



红尘里的Haskell（之二）——老司机带逛Hackage

Felis sap... 

函数式编程、编程语言、编程 话题的优秀回答者

已关注

开源哥、圆角骑士魔理沙、祖与占、刘雨培、草莓大福等 160 人赞了该文章

本期红尘里的Haskell内容：带逛Hackage~

标准库

- base：几乎所有库的起点。包含了一个基本像样的标准库Prelude，实现了Haskell 2010标准规定的模块，同时附加了不少GHC扩充的模块，如多线程、异常处理、FFI、parsing/printf组合子、一些type level编程所需的定义，等等。是每一名Haskell程序员都应该非常熟悉的库。

时间与日期

- time：ghc自带的日期/时间库，对不同日期/时间类型有一定的type level区分。
- clock：一个高精度的时间库。

数据类型

- bytestring：正如其名，一个immutable bytestring的高性能实现。分为strict（单一buffer）/lazy（块状链表）版本，同时提供了一个builder的API，可以用于快速拼接大bytestring。
- text：一个基于bytestring的高性能Unicode字符串类型，性能比标准库里的String（[Char]）更佳。

新Hask

 赞同 160 31 条评论 分享 收藏

...

仙境里的Haskell（之八）—— M

- containers: 基于BST (Binary Search Tree) 的Map/Set、基于Finger Tree的Sequence、基于Patricia Trie的IntMap/IntSet。相关操作的时间复杂度详见文档。
- hashable/unordered-containers: 基于HMAT (Hash-Mapped Array Trie) 的HashMap/HashSet。一般性能高于containers的Map/Set, 推荐使用。
- hashtables: 当需要mutable的Hash Table时, 推荐使用这个库。有3种不同Hash策略可选, 底层使用C实现。
- dlist: 一种特殊的List结构, 内部基于List的CPS形式, 可以O(n)时间内与普通List相互转换, O(1)时间拼接。因为O(1)拼接的特性, 非常适合用于writer monad的输出类型。
- fgl: Mark Erwig的Inductive Graph库, 多种图的表示与图论算法实现。需要表示图时推荐使用这个库, 而非containers中的Data.Graph模块。

二进制序列化

- binary/cereal: 两个常用的序列化库, 能够定义在数据与bytestring之间的Get/Put操作。前者是ghc boot library, 同时性能逊于后者。cereal的一个衍生项目是safecopy, 定义了带版本控制的序列化操作。
- store: 一个最新的序列化库, 牺牲跨平台性换取性能, 同时自带能为用户自定义数据类型进行序列化的Template Haskell实现。推荐使用。

文本解析

- attoparsec: 支持解析ByteString/Text的parsing组合子库, 支持回溯操作 (有Alternative instance), 注重性能。
- parsers: 一个parsing组合子抽象库, 定义了若干个类型类 (Parsing/CharParsing/TokenParsing/LookAheadParsing), 实现这些类的实例可以获得诸如lexing、permutation parsing、expression parsing等额外功能。parsers有多个后端, 包括base中的ReadP、attoparsec、parsec和trifecta。
- trifecta: 另一个parsing组合子库, 一大特色功能是生成可读性好的带高亮错误信息。
- megaparsec: parsec的现代fork, 与attoparsec/trifecta相比, 支持自定义错误类型、作为monad transformer使用, 同时自带了parsers额外功能的实现。
- Earley/pinchot: 支持任意CFG语法的解析。与alex/happy不同的是, 无需用预处理器生成难以调试的Haskell代码, 仍然有一个好用的嵌入式DSL可以指定目标语法。
- boomerang: 可以通过一套代码, 同时定义出一个语法的parser和pretty-printer。
- polyparse: 这个库收集了许多种支持不同特性的parsing组合子, 对于写parsing组合子的初学者, 实现颇有学习价值。

正则表达式

- pcre-he

▲ 赞同 160 ▼

● 31 条评论

➤ 分享

★ 收藏

...

