C1P1.6 使用cond重新定义if

语言：Scheme

解释器：DrScheme

代码：\

#lang scheme

(define (new-if predicate then-clause else-clause)

(cond (predicate then-clause)

(else else-clause)))

(define (abs x)

(if (< x 0)

(- x)

x))

(define (square x)

(\* x x))

(define (average x y)

(/ (+ x y) 2))

(define (improve guess x)

(average guess (/ x guess)))

(define (good-enough? guess x)

(< (abs (- (square guess) x)) 0.001))

(define (sqrt-iter guess x)

(new-if (good-enough? guess x)

guess

(sqrt-iter (improve guess x)

x)))

(define (sqrt x)

(sqrt-iter 1.0 x))

(sqrt 3)

实验报告：\

这里的代码是上一节的牛顿的逐步逼近法求平方根写作Scheme的样子，其中的if被替代成了用cond过程实现的new-if。当试图运行此程序的时候，DrScheme死机了，为什么？

我一开始认为问题出在new-if里的sqrt-iter过程，事实上问题也确实出现于此，只是与我想象的不一样，我认为cond无法支持递归。

事实上，问题出在cond与if的区别：new-if是常规过程，会使应用序计算所有参数，所以使用递归时，会导致”无限递出”，即代换结束后总有需要再代换的东西(即sqrt-iter)。所以if或cond必须是特殊的，不然就无法实现递归过程，即要根据条件判断的结果决定计算哪部分参数。