一个λ后面的占位符都不同，以避免混淆．

例 3. 求$((\lambda f.\lambda x.f(f(x)))(\lambda y.y^2))(5)$的值．

解答. 占位符f被替换为了$(\lambda y.y^2)$. 原式变为了$\lambda x.((\lambda y.y^2)(\lambda y.y^2)(x))(5)$. 然后将占位符$x$替换为5，就得到了$((\lambda y.y^2)(\lambda y.y^2)(5))=(\lambda y.y^2)(25)=625$.

问题 2. 求$(((\lambda f.\lambda x.f(f(f(x))))(\lambda g.\lambda y.g(g(y))))(\lambda z.z+1))(0)$. 你可以创造更加方便的记号组织你的思路．

提示1: 括号太多了？下面的办法可以帮助你理清等式结构：从0开始，每一次遇见一个左括号就把数加一，并在这个括号下面写上这个数当前的值；遇见一个右括号就把数减一并写上当前的值．最近的一样的数字就是一对完整的括号．

提示2: 表达式复杂到一定程度的时候，记得想一想其直观意义．

当无法做出答案时：做不出来也没关系，因为有些定义没有很明确地给出．向后面读完再看吧．

答案. 可以借鉴连等式的记号．

$$
\begin{aligned}
&(((\lambda f.\lambda x.f(f(f(x))))(\lambda g.\lambda y.g(g(y))))(\lambda z.z+1))(0)\\
=&(\lambda x.(\lambda g.\lambda y.g(g(y)))((\lambda g.\lambda y.g(g(y)))((\lambda g.\lambda y.g(g(y)))(x)))(\lambda z.z+1))(0)\\
=&(\lambda g.\lambda y.g(g(y)))((\lambda g.\lambda y.g(g(y)))(\underline{(\lambda g.\lambda y.g(g(y)))(\lambda z.z+1)}))(0)\\
=&(\lambda g.\lambda y.g(g(y)))\left((\lambda g.\lambda y.g(g(y)))\underbrace{(\lambda y.((\lambda z.z+1)((\lambda z.z+1)(y))))}_{\text{对}y\text{应用}2\text{次加法}:=Q}\right)(0)\\
=&(\lambda g.\lambda y.g(g(y)))\underbrace{(\lambda y.Q(Q(y)))}_{\text{对}y\text{应用}4\text{次加法}:=R}(0)\\
=&\lambda y.R(R(y))(0)\\
=&0+8=8.
\end{aligned}
$$

上述仅仅说明了对应规则而忽略了对于定义域的讨论．实际上，对于定义域特殊的限制可以附加类型信息表示．我们今天先不做讨论．

2 无类型的λ-演算(untyped λ calculus)

实际上，上面的计算过程和我们进行数学课程的化简工作并无二异．我们为其起一个贴切的名字–演算．由于其中包含λ，又不包含类型信息，故称为无类型的λ-演算．现在是时候规范一下我们对它的定义了．

有两方面定义需要被规范：语法(grammar)和语义(semantics)．语法是指：什么是对，什么是错．比如，“$\lambda x.y$”算合法的算式吗？“$\lambda\lambda$.Bonjour”算合法的算式吗？倘若不能对其语法合法验证，就无从谈起其运算，更无论正确的运算．在定义了语法之后，还需要知道“这个表示什么意思”，“我们可以进行怎样的操作”．这就是语义．

定义 4. $\mathcal{V}$是变量构成的集合(通常有无穷多个)，A是由$\mathcal{V}$中一个或者多个元素组成的集合(称为字母表, alphabet)．不能出现在$\mathcal{V}$中的特殊符号(special symbols)有“(”、“)”、“λ”以及“.”．令$A^\star$为A中元素构成的有限字符串．一个没有语法错误的的λ表达式(lambda-term)是最小的$\Lambda\subseteq A^\star$，使得[2]:

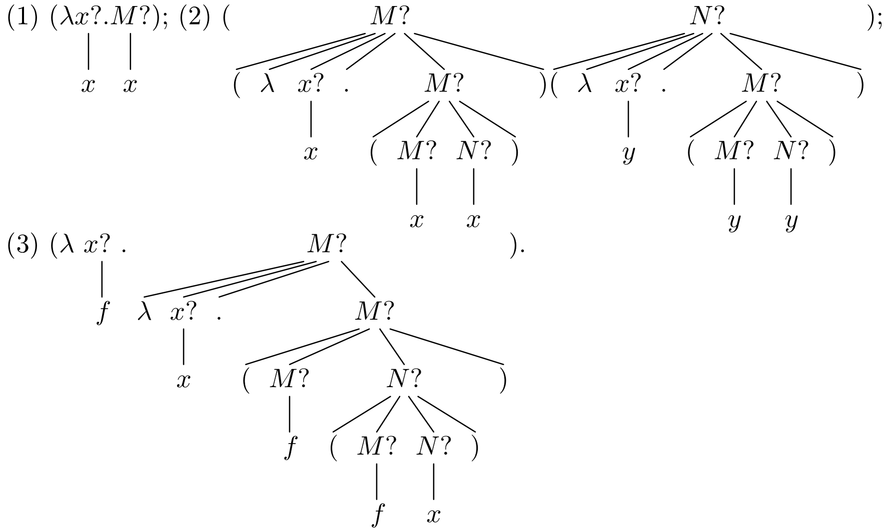
- 只要$x\in\mathcal{V}$，那么$x\in\Lambda$. ——————————变量(variables)
- 只要$M,N\in\mathcal{V}$，那么$(MN)\in\Lambda$. ——————————应用函数规则(applications)
- 只要$x\in\mathcal{V}$且$N\in\Lambda$，那么$(\lambda x.M)\in\Lambda$. ——————————抽象函数声明(abstraction)

2. 右边是每个规则的名称．

例 5. 根据定义判断下列是否为合法的λ项. 默认$\mathcal{V}$是英文二十六个字母构成的有限长度字符串以及整数下标(例如: succ_1,a,b,c). 且$A=\mathcal{V}$. (实际上这也是默认情况下我们的假设, 以后不再指出)

(1) $\lambda x.x$; (2) $((\lambda x.(x\,x))(\lambda y.(y\,y)))$; (3) $(\lambda f.(\lambda x.(f(f\,x))))$.