仙境里的Haskell (之八) — M

- [ansi-wl-pprint](#): 这个pretty-printer库的特色是支持生成彩色的Doc并打印到终端！（底层由ansi-terminal支持）

命令行参数解析

- [optparse-applicative](#): 特性丰富的命令行参数解析库。另外还有[optparse-generic](#)库，对于Generic类的实例类型可以自动生成一个对应的命令行参数parser。

Shell编程相关

- [shelly/turtle](#): 实现了带类型shell编程的框架。与传统的shell scripting相比可维护性肯定是提高不少，不过是否过多牺牲了灵活性和开发速度，大家不妨自己试用然后得出结论。
- [shake](#): 一个用于实现build system的框架，设计目标是取代GNU make。
- [ShellCheck](#): 面向传统shell script的静态分析/查错工具。

JSON/YAML/XML/CSV

- [aeson](#): Haskell生态系统中的JSON事实标准库。支持用Generic/Template Haskell为用户自定义数据类型生成JSON范式。
- [yaml](#): 处理YAML格式的库。复用了aeson中的类型，因此可以看作是“带注释的JSON语法糖”（
- [taggy](#): 解析与表示XML/HTML的库。配合[taggy-lens](#)使用效果最佳，这个比传统的XPath/CSS Selector可读性和效率都更好。
- [cassava](#): 处理CSV格式的库。

图像I/O

- [JuicyPixels](#): 支持多种图像格式的读写。原生Haskell实现，没有C依赖。

矢量图绘制

- [diagrams](#): 绘制矢量图/动画的DSL。可以绘制到多个backend: SVG/HTML5 Canvas/Cairo/PostScript等。
- [gloss](#): 基于OpenGL的矢量图绘制，与diagrams相比特性更简单，开发速度更快。

- [pandoc](#)：实现了一个加强版的markdown语法以及支持与多种文档格式相互转换的文档类型。

数据库相关

- [persistent](#)：实现了一个用于描述Haskell数据类型序列化的DSL，并支持诸如PostgreSQL/SQLite/Redis/MongoDB等后端。
- [acid-state](#)：为任意支持cereal序列化的数据类型实现一个满足ACID性质的简单数据库，用于少量数据的持久化非常适合。
- [TCache](#)：作用与acid-state类似，但是基于Key-Value数据库，并且支持多后端。在数据量较大而又不需要SQL schema约束时比较适合。

随机数生成

- [mwc-random](#)：高质量的随机数生成库。随机数的发生需要在PrimMonad中执行；支持种子的freeze/load操作。
- [MonadRandom](#)：mtl-style的monad transformer class，支持随机数生成。底层基于标准库System.Random的实现，随机数种子bit较少，与mwc-random相比，用随机数质量换取pure的界面（种子可以很方便地传值，不需要在PrimMonad里freeze）

日志库

- [monad-logger](#)：功能齐备的日志库。需要在带MonadIO支持的monad中使用。
- [monad-journal](#)：功能较简单的日志库，与普通的writer monad相比增加了读取/清除日志的功能。支持pure的界面。

流式I/O库

- [pipes](#)：流式I/O库，与conduit相比，语义更简明（有更好的equational reasoning），文档更加清晰。
- [conduit](#)：最常用的流式I/O库。与pipes相比，资源管理/异常处理更加完备。
- [machines](#)：一个支持多个upstream合并的流式I/O库。

模板库

- [mustache](#)：mustache模板的Haskell实现。
- [shakes](#)

加密/解密

- cryptonite: Haskell社区的加密/解密算法事实标准库。

Web客户端

- http-conduit: 基于http-client的Web客户端库，支持将response body转为conduit中的producer，实现流式读取。
- wreq: 同样基于http-client，不支持conduit界面，不过有cookie persistence/OAuth支持/基于lens的API等特色功能。不需要流式I/O时，推荐使用这个库。
- webdriver: Selenium的Haskell客户端，便于写headless browser test。

Web框架/服务器相关

- servant: 闪瞎钛合金狗眼的Web框架，基于一个type level的DSL规定api endpoint，并可基于类型自动生成文档、其他语言客户端代码等等。用于实现restful服务非常适合。
- yesod: 可与Python的Django框架类比，大而全的Web框架，batteries included。
- scotty: 小型的Web框架，需要极短时间内架设简单站点时不妨使用这个。
- wai: 大多数Web框架底层基于wai (Web Application Interface)，规定了一个web application的类型，基于这个类型可以实现standalone的web server，或者叠加一系列中间件。
- websockets: WebSockets协议的server/client库。
- warp: 原生Haskell实现的standalone web server，功能齐全（在nginx之前就有了完整HTTP/2支持~）

