

Haskell

✎ 修改

Haskell有多少跟State/Reference有关的东西？✎ 修改

比如IO/IORef, State, ST/STRef, Lens等等。。。

✎ 修改

- 关注问题
- ✎ 写回答
- + 邀请回答
- 💬 添加评论
- 🚩 分享
- ★ 邀请回答
- 🚩 举报
- ...

关注者

51

被浏览

1,520

他们也关注了该问题



查看全部 1 个回答

 Felis sapiens ⭐

函数式编程、编程语言、编程 话题的优秀回答者

霜月琉璃、题叶、hhhhhhhhh、Bellev、圆角骑士魔理沙等 19 人赞同了该回答

实现 State/Reference 的方法分为 2 类：纯函数显式传递 state，或者使用 rts 提供的 state 相关性。

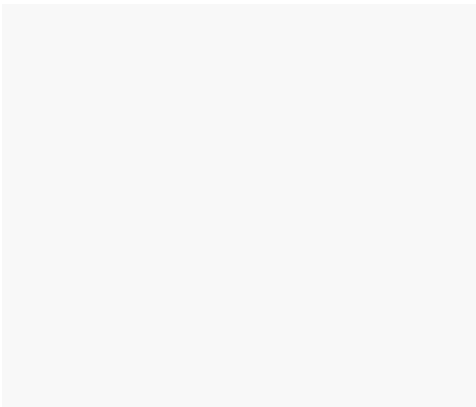
纯函数这个原理当然很简单， $s \rightarrow (a, s)$ 用表示结果类型为 a ，并且变更一个类型为 s 的状态的计算过程，这个在 transformers 库里有定义。State/StateT 可以通过 monad bind 将一系列状态类型为 s 的计算过程结合起来，最终输入初始状态可以获取计算结果和最终状态。

纯函数模拟的状态简单高效，但不能解决多线程共享状态和随机访问的问题，所以有时还是需要底层实现的 state。首先 ghc-prim 中提供了许多带副作用的指针/引用/数组的读写操作，这些带副作用的操作类型签名大致都是 $\text{State} \# s \rightarrow (\# \text{State} \# s, a \#)$ ，这个 $\text{State} \#$ 是 state token（比如 IO monad 对应的 state token 就是 RealWorld 类型，这个类型经过 codegen 以后不产生运行时表示，仅用于在 codegen 之前确定副作用的发生顺序），然后基于这些操作，可以封装出 IO/ST monad 中的带副作用读写。跟纯函数的版本相比，我们可以像操作 C 数组一样随机访问，可以调用 CAS 指令做原子操作，可以用各种 unsafeFreeze/unsafeThaw 将可变/不可变的数组相互转换，等等。

实现原理就以上两类，至于提供给用户的接口，比如 mtl 中的 MonadState，或者各种基于 free monad 的 effect system 中的 State，它们在同一个接口下可以切换不同的 State 实现。

P.S.

每次调用 ghc-prim 里带副作用的 primop，手动传递 RealWorld token 时，感觉就像



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

241

文章

40

关注者

14,871

已关注

💬 发私信

被收藏 2 次

haskell

0 人关注

ptc 创建

大概会是这样

0 人关注

sophomore2 创建

相关问题

如何理解Haskell中的Arrow及其用途? 2 个回答

Haskell 的 Typeclass 怎么理解? 14 个回答



发布于 2017-05-15

赞同 19



2 条评论

分享

收藏

感谢



收起 ^

查看全部 1 个回答

为什么业界很少使用 Haskell? 5 个回答



Haskell中如何代价最小的操作（插入、删除）一棵树? 4 个回答

有没有比较实用、成熟的 Haskell 应用? 8 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



数学妙啊！妙！

张英锋 等

289,069 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