解答. 他们都是. 他们都可以由上述三条规则生成出来. 其中后面带有?的表示在匹配上述定义中还可以继续往下延伸的占位符(如M,N,x), 没有?的项是实际填入的值或者定义中不可再展开的符号.



问题在于, 刚刚介绍的内容中, 括号使用太频繁了. 我们引入一些约定俗成的记号, 来减小对于括号使用的次数.

惯例 6.

- 我们省略一整项最外面的表达式. 如$(M\,N)$可以写作$M\,N$.
- “应用函数规则”规则是左结合的. 意味着当你看到$M\,N\,P$, 它意味着$(M\,N)P$而不是$M(N\,P)$.
- “抽象函数声明” 规则的身体(句点后面的元素)应该延伸的越远越好. 比如当你看到$\lambda x.MN$意味着$\lambda x.(M\,N)$而不是$(\lambda x.M)N$.
- 多个λ抽象规则连续使用, 可以把前面缩写起来. 如$\lambda x.\lambda y.\lambda z.M$可以写作$\lambda xyz.M$.

2.1 自由和约束的变量, α-等价演算

在最开头的时候, 你看到了$\lambda x.x$中的x只是占位符, 所以$\lambda x.x$和$\lambda y.y$尽管形式不同, 表达的意思确是一样的. 我们说这两者是α-等价的. 即$M \underset{\alpha}{=\!=} N$. 如果希望表达两者每一个字符都是相等的, 可以用$M\equiv N$.

问题 3. 有把大学数学忘记的老师这样教学生: “函数就是$f(x)$或者$y(x)$. 因变量不用x, 自变量不用y. 大家记住了吗?”试说明为什么这是非常严重的常识性错误.

答案. 请你自由发挥.

我们回到这个表示函数的这条“抽象函数声明”规则. 对于形如“$\lambda x.M$”的式子, 由于有了λx这一个函数声明, 就导致了这一项里面所有的x都被约束了. 我们说这叫做变量x**受约束出现**(bound occurence). 这一项的λx, 我们称为**绑定子**(binder).

对于式子中的一个变量而言, 不是受约束出现, 就是**自由出现**(free occurence).

例 7. 请说明$(\lambda x.x\,y)(\lambda y.\,y\,z)$中哪些变量是自由出现的? 哪些是约束出现的? $(\lambda z.z(\lambda z.z\,z))$呢?

解答. $(\lambda x.x\,y)$中, x是约束出现, y是自由出现的. $(\lambda y.\,y\,z)$中, y是约束出现的, z是自由出现的. 注意, 尽管有两个分项中都出现了y, 但是一个是自由出现的, 另一个是约束出现的. 如果把所有的受约束出现的变量周围画上 $\boxed{\text{方框}}$, 并用颜色标注出它的绑定子的话, 就像$(\lambda x.\boxed{x}\ y)(\lambda y.\boxed{y}\ z)$.

问题 4. (1) 演算add $\overline{2}\,\overline{3}$和mult $\overline{2}\,\overline{3}$. (2) 证明: add $\overline{n}\,\overline{m} \underset{\beta}{\twoheadrightarrow} \overline{n+m}$; (3) 证明mult $\overline{n}\,\overline{m} \underset{\beta}{\twoheadrightarrow} \overline{n\cdot m}$. 第一个问题的add $\overline{2}$ $\overline{3}$部分已经为你写好了.

解答范例: (不唯一, 你可以使用"应用n次到函数f"这一算子 – f^n, 这对后两问有巨大帮助).

$$
\begin{aligned}
(\text{add})\ (2)(3) &= (\lambda nmfx.nf(mfx))\ (\lambda fx.f(fx))(\lambda fx.f(f(fx))) \\
&\twoheadrightarrow_\beta \lambda fx.(\lambda fx.f(fx))f((\lambda fx.f(f(fx)))fx) \\
&\underset{\beta}{\twoheadrightarrow} \lambda fx.(\lambda fx.f(fx))\underline{f}\quad \underline{(f(f(fx)))}\ \text{这是两个不同的部分} \\
&\underset{\alpha}{=\!=} \lambda fx.(\lambda gy.g(gy))f(f(f(fx))) \\
&\underset{\beta}{\twoheadrightarrow} \lambda fx.\,(\lambda y.\,f(fy))f(f(f(fx))) \\
&\underset{\beta}{\twoheadrightarrow} \lambda fx.\,(\,f(f\,(f(f(fx))))) \\
&= \overline{5}
\end{aligned}
$$