测试

- QuickCheck: 大名鼎鼎的property-based testing框架，需要测试Haskell函数的性质时，可以自动生成随机测例进行测试，找到反例时自动收缩到较小反例，按Matt Might说法：投入20%的时间获得形式验证80%的效果。
- smallcheck: 与QuickCheck类似，生成测例的策略是穷举较小范围内的所有输入。
- hspec: 功能丰富的单元测试框架。

性能测试

- criterion

- [ekg](#): 实时监测Haskell服务器各种数据（尤其是空间占用。。）
- [threadscoope](#): 可视化分析ghc eventlog，用于调试与并行/并发相关的性能问题。

数值计算相关

- [arithmoi](#): 数论相关算法库。
- [hmatrix](#): 基于blas/lapack的矩阵运算库。
- [linear](#): 线性代数抽象库。
- [statistics](#): 统计学算法库。
- [ad](#): 自动微分算法库。
- [integration](#): 数值积分库。

并行/并发相关

- [yarr](#): 类似repa的并行数组库，继承了repa的shape polymorphism和fusion等特性。
- [monad-par](#): 数据流并行库，支持pure/带IO的fork/join操作实行并行计算，无需考虑调度策略。
- [async](#): 高层次的线程抽象，有妥善的错误处理机制。
- [stm](#): STM（软件事务内存）库。与传统的基于锁的并行程序相比，STM的优势在于程序的可组合性强，不用担心复杂逻辑带来的死锁问题。

分布式编程

- [distributed-process](#): Cloud Haskell的核心抽象，实现支持远程调用的Process monad。具体使用方法参见Cloud Haskell系列库的文档。
- [transient](#): 一个新兴的分布式计算库，特色是支持使用GHCJS编译，组成浏览器/服务器的异构计算框架。

第三方Prelude

- [classy-prelude](#): 开发大型Haskell项目常见的问题是Prelude功能不足，需要大量手动的qualified import、lifting等boilerplate work。classy-prelude很好地解决了这个问题。

macro省去许多。

- c2hs: 与bindings-DSL类似。

窗口管理器

- xmonad: 基于X11的独立窗口管理器，可以用Haskell代码实现复杂特性的拓展。

游戏开发相关

- sdl2: sdl2的绑定库。这个是目前做得最完善的游戏引擎绑定，文档较全，而sdl2本身除渲染外也有音频支持。
- GLFW-b: glfw3的绑定库。
- GLUT: freeglut的绑定库。
- OpenGL: opengl的绑定库。

FRP引擎

- reactive-banana: higher-order的FRP引擎，杜绝了常见的空间泄漏，文档完善。
- Yampa: 基于Arrow (Signal Function) 的FRP引擎。Yale Haskell实验室出品。
- reflex: 类似reactive-banana，特色是GHCJS支持较好。

数据类型泛型 (Datatype Generics) 相关

- syb: 具体可查阅syb文档，与论文《Scrap Your Boilerplate》
- uniplate: syb的高性能替代实现。
- generics-sop: 一个基于Sum-Of-Products抽象的generics库，与普通的binary sum/product相比，SOP更加符合直觉，且容易生成性能更佳代码。

type level专场

- singletons: 使用GHC高级特性进行有限的依赖类型编程。
- dimensional: 由类型系统进行静态检查的单位系统，需要表达度量衡时推荐使用。

- 现。与extensible-effects相比，性能稍差但更优雅（移除了effect的Typeable限制）
- IOSpec：基于free monad的测试框架。

形式化方法

- sbv：基于SMT求解器的形式化验证库，API与QuickCheck类似，且支持生成smtlib格式，调用多个后端。
- ersatz：使用Haskell代码表达SAT/QSAT约束。求解器为原生Haskell实现，无需调用minisat之类的后端。

Lambda calculus专场

- bound：generalized de bruijn indices的实现。
- unbound-generics：基于GHC Generics实现的locally nameless substitution。

