

# 对 Monad 的理解

姓名：张凯

学号：2012013311

邮箱：nezharen@163.com

Haskell 作为一种函数式编程的语言，其所有计算过程都依赖于数学推导，可以说是十分强大。然而这一种语言却有一种硬伤，那就是在面对非纯函数时，例如输入输出函数、求随机数等函数的结果并不总是相同的情况时，就会违背 Haskell 的特性。为了解决这一问题，我们采用的是为函数增加参数的方式，例如求随机数的时候会有一个种子参数。因此 rand 函数的参数是一个 Integer 类型的 seed，返回值是 (val, seed')。seed' 是一个新的种子。然而在我们每次使用非纯函数时如果都要手动控制这些附加参数，可能会非常繁琐。考虑到这类函数的特征，往往是以一个状态作为参数，然后返回一个结果与一个新的状态。因此可以将这类函数总结为一种类型，这便是 Monad 类型类。在 Monad 中定义了一种操作符 “>>=”，它表示绑定，课件中的 bind 函数代码已经非常明确的表示了这个操作符的过程。因此 `rand >>= \a` 的结果是 rand seed 的结果绑定到了 a。do 语句块只是一个语法糖，对 Monad 语句进行了简化从而方便编程人员使用。