答案. (1) 仿照上例,

$$
\begin{aligned}
\text{mult}\,\overline{2}\,\overline{3} &\overset{\text{def}}{=\!=} (\lambda nmf.n\,(mf))\,(2)\,(3) \\
&\underset{\beta}{\twoheadrightarrow} (\lambda f.2(3\,f)) \\
&\underset{\beta}{\twoheadrightarrow} (\lambda f.(\lambda fx.\,f^2x)\,((\lambda fx.\,f^3x)\,f)) \\
&\underset{\beta}{\twoheadrightarrow} (\lambda f.(\lambda fx.\,f^2x)\,(\lambda x.\,f^3x)) \\
&\underset{\alpha}{=\!=} (\lambda f.\lambda gy.g^2y(\lambda x.\,f^3x)) \\
&\underset{\beta}{\twoheadrightarrow} \lambda f.\lambda y.\,f^6y \\
&\underset{\alpha}{=\!=} \lambda f.\lambda x.\,f^6x = \overline{6}\,.
\end{aligned}
$$

(2) 只要在合适的地方加上缩写即可.

$$
\begin{aligned}
(\text{add})\ (p)(q) &= (\lambda nmfx.nf(mfx))\,(\lambda fx.f^p\,x)(\lambda fx.f^qx) \\
&\twoheadrightarrow_\beta \lambda fx.(\lambda fx.f^p\,x)f((\lambda fx.f^qx)fx) \\
&\underset{\beta}{\twoheadrightarrow} \lambda fx.(\lambda fx.f^p\,x)))\underline{f}\quad \underline{(f^qx)}\ \text{这是两个不同的部分} \\
&\underset{\alpha}{=\!=} \lambda fx.(\lambda gy.g^p\,y)))f(f^qx) \\
&\underset{\beta}{\twoheadrightarrow} \lambda fx.\,(\lambda y.\,f^py)(f^qx) \\
&\underset{\beta}{\twoheadrightarrow} \lambda fx.\,(f^{p+q}x) \\
&= \overline{p+q}.
\end{aligned}
$$

(3) 和上面的一样.

$$
\begin{aligned}
\text{mult}\,\overline{p}\,\overline{q} &\overset{\text{def}}{=\!=} (\lambda nmf.n\,(mf))\,(\overline{p})\,(\overline{q}) \\
&\underset{\beta}{\twoheadrightarrow} (\lambda f.\overline{p}(\overline{q}\,f)) \\
&\underset{\beta}{\twoheadrightarrow} (\lambda f.(\lambda fx.\,f^px)\,((\lambda fx.f^qx)\,f)) \\
&\underset{\beta}{\twoheadrightarrow} (\lambda f.(\lambda fx.\,f^px)\,(\lambda x.\,f^qx)) \\
&\underset{\alpha}{=\!=} (\lambda f.\lambda gy.g^py(\lambda x.\,f^qx)) \\
&\underset{\beta}{\twoheadrightarrow} \lambda f.\lambda y.\,f^{pq}y \\
&\underset{\alpha}{=\!=} \lambda f.\lambda x.\,f^{pq}x = \overline{pq}\,.
\end{aligned}
$$

对于后面的图示为$(\lambda z.\boxed{z}(\lambda z.\boxed{z}\boxed{z}))$. 注意这里的蓝色的$z$并不是受红色的$z$的约束!

为了方便起见, 在一项λ-表达式中的自由变量构成的集合记作$\mathrm{FV}(M)$. 那么其可以由如下的定义计算:

- $\mathrm{FV}(x) = \{x\}$
- $\mathrm{FV}(M\,N) = \mathrm{FV}(M) \cup \mathrm{FV}(N)$
- $\mathrm{FV}(\lambda x.M) = \mathrm{FV}(M) \backslash \{x\}$

开头中说到, 希望对变量进行替换. 下面, 我们仔细地探讨变量替换到底是什么意思. 请你通过高中数学知识, 尝试判断下面的变量代换是否正确.

问题 5. 说一说, 下面的变量是正确的代换吗? 先从高中数学的感觉谈一谈为什么, 然后根据下面的运算法则核对你的答案.

(1) $\lambda\, x.x$, 在x替换为y之后, 有$\lambda x.y$;

(2) $\lambda x.\lambda y.x\, y$把$x$替换为$z$之后, 有$\lambda x.\lambda y.z\, y$.

答案. 都不对. 第一个就像是把$f(x) = x$替换为了$f(x) = y$; 第二个是把$f(x, y) = x\, y$替换为了$f(x, y) = zy$. 变量的自由性改变了.

惯例 8. 如果x, y是变量, M是某一λ项, 通常用$M\{y / x\}$表示把M中的x字面上替换为y. 其遵循:

- $x\{y / x\} \equiv y$
- $z\{y / x\} \equiv z$, 在$x \neq z$的情况下
- $M\,N\{y / x\} \equiv (M\{y / x\})(N\{y / x\})$
- $(\lambda x.M)\{y / x\} \equiv \lambda\, y.(M\{y / x\})$
- $(\lambda z.M)\{y / x\} \equiv \lambda z.(M\{y / x\})$, 在$x \neq z$的情况下.

有了什么叫做变量代换的初步讨论, 我们就可以为α-等价下一个定义了.

定义 9. (α-等价) 对于任何一个λ-项M, 只要变量y在M中没有出现过, 我们就可以把M中的一个变量**重新命名**为y. 即$\lambda x.M \underset{\alpha}{=\!=} \lambda y.(M\{y / x\})$. 我们说这两个式子$\alpha$-等价.

α-等价关系具有对称性(如果$M \underset{\alpha}{=\!=} N$, 那么自然$N \underset{\alpha}{=\!=} M$); 传递性(如果$A \underset{\alpha}{=\!=} B, B \underset{\alpha}{=\!=} C$, 那么$A \underset{\alpha}{=\!=} C$); 自反性($M \underset{\alpha}{=\!=} M$).

