**函数式编程与量子力学**

惰性（或延迟）求值是函数式编程一项有趣的技术，考虑下面的代码片断：   
var s1 = somewhatLongOperation1();   
var s2 = somewhatLongOperation2();   
var s3 = concatenate(s1, s2);   
  
      在一个命令式语言中求值顺序是确定的，因为每个函数都有可能会变更或依赖于外部状态，所以就必须有序的执行这些函数：首先是somewhatLongOperation1，然后 somewhatLongOperation2，最后 concatenate，在函数式语言里就不尽然了。   
      只要确保没有函数修改或依赖于全局变量，somewhatLongOperation1 和 somewhatLongOperation2 可以被并行执行。但是如果我们不想同时运行这两个函数，还有必要保证有序的执行他们呢？答案是不。我们只在其他函数依赖于s1和s2时才需要执行这两个函数。我们甚至在concatenate调用之前都不必执行他们——可以把他们的求值延迟到concatenate函数内实际用到他们的位置。如果用一个带有条件分支的函数替换concatenate并且只用了两个参数中的一个，另一个参数就永远没有必要被求值。在函数式语言中，不确保一切都（完全）按顺序执行，因为函数式只在必要时才会对其求值。   
      例如，在JavaScript中，我们可能这么写：   
  
function concatenate(s1, s2)   
{   
       if(cond1) s1();   
       s2();   
       ……   
}   
var s3 = concatenate(somewhatLongOperation1,somewhatLongOperation2);   
  
      假如cond1的条件不满足，那么somewhatLongOperation1就不需要被执行，这样从一定程度上强化了程序逻辑的优化潜力。   
  
   近代量子物理学从很大程度上来讲，是对经典物理学的一次颠覆，而量子物理理论中最不可思议的部分就是它的“不确定性”。电子像幽灵般化身为“波”，同时穿过双缝，在荧光屏上呈现出干涉条纹，然而当你在某个地方放上一个检测装置时，它又坍缩成为一个“点”，要么被检测到通过装置，要么不被检测到。   
    围绕着量子物理的不确定性，有很多种解释，比较知名的有高维宇宙论，多宇宙论、平行宇宙论、多历史理论等等，它们分别都能对量子的怪异行为做出某些解释，然而也都被一些诸如薛定鄂猫之类的魔鬼所困扰。   
不过，如果以程序的观点来考虑的话，那么一种解释是，这个世界是被“延迟求值”的。当你不去观察电子的时候，电子的“值”（粒子）是不存在的，因为没有被求值，只有它的“函数”（波）引用存在于这个宇宙系统中。当它被某些事件关联而求值的时候，它的值才存在。   
      当你观察一个微观的电子时，因为它的函数尺度比较小，层级比较低，相关联的干涉行为比较少，所以它的求值次序和结果表现为一定的随机性，而当你观察一个宏观物体时，由于它的函数尺度比较大，层级比较高，相关联的干涉行为比较多，因此它被求值的概率较大，而且求值的结果表现出很大程度上的精确性（因为被多次求值或者传递了足够多的参数）。