

函数式编程

Haskell

✎ 修改

## 如何理解这两个函数的区别（Haskell）？✎ 修改

Let vs. Where - HaskellWiki 中提到了这两个fib函数的效率差别 fib = (map fib' [0 ..] !!)...

显示全部 ▾

关注问题

✎ 写回答

+ 邀请回答

💬 添加评论

🚀 分享

★ 邀请回答

🚩 举报

...

查看全部 5 个回答

 Felis sapiens ⭐

函数式编程、编程语言、编程 话题的优秀回答者

酿酿酿酿酿泉、祖与占、头顶青天红美铃等 14 人赞同了该回答

两个版本语义等价，后一个是前一个的 eta-expansion 的产物。

后一个版本每次 fib' 的调用会重新调用 fib，也就是重新打表，失去了记忆化的效果。不过 ghc 的 let-floating 优化能够搞定，你开 -O2 试试。（ghci 无优化）

Haskell-cafe 的类似讨论：[\[Haskell-cafe\] Eta-expansion destroys memoization?](#)

发布于 2017-05-21

▲ 赞同 14 ▾

💬 1 条评论

🚀 分享

★ 收藏

♥ 感谢

...

更多回答

 韩冬

俄耳甫斯的里拉琴 仍可照物的小铜镜

8 人赞同了该回答

这个事儿挺好理解的，如果不进行任何优化的话，第二个 fib' 处在一个函数体里面，将会被每次执行 fib 的时候重新动态定义（heap-allocation!）。而一个 fib' 不在，所以第一个 fib 函数执行的时候直接进入了 let ... in 后面的闭包里面求值去了。GHC 的 float-in/float-out 优化就是解决这个的。

当然因为 fib' 没有被浮出，导致了一个灾难性的后果就是 map 生成的列表没办法被记忆下来，于是就慢很多了。

写JS的时候就经常需要手动做闭包浮出优化（把一些被闭包捕获的变量变成参数，然后就可以把闭包定义的位置从函数体/循环体里移到外面了，从而避免重复定义）。

PS. 一个反直觉的事实是，GHC 不能自由地做 eta 转换，因为 f 可能是  $\lambda x. f x$ ，但  $\lambda x. f x$  就不是了（是一个实打实的可以进入的闭包）。

啊，可恶的  $\lambda$ 。

编辑于 2017-05-26

▲ 赞同 8 ▾

💬 2 条评论

🚀 分享

★ 收藏

♥ 感谢

...

 Kakyoin Noriaki

渔学家

1 人赞同了该回答

为什么要用list做记忆化.....list的随机访问是O(n)的啊。

关注者

61

被浏览

4,432

他们也关注了该问题



腾讯云新注册用户域名抢购1元起

.com

.cn

.xyz

腾讯云新用户域名抢购1元起



更有多种爆款组合选购



关于作者



Felis sapiens

⭐ 函数式编程、编程语言、编程 话题的优秀回答者

👤 电影旅行敲代码、Antokha Yuuki、暮无井见铃也关注了她

回答

268

文章

40

关注者

14,871

已关注

发私信

被收藏 4 次

编程语言与编译原理

酿酿酿酿酿泉 创建

5,338 人关注

PL

Isdsjy 创建

2 人关注

CS

2001 创建

1 人关注

技术

iintothewind 创建

0 人关注



对第二个函数每次调用都会重新计算一遍fib'。因为编译器不知道fib 1和fib 2中的fib'是不是一个东西。

不理解底层的渣渣不太理解你们Haskell为啥用了eta展开就没法做lambda lifting.....OCaml程序员用了一大堆eta展开也没啥效率问题啊。

==

展开阅读全文

赞同 1

5 条评论

分享

收藏

感谢

...

查看全部 5 个回答

相关问题

- Haskell 怎么推导函数类型呢？ 2 个回答
- Haskell 最有代表性的一段程序是什么？ 6 个回答
- Haskell 里面的 Functor 是个什么概念？ 6 个回答
- Haskell 有什么缺陷？ 8 个回答
- haskell中 foldr 与foldl的差别？ 4 个回答

相关推荐



森懂物理学：理解世界的极简指南  
共 31 节课 [试听](#)



C 语言解惑：指针、数组、函数和多文件编程  
5 人读过 [阅读](#)



刘看山 · 知乎指南 · 知乎协议 · 隐私政策  
应用 · 工作 · 申请开通知乎机构号  
侵权举报 · 网上有害信息举报专区  
违法和不良信息举报：010-82716601  
儿童色情信息举报专区  
电信与服务业务经营许可证  
网络文化经营许可证  
联系我们 © 2018 知乎