为了方便排版和阅读, 我们把若P则Q写作$\frac{P}{Q}$: 上面是前提, 下面是结论. 实际上, α-等价的规则有如下几条:

$$(1, \text{自反})\,\frac{}{M = M} \qquad (2, \text{对称})\,\frac{M = N}{N = M}$$

$$(3, \text{传递})\,\frac{M = N \qquad N = P}{M = P} \qquad (4, \text{等价})\,\frac{M = M' \qquad N = N'}{MN = M'N'}$$

$$(5, \xi)\,\frac{M = M'}{\lambda x.M = \lambda x.M'} \qquad (6, \alpha)\,\frac{y \notin M}{\lambda x.M = \lambda y.(M\{y / x\})}$$

例 10. 写出下列相互等价表达式的α-等价推演过程, 每一步骤只能使用一条规则:

(1) $\lambda x.\lambda y.x\, y \underset{\alpha}{=\!=} \lambda x.\lambda y.\, y\, x$. (2) $\lambda z.z(\lambda z.z\, z) \underset{\alpha}{=\!=} \lambda z.z(\lambda y.y\, y)$.

解答. (1)$\lambda x.\lambda y.x\, y \underset{\alpha,6}{=\!=\!=} \lambda x.\lambda z.x\, z \underset{\alpha,6}{=\!=\!=} \lambda y.\lambda z.y\, z \underset{\alpha,6}{=\!=\!=} \lambda y.\lambda x.y\, x$.

(2) $\lambda z.z(\lambda z.zz) \underset{\alpha,6}{=\!=\!=} \lambda z.z(\lambda y.yy)$.

问题 6. 判断下列的三个表达式中, 哪两个是α-等价的. 给出推演过程. (提示: 留意**每一个位置上的**变量是自由出现的还是约束出现的有助于初步地判断).

(1) $\lambda x.y\,\lambda a.a\,x$
(2) $\lambda x.z\,\lambda b.b\,x$
(3) $\lambda a.\,y\,\lambda b.\,b\,a$

答案. (1)和(3). 对于变量的自由性可以直接看出来. 推导和上例一样, 做一次交换变量即可.

问题 7. 给出命题: 如果$P \underset{\alpha}{=\!=} Q$, 那么$\mathrm{FV}(P) = \mathrm{FV}(Q)$. 先用自己的语言叙述这个命题, 然后给出证明[3].

答案. 如果P和Q是α-等价的, 那么P和Q的自由变量构成的集合(在相对位置层面)是一样的. 考虑推导过程. 证明见OpenLogic项目的lam:syn:alp:lem:fv-one词条.

问题 8. 有同学对求和记号产生了疑问: 对于$\sum_{i=0}^{10} i$来讲, 如果通过变量替换的方法把i代换为$i+1$, 原式就会变为$\sum_{i+1=0}^{10} i+1$, 与原来的结果不一致. 这说明α-等价变换的核心思想失效了吗? 提示: $\sum_{i=a}^{b} f(i)$只是$\sum_{a\leqslant i\leqslant b} f(i)$的简写.

答案. 使用$\sum_{a\leqslant i\leqslant b} f(i)$就成立了. 实际上$\sum_{i=a}^{b} f(i)$并没有选的足够好来满足这一特性, 这个记号只是足够简单.

2.2 变量替换

上节了解了如何对变量重新命名. 下面来探讨变量替换. 变量替换意味着我们可以把一个变量替换为一个λ-项. 为了方便起见, 还是沿用上节的记号

惯例 11. 用记号$M[N/x]$表示在项M中把变量x换成另一项N.

但是事情总是没有那么简单: 来源主要在引进的λ-项中也可能有被绑定的变量, 万一这两部分重叠了, 我们的表意就不一样了.

- 我们只能替换自由变量. 因为受约束的变量实质上只是一个占位符, 并不是一个具体的东西, 自然不能被具体的表达式所替换. 因此, $x(\lambda x y.\,x)[N/x]$实际的结果是$N(\lambda x\,y.\,x)$, 而不是$N(\lambda x y.N)$.
- 需要注意不要不小心把原本是自由的变量变成约束的变量. 例如, 对于$M \equiv \lambda x.\,y\,x, N \equiv \lambda z.\,x\,z$. x在N中自由, 但是在M中受约束. 把M中的变量y替换为N应该怎么办呢?
    - 如果按照前面的规则, 就成了$M[N/y] = (\lambda x.y\,x)[N/y] = \lambda x.\,N\,x = \lambda x.\,(\lambda z.\,x\;z)x$.
    - 不好! 后面本应该是自由的变量x这里居然被最前的x约束了! 他们本不应该是同一个x, 但是现在他们却是同一个x了. 前后两个意思就不一样了. 怎么办?
    - 实际上在实际带入之前为它改个名就行了. 比如我们可以把M中受约束的变量经过α-等价变换变为另一个不冲突的名字. 比如$M[N/y] = (\lambda x'.y\,x')[N/y] = \lambda x'.N\,x' = \lambda x'.(\lambda z.x\,z)x'$. 可见这种代换有时候必须强制重新命名. 这时候最好选一个在两个式子中从来没有出现过的变量, 我们称这样的变量为**新鲜的**(fresh).

定义 12. 为防止变量替换时造成的意外约束, 在替换N中的自由变量x为项M的时候, 可以用符号简单记作$M[N/x]$, 定义如下:

- $x[N/x] \equiv N$,

[3]. 完成此问题至少需要学过结构归纳法和集合的运算规则(即本科一年级«离散数学»课程内容). 高中生可跳过. 或参考OpenLogic项目的lam:syn:alp:lem:fv-one词条.

