3. 编程基础Ⅲ：Lambda演算

一. Lambda演算

形式化地，从一个标识符(identifier)的可数无穷集合开始(比如{a, b, c, …, x, y, z, x1, x2, …})，则所有的Lambda表达式可以通过下述以BNF范式表达的上下文无关文法描述：

<表达式> ::= <标识符>

<表达式> ::= (λ<标识符>. <表达式>)

<表达式> ::= (<表达式><表达式>)

前两条规则用来生成函数，第三条规则描述了函数是如何作用在参数上的.

Example：

· $(\lambda x.\ 2x)$

$(\lambda x\ y.\ x+y)\ a\ b$

定义(λ项)：假设有一个无穷的字符串集合，里面的元素被称为变量(和程序语言中变量不同，这里就是指字符串本身)，那么λ项定义如下：

1. 所有的变量都是λ项(名为原子).
2. 若M和N是λ项，那么(M N)也是λ项(名为应用).
3. 若M是λ项而φ是一个变量，那么(λφ. M)也是λ项(名为抽象).

Example：

$(\lambda x.(xy))$ $(x(\lambda x.(\lambda x.x)))$ $((((ab)c)d)e)$

$(((\lambda x.(\lambda y.(yx)))a)b)$ $((\lambda y.y)(\lambda x.(xy)))$ $(\lambda x.yz)$

符号约定(省略)

1. λ项最外层的括号可以省略，如(λx.x)可省略为λx.x.
2. 左结合的应用型的λ项，如(((M N)P)Q)可省略为M N P Q.
3. 抽象型的λ项(λφ. M)中，M最外层的括号可以省略，如λx.(yz)可以省略为λx.yz.

Example： $\lambda x.\lambda y.\ yxab$ $(\lambda x.(\lambda y.(((yx)a)b)))$

$\lambda x.\lambda y.\ ab\ \lambda z.z$ $(\lambda x.(\lambda y.((ab)(\lambda z.z))))$

定义2（语法全等）：用恒等号表示两个λ项完全相同，即

$$M \equiv N$$

表示M和N有完全相同的结构，且对应位置上的变量也完全相同。

定义3（自由变量）对于一个λ项P，定义P中**自由变量**集合$FV(P)$：

1. $FV(\phi) = \{\phi\}$.
2. $FV(\lambda\phi.M) = FV(M) \setminus \{\phi\}$.
3. $FV(M\ N) = FV(M) \cup FV(N)$.

在第2条中，若M中有φ，则称φ是被**约束**的。

若$FV(P)$为空集，则称φ是**封闭**的，P是**组合子**。

定义4（α变换与α等价）：设$\lambda\phi.M$出现在一个λ项P中，且设$\psi \notin FV(M)$，那么称将$\lambda\phi.M$替换成$\lambda\psi.[\psi/\phi]M$的操作为P的α变换。λ项P与Q是α等价的当且仅当P经过有限步α变换后得到Q，记作$P \equiv_\alpha Q$.

定义5（β规约）：形如$(\lambda\phi.M)N$的λ项被称为β可约式，对应的项$[N/\phi]M$称为β缩减项。当P中含有$(\lambda\phi.M)N$时，将P中的$(\lambda\phi.M)N$整体替换成$[N/\phi]M$，用R指称替换后得到的项，此时称P被β缩减为R，记作$P \rhd_{1\beta} R$。当P经过有限步的β缩减后得到Q，则称P被β规约到Q，记作$P \rhd_\beta Q$.

定理：

(1)

(2)（Church-Rosser）

(3)

(4)

(5)

符号约定2：用粗体大写字母及其字符串表示对应组合子。

二. 算术计算与逻辑谓词.

三. 递归与 Y Combinator

四、有序对.