底层相关

- ghc-prim：需要使用GHC底层黑魔法时，焚香沐浴，然后import GHC.Prim。
- primitive：PrimMonad的实现，以及多种ghc-prim中的底层类型的封装（Addr、Array/ByteArray/MutVar等）
- memory：包含多种底层内存操作的接口、Hashing算法实现、bytearray类型。
- unix/Win32：操作系统API的绑定库。
- network：几乎所有网络相关库的依赖。需要进行socket级别网络编程时使用。

Haskell工具链科普篇和Hackage带逛篇告一段落。从下期开始，转向专题化讲解，每个专题引入一个概念，讲解其来源和实用意义，然后示范简单的实现方式，最后指向对应的成熟Haskell库。这个想法启发自@刘鑫老师的各种语言花式造parsec的文章。下期暂定先做“基于属性的单元测试”吧。发稿时间不定，如果大家有特别感兴趣的专题也欢迎评论区留言。Adieu~

编辑于 2016-08-03

Haskell

函数式编程



魔鬼中的天使

主要会讨论关于函数式编程 (haskell、scala) 的内容。我会尽力讲的清晰明了，带你...

关注专栏

推荐阅读

合 Haskell 用家的 Idris 入坑指南

没有 Cabal 或者 Stack 这样的包管理器，也没有 Idris Platform 之类的东西，到官网下二进制包解压完就可以了。然而，Idris 确实有「包描述文件」ipkg。安装网上的包的时，你需要把包下载下...

elle...

发表于雾雨魔法店



红尘里的Haskell (之一) —— Haskell工具链科普

Felis...

发表于魔鬼中的天...



幻想中的Haskell - Compiling Combinator

圆角骑士魔...

发表于雾雨魔法店

学H

几年
滴ler
开G
FTP
Fold
然了

Cosm

31 条评论

⇌ 切换为时间排序

写下你的评论...



Ivan Yang

2 年前

这就是你不动点高校太监了的理由？（逃

👍 2



Felis sapiens (作者) 回复 dram

2 年前

之后可能会有专题讲parsing组合子，不过不会针对trifecta的吧。。直接看文档应该够了。trifecta有不少功能其实是在parsers里面的。。另外可以参考下idris的parsing部分，用的是trifecta

👍 1

查看对话

以上为精选评论 ?



祖与占

2 年前

先赞为敬!

👍 赞

赞同 160



31 条评论

分享

★ 收藏



仙境里的Haskell (之八) —— M

无膜为敬！

👍 赞



Felis sapiens (作者) 回复 Ivan Yang

2 年前

一言难尽，一言难尽。。（

👍 3 💬 查看对话



不动点帕琪

2 年前

先拜为敬！

👍 赞



大石匠

2 年前

答主，功德无量

👍 赞



祖与占

2 年前

第三方prelude加个protolude, 其他还有很多选择（

👍 赞



Felis sapiens (作者) 回复 祖与占

2 年前

什么base-prelude或者mtl-prelude之类的想了想还是省了，太轻量级了（

👍 赞 💬 查看对话



圆角骑士魔理沙

2 年前

先抱为敬！

👍 赞



Vani

2 年前

期待专题讲座。XD

👍 赞



nullday

2 年前

distributed-process 的地址失效了

👍 赞



Va

2 年前

▲ 赞同 160



💬 31 条评论

➦ 分享

★ 收藏



仙境里的Haskell (之八) —— M



赞



Felis sapiens (作者) 回复 圆角骑士魔理沙

2 年前

莎莎么么哒



赞



查看对话



Felis sapiens (作者) 回复 Vani

2 年前

这个因人而异吧，简单的教程还是有很多人需要的



赞



查看对话



Felis sapiens (作者) 回复 nullday

2 年前

fixed，感谢提醒



赞



查看对话



Vani 回复 Felis sapiens (作者)

2 年前

就是感觉现在简单的教程太多了而已，稍微深一点的就找不到资料了。（耸肩



赞



查看对话



祖与占 回复 Vani

2 年前

只是稍微深的资料你看不懂（耸肩



赞



查看对话



nullday 回复 Felis sapiens (作者)

2 年前

没事，因为自己也在做haskell 的计算框架，所以就点了一下。233333



赞



查看对话



Vani 回复 祖与占

2 年前

所以需要老司机们来循循善诱啊（扶额



赞



查看对话