- $y[N/x] \equiv y$, 当$x \neq y$的时候
- $(M\,P)[N/x] \equiv (M[N/x])(P[N/x])$
- $(\lambda x.\,M)[N/x] \equiv \lambda x.M$
- $(\lambda x.M)[N/x] \equiv \lambda y.(M[N/x])$, 当$x \neq y$且$y \notin \mathrm{FV}(N)$的时候
- $(\lambda x.M)[N/x] \equiv \lambda y'.(M\{y'/y\})[N/x]$,当$x \neq y$, $y \notin \mathrm{FV}(N)$, y'是新鲜的时候

问题 9. 请对比$M\{y/x\}$和$M[N/x]$, 说说哪里有不同, 并且指出多加的部分是为了解决上面讨论的哪些问题.

答案. 多加了自由变量的限制. 为了防止替换掉受约束变量, 在前提条件中加了“自由变量x”这一限定词, 为了防止重名, 多出了(6), (7)两条规则.

2.3 β-化简

刚刚只是见到了通过变量的重命名、变量替换的手法在各个不同的λ项之间替换. 接下来我们就要看如何将表达式进行化简. 也就是“意思上的相等”. 接下来我们要做的是如何表示“把某个值带到某个函数里面去”或“把一个值应用于函数”. 这就是β-化简要做的事情.

前文提到, 高中数学中写$f(5)$表示按照对应规则f查找5会变成什么; 在λ-表达式中我们会写$f\ 5$. 如果我们知道f的规则是$f(x) = x + 1$, 那么这里就可以写$(\lambda x.x + 1)5$.

像这样$(\lambda x.M)N$的表达式 – 正如语法名称“应用函数规则”所隐含的那样 – 表示把前面的占位符全都代换为N – 即$M[N/x]$. 这样的一步就可以看做对表达式进行了一次化简.

例 13. 按上述规则化简$(\lambda x.y)((\lambda z.\,z\,z)(\lambda w.\,w))$.

解答.

$$\begin{aligned}(\lambda x.y)\big(\underline{(\lambda z.zz)\boxed{(\lambda w.w)}}\big) &\xrightarrow[\beta]{} (\lambda x.y)\big(\underline{(\lambda w.w)\boxed{(\lambda w.w)}}\big)\\ &\xrightarrow[\beta]{} \underline{(\lambda x.y)\boxed{(\lambda w.w)}}\\ &\xrightarrow[\beta]{} y.\end{aligned}$$

另一种方法是: $\underline{(\lambda x.y)\boxed{((\lambda z.zz)(\lambda w.w))}} \xrightarrow[\beta]{} y$. 可以发现选择不同的步骤进行化简的步数也不一样.

实际上, 最终答案并不会因为我们如何化简这一过程而发生改变(后面会有证明). 最后化简的结果只要没有形如$(\lambda x.M)N$的形式, 我们就说它已经化简到**最简形式**(normal form)了. 如果经过零步或多步, 化简操作把初始的M化为了M'(记作$M \twoheadrightarrow_{\beta} M'$), 我们就说$M$的求值结果(evaluates to)为$M'$.

实际上, 并不是所有的式子都有最简结果的. 例如$(\lambda x.x\,x)(\lambda y.\,y\,y\,y)$在求值过程中会持续膨胀!

问题 10. 根据规则, 写出$(\lambda x.x\,x)(\lambda y.\,y\,y\,y)$的前几步. $(\lambda x.x\,x)(\lambda x.\,x\,x)$有最简形式吗?

答案. 没有. 第一个一直膨胀, 第二个一直不变.

$$\begin{aligned}(\lambda x.xx)(\lambda y.yyy) &\xrightarrow[\beta]{} (\lambda y.yyy)(\lambda y.yyy)\\ &\xrightarrow[\beta]{} (\lambda y.yyy)(\lambda y.yyy)(\lambda y.yyy)\\ &\xrightarrow[\beta]{} (\lambda y.yyy)(\lambda y.yyy)(\lambda y.yyy)(\lambda y.yyy)\\ &\xrightarrow[\beta]{} \cdots\end{aligned}$$

总结一下，β-化简的操作规则大体有以下几条：

$$
\begin{array}{ll}
(1,\beta) & \dfrac{}{(\lambda x.M)N \underset{\beta}{\longrightarrow} M[N/x]} \\[2ex]
(2,\text{等价}_{\text{左}}) & \dfrac{M \underset{\beta}{\longrightarrow} M'}{MN \underset{\beta}{\longrightarrow} M'N} \\[2ex]
(3,\text{等价}_{\text{右}}) & \dfrac{N \underset{\beta}{\longrightarrow} N'}{MN \underset{\beta}{\longrightarrow} MN'} \\[2ex]
(4,\xi) & \dfrac{M \underset{\beta}{\longrightarrow} M'}{\lambda x.M \underset{\beta}{\longrightarrow} \lambda x.M'}
\end{array}
$$

3 表达一切!

我们前面说到，λ-表达式只能有一堆奇怪的东西，而似乎连加减法都没办法做. 接下来试图使用λ-表达式来编码一些常见的概念.

3.1 真与假

我们定义如下的表达式为真 (True, $\mathbb{T}$) 与假 (False, $\mathbb{F}$):

$$
\begin{aligned}
\mathbb{T} &= \lambda xy.x \\
\mathbb{F} &= \lambda xy.y
\end{aligned}
$$

我们定义and算子为and $= \lambda\, a\, b.\, a\, b\, \mathbb{F}$. 可以发现在$\beta$-化简的意义下的and和逻辑符号中的and只是记号上的不同.

问题 11. 通过β-化简验证: (1) and $\mathbb{T}\,\mathbb{T} \underset{\beta}{\twoheadrightarrow} \mathbb{T}$; (2) and $\mathbb{T}\mathbb{F} \underset{\beta}{\twoheadrightarrow} \mathbb{F}$; (3) and $\mathbb{F}\,\mathbb{T} \underset{\beta}{\twoheadrightarrow} \mathbb{F}$; (4) and $\mathbb{F}\,\mathbb{F} \underset{\beta}{\twoheadrightarrow} \mathbb{F}$. 提示: and $\mathbb{T}\,\mathbb{T} \underset{\beta}{\twoheadrightarrow} \mathbb{T}$实际上是(and) ($\mathbb{T}$) ($\mathbb{T}$). 注意括号顺序!

答案.

$$
\begin{aligned}
(\lambda ab.ab\mathbb{F})\,\underline{\mathbb{T}\mathbb{T}} &\underset{\beta}{\twoheadrightarrow} \mathbb{T}\,\underline{\mathbb{T}\mathbb{F}} \\
&\underset{\beta}{\twoheadrightarrow} \mathbb{T} \\
(\lambda ab.ab\mathbb{F})\,\underline{\mathbb{T}\mathbb{F}} &\underset{\beta}{\twoheadrightarrow} \mathbb{T}\,\underline{\mathbb{F}\mathbb{F}} \\
&\underset{\beta}{\twoheadrightarrow} \mathbb{F} \\
(\lambda ab.ab\mathbb{F})\,\underline{\mathbb{F}\mathbb{T}} &\underset{\beta}{\twoheadrightarrow} \mathbb{F}\underline{\mathbb{T}\mathbb{F}} \\
&\underset{\beta}{\twoheadrightarrow} \mathbb{F} \\
(\lambda ab.ab\mathbb{F})\,\underline{\mathbb{F}\mathbb{F}} &\underset{\beta}{\twoheadrightarrow} \mathbb{F}\underline{\mathbb{F}\mathbb{F}} \\
&\underset{\beta}{\twoheadrightarrow} \mathbb{F}.
\end{aligned}
$$

实际上and算子的定义不是唯一的，定义and $= \lambda\, a\, b.b\, a\, b$也可以完成一样的效果.

问题 12. 根据上面的推导过程，请你构造出or算子和not算子.

答案. 首先定义not算子为$\lambda ab.ba$; 然后实际上or算子就是not and.

实际上这种情形可以表达控制流. 比如 if_then_else=$\lambda x.\ x$. 比如,

$$\text{if\_then\_else}\ \mathbb{T} M N \underset{\beta}{\twoheadrightarrow} M$$
$$\text{if\_then\_else}\ \mathbb{F} M N \underset{\beta}{\twoheadrightarrow} N$$

3.2 自然数

在解决这个之前, 首先约定俗成地, 记$f^n x := \underbrace{f(f(f_{\dots}(fx)_{\dots}))}_{n次}$. 我们的自然数$n$被定义做: $\overline{n} = \lambda\, f x.(f^n x)$.

问题 13. 请你写出$\overline{0}$,$\overline{1}$,$\overline{2}$,$\overline{3}$的λ-表达式的定义.

答案.

$$\begin{aligned}
\overline{0} &= \lambda f x.x \\
\overline{1} &= \lambda f x.\ f x \\
\overline{2} &= \lambda f x.\ f\,(f x) \\
\overline{3} &= \lambda f x. f(f(f x)) \\
&\dots
\end{aligned}$$

在定义了之后, 自然要定义其操作. 比如“寻找自然数的后继”, 即“加1”. 定义为

$$\text{succ} = \lambda n f x.\ f(n f x)$$

比如要计算succ $\overline{1} = (\lambda n\ f x.\ f(n\ f x))(\lambda f x.\ f x) \underset{\beta}{\twoheadrightarrow} \lambda f x.\ f((\lambda f x.\ f x)\ f x) \underset{\beta}{\twoheadrightarrow} \lambda f x. f(f x) = \overline{2}$.

问题 14. 请使用数学归纳法证明: succ $\overline{n} = \overline{n+1}$.

答案. 基础仿照上述可证明. 假设对于n成立, 那么

$$\begin{aligned}
\text{succ}\ \overline{n} &= (\lambda n f x.\ f(n f x))(\lambda f x.\ f^n x) \\
&\underset{\beta}{\twoheadrightarrow} \lambda f x. f((\lambda f x.\ f^n x) f x) \\
&\underset{\beta}{\twoheadrightarrow} \lambda f x. f(f^n\, x) \\
&\underset{\beta}{\twoheadrightarrow} \lambda f x.\ f^{n+1} x \\
&= \overline{n+1}.
\end{aligned}$$

自然也可以定义自然数的加法和乘法. 可以定义

$$\begin{aligned}
\text{add} &= \lambda n m f x.n f(m f x) \\
\text{mult} &= \lambda n m f.n\,(m f)
\end{aligned}$$

问题 15. (1) 演算add $\overline{2}\ \overline{3}$和mult $\overline{2}\ \overline{3}$. (2) 证明: add $\overline{n}\ \overline{m} \underset{\beta}{\twoheadrightarrow} \overline{n+m}$; (3) 证明mult $\overline{n}\ \overline{m} \underset{\beta}{\twoheadrightarrow} \overline{n \cdot m}$. 第一个问题的add $\overline{2}\ \overline{3}$部分已经为你写好了.

解答范例: (不唯一, 你可以使用“应用n次到函数f”这一算子 – f^n, 这对后两问有巨大帮助).

$$\begin{aligned}
(\text{add})\ (2)(3) &= (\lambda n m f x.n f(m f x))\ (\lambda f x. f(f x)))(\lambda f x. f(f(f x))) \\
&\twoheadrightarrow_\beta \lambda f x.(\lambda f x. f(f x)))\, f((\lambda f x. f(f(f x)))\, f x) \\
&\underset{\beta}{\twoheadrightarrow} \lambda f x.(\lambda f x. f(f x)))\, \underline{f} \quad \underline{(f(f(f x)))}\ \text{这是两个不同的部分} \\
&\underset{\alpha}{=\!=} \lambda f x.(\lambda g y. g(g y)))\, f(f(f(f x))) \\
&\underset{\beta}{\twoheadrightarrow} \lambda f x.\ (\lambda y.\ f(f y))\, f(f(f(f x))) \\
&\underset{\beta}{\twoheadrightarrow} \lambda f x.\ (\ f(f\ (f(f(f x))))) \\
&= \overline{5}
\end{aligned}$$

答案. (1) 仿照上例,

$$\begin{aligned}
\text{mult}\,\overline{2}\,\overline{3} &\overset{\text{def}}{=\!=} (\lambda nmf.n\,(m\,f))\,(2)\,(3)\\
&\underset{\beta}{\twoheadrightarrow} (\lambda f.2(3\,f))\\
&\underset{\beta}{\twoheadrightarrow} (\lambda f.(\lambda fx.\,f^2x)\,((\lambda fx.\,f^3x)\,f))\\
&\underset{\beta}{\twoheadrightarrow} (\lambda f.(\lambda fx.\,f^2x)\,(\lambda x.\,f^3x))\\
&\underset{\alpha}{=\!=} (\lambda f.\lambda gy.\,g^2y(\lambda x.\,f^3x))\\
&\underset{\beta}{\twoheadrightarrow} \lambda f.\lambda y.\,f^6y\\
&\underset{\alpha}{=\!=} \lambda f.\lambda x.\,f^6x = \overline{6}\,.
\end{aligned}$$

(2) 只要在合适的地方加上缩写即可.

$$\begin{aligned}
(\text{add})\,(p)(q) = {} & (\lambda nmfx.n\,f(m\,fx))\,(\lambda fx.\,f^p\,x)(\lambda fx.\,f^qx)\\
\twoheadrightarrow_\beta {} & \lambda fx.(\lambda fx.\,f^p\,x)\,f((\lambda fx.\,f^qx)\,fx)\\
\underset{\beta}{\twoheadrightarrow} {} & \lambda fx.(\lambda fx.\,f^p\,x)))\,\underline{f}\quad \underline{(f^qx)}\ \text{这是两个不同的部分}\\
\underset{\alpha}{=\!=} {} & \lambda fx.(\lambda gy.\,g^p\,y)))\,f(f^qx)\\
\underset{\beta}{\twoheadrightarrow} {} & \lambda fx.\,(\lambda y.\,f^py)(f^qx)\\
\underset{\beta}{\twoheadrightarrow} {} & \lambda fx.\,(f^{p+q}x)\\
= {} & \overline{p+q}.
\end{aligned}$$

(3) 和上面的一样.

$$\begin{aligned}
\text{mult}\,\overline{p}\,\overline{q} &\overset{\text{def}}{=\!=} (\lambda nmf.n\,(m\,f))\,(\overline{p})\,(\overline{q})\\
&\underset{\beta}{\twoheadrightarrow} (\lambda f.\overline{p}(\overline{q}\,f))\\
&\underset{\beta}{\twoheadrightarrow} (\lambda f.(\lambda fx.\,f^px)\,((\lambda fx.\,f^qx)\,f))\\
&\underset{\beta}{\twoheadrightarrow} (\lambda f.(\lambda fx.\,f^px)\,(\lambda x.\,f^qx))\\
&\underset{\alpha}{=\!=} (\lambda f.\lambda gy.\,g^py(\lambda x.\,f^qx))\\
&\underset{\beta}{\twoheadrightarrow} \lambda f.\lambda y.\,f^{pq}y\\
&\underset{\alpha}{=\!=} \lambda f.\lambda x.\,f^{pq}x = \overline{pq}\,.
\end{aligned}$$

上面的讨论中, 已经发现虽然写出来是一长串复杂难以理解的内容, 但是它与我们思想中的自然数、加法、乘法是具有一样的功能的. 我们说这就是λ-表达式对我们思维的编码.

我们再来看一个例子: 判断一个数是不是为0. 我们思想中的判定是否为0的函数为: $\text{iszero}(0) = \text{true}$, $\text{iszero}(m) = \text{false}(m \neq 0)$. 它的λ-编码为:

$$\text{iszero} = \lambda nxy.n(\lambda z.\,y)x.$$

问题 16. 通过写出iszero $\overline{0}$, iszero $\overline{2}$的演算过程, 感受此定义为什么是正确的.

($\mathbb{T}$ $=\lambda xy.x$; $\mathbb{F}$ $=\lambda xy.y$; $\overline{n} = \lambda\,f\,x.(f^nx)$.).

答案. 首先看iszero $\bar{0}$:

$$\begin{aligned}(\text{iszero})(\bar{0}) &= (\lambda nxy.\,(n(\lambda z.\,y)x))(\lambda fx.x)\\ &= (\lambda nxy.\,(n(\lambda z.\,y)x))(\lambda gw.w)\\ &\underset{\beta}{\twoheadrightarrow} \lambda xy.\,((\lambda gw.w)\,(\lambda z.\,y)x)\\ &\underset{\beta}{\twoheadrightarrow} \lambda xy.x = \mathbb{T}.\end{aligned}$$

但是

$$\begin{aligned}(\text{iszero})(\bar{2}) &= (\lambda nxy.\,(n(\lambda z.\,y)x))(\lambda gw.\,g^2w)\\ &\underset{\beta}{\twoheadrightarrow} (\lambda xy.\,((\lambda gw.\,g^2w)(\lambda z.\,y)x))\\ &\underset{\beta}{\twoheadrightarrow} \lambda xy.\,(\lambda z.\,y)^2x\\ &\underset{\beta}{\twoheadrightarrow} \lambda xy.\,y = \mathbb{F}.\end{aligned}$$

当然, 上述的内容看上去相当的随机. 倘若自己思考可能要花费相当多的时间来尝试. 下面我们要做的是看一看能不能找到他们中间的公共部分, 从而机械化这个过程.

在这之前, 最后再看最后一类函数的例子 – 递归函数.

3.3 不动点和递归函数

定义 14. 点x是函数f的一个不动点, 满足$f(x)=x$.

类似地, 如果F,N是λ-项, N是F的一个不动点如果$FN \underset{\beta}{=} N$.

实际上, 令人惊讶的是:

定理 15. 在无类型的λ-演算中, 每一个项F都有一个不动点.

证明. 令$A=\lambda x\,y.y(x\,x\,y)$, 定义$\Theta = A\,A$. 令$F$是任意的$\lambda$-项, 通过$N=\Theta\,F$, 就可以声称$N$是$F$的一个不动点.

$$\begin{aligned}N &= \Theta F\\ &= AAF\\ &= \lambda xy.y\,(xxy)\,\text{AF}\\ &\underset{\beta}{\twoheadrightarrow} F(AAF)\\ &= F(\Theta F)\\ &= FN\end{aligned}$$

这表明了$N=\Theta\,F=F\,N$.

□

现在假设我们希望表达阶乘函数: fact. 从高中数学中我们知道$\text{fact}(x)=\begin{cases}x\cdot\text{fact}(x-1), x\geqslant 1\\ 1, x=0\end{cases}$; 翻译为$\lambda$-表达式就是:

$$\text{fact}\,n = \lambda n.\text{if\_then\_else}\,(\text{iszero}\,n)(\bar{1})(\text{mult}\,n(\text{fact}\,(\text{pred}\,n)))$$

其中pred函数由于比较复杂, 在前面并没有展开介绍. 它大概实现的作用是: $\text{pred}(\bar{n})=\begin{cases}\overline{n-1}, n>0\\ \bar{0}, n=0\end{cases}$. 有兴趣的同学可以试着找到他们的$\lambda$-表达式.

现在要做的是, 希望把等式右边的fact消去, 从而使得等式左右两边只有一边出现fact. 换句话说, 这就是“解含有fact的方程”.

让我们看看不动点对解这样的方程有什么帮助. 首先改写形式为

$$\begin{aligned}\mathrm{fact} &= \lambda n.\mathrm{if\_then\_else}\,(\mathrm{iszero}\,n)(\bar{1})(\mathrm{mult}\,n(\mathrm{fact}\,(\mathrm{pred}\,n)))\\ &= \underbrace{\lambda f\,.(\lambda n.\mathrm{if\_then\_else}\,(\mathrm{iszero}\,n)(\bar{1})(\mathrm{mult}\,n(f\ (\mathrm{pred}\,n))))}_{\text{先记作}F}\,\mathrm{fact}\\ &= F\,\mathrm{fact}\end{aligned}$$

这时候使用不动点的定义, 就可以解出fact. 根据定理15最后的等式知道:

$$\begin{aligned}\mathrm{fact} &= \Theta\,F\\ &= \Theta\,(\lambda f\,.(\lambda n.\mathrm{if\_then\_else}\,(\mathrm{iszero}\,n)(\bar{1})(\mathrm{mult}\,n(f\ (\mathrm{pred}\,n)))))\end{aligned}$$

注意等式右边的fact就被消掉了, 转而就解出了fact的真实值.

例 16. 求fact $\bar{2}$的前几步.

解答. 我们把多步求值放在一起, 以防止过于杂乱:

$$\begin{aligned}\mathrm{fact}\,\bar{2} &\underset{\beta}{\twoheadrightarrow} F\,\mathrm{fact}\,\bar{2} \quad \left(\text{定理 15}: \Theta\,F \underset{\beta}{\twoheadrightarrow} F(\Theta\,F)\right)\\ &\underset{\beta}{\twoheadrightarrow} \mathrm{if\_then\_else}\,(\mathrm{iszero}\ \bar{2}\,)(1)(\mathrm{mult}\,\bar{2}\,(\mathrm{fact}\,(\mathrm{pred}\,\bar{2})))\\ &\underset{\beta}{\twoheadrightarrow} \mathrm{if\_then\_else}\,(\mathbb{F})(1)(\mathrm{mult}\,\bar{2}\,(\mathrm{fact}\,(\mathrm{pred}\,\bar{2})))\\ &\underset{\beta}{\twoheadrightarrow} \mathrm{mult}\,\bar{2}\,(\mathrm{fact}\,(\mathrm{pred}\,\bar{2})) \underset{\beta}{\twoheadrightarrow} \mathrm{mult}\,\bar{2}\,(\mathrm{fact}\,1)\\ &\ldots.\\ &\underset{\beta}{\twoheadrightarrow} \mathrm{mult}\,(\bar{2}\,\mathrm{mult}(\bar{1}\,\bar{1})).\end{aligned}$$