

编程Haskell编译原理编译器词法分析✎修改

关注者76被浏览8,577

Swift, Haskell这种可以自定义运算符的语言（不仅仅是重载），在实现编译器时跟其他语言又什么区别吗？✎修改

他们也关注了该问题



✎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

...

查看全部 9 个回答

 Felis sapiens  函数式编程、编程语言、编程 话题的优秀回答者

彭飞、草莓大福、开源哥等 14 人赞同了该回答

Swift不清楚。。

Haskell里运算符重载实际上就是type class的意思，用户自定义类型如果是某个class的instance，实现了这个class的函数，那么在所有需要用到这个class下类型的地方，都可以使用这个自定义类型了。比如自定义类型是Show的instance，实现了show函数，就可以print，是Eq的instance，实现了(==)，就可以比较等于/不等于，是Ord的instance，实现了compare，就可以比较大小关系。

编译器实现type class的方法（假设通过了类型检查），有static monomorphization，intentional type analysis和dictionary passing等方法。

dictionary passing举个例子，对于一个类型签名为Show a => a -> ...的函数，替换成一个(a -> [Char]) -> a -> ...的函数，这里额外多传进去的这个(a -> [Char])的东西就是show函数的实现；对于更复杂的、有多个函数的class，就改成先传一个包含相应函数实现的dictionary进去。这种方法跟C++里对象的成员函数隐式地多传一个this指针有异曲同工之妙。

static monomorphization是将所有用到Show a => a -> ...的函数的地方全部静态改写成a -> ...，针对不同的show生成不同版本的函数，类似于C++的template instantiation。。intentional type analysis就直接是运行时判断类型，然后dispatch到相应的show函数。一般而言运行时判断的方法会牺牲一些性能，但是方便支持seperate compilation等特性；编译期展开的方法生成代码性能更好，但不能实现separate compilation（想想C++里那些header-only的模板库），对类型系统有更多限制。

编辑于 2015-01-13

▲赞同 14▼

●4 条评论

🚩分享

★收藏

❤感谢

...

更多回答

 罗宸 你说好代码相见恨晚，我说造轮子你不够勇敢

10 人赞同了该回答


其实Haskell的处理逻辑比你想象的要简单得多得多，一个东西是不是中缀运算符可以说从词法上就已经确定了。

比如你都不用看函数定义，就直接看某个表达式，比如 a <+> b，你就知道这是『identifier』『infix operator』『identifier』了，这个过程中是直接把某些标点符号的序列识别成中缀运算符的tokens的，稍微特殊一点的就是back quote括起来的identifier也会被当作中缀运算符。

当然在语法分析阶段，由于要考虑算符优先级和结合性，所以需要提前扫一遍找到所有infix语句。



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624文章40关注者14,871

已关注发私信

被收藏 2 次

Programming Languages 3,642 人关注 彭飞 创建

FP 1 人关注 sepiggy 创建

相关问题

各个编程语言都有哪些令该语言的编译器/解释器开发者想掀桌的地方？ 10 个回答

Swift就不清楚了。

不过有类似能力，甚至更强能力的语言有很多，比如Coq，Agda，Idris都具备自定义语法的能力。例如我之前的回答：

从阅读代码上考虑，if里的
的">"、"<"用哪个更容易看懂？
[展开阅读全文](#)



[赞同 10](#) [1 条评论](#) [分享](#) [收藏](#) [感谢](#) ...

祖与占
函数式编程 话题的优秀回答者

5 人赞同了该回答

The GHC parser has lots of hacks . For example, have you ever wondered how it parses fixities even though you can specify them in arbitrary places (e.g. inside a where clause)? It actually parses everything with the same fixity and precedence and then rebalances the trees in the Renamer (!)

[展开阅读全文](#)

[赞同 5](#) [2 条评论](#) [分享](#) [收藏](#) [感谢](#) ...

[查看全部 9 个回答](#)

请问大家了解函数式语言编译器的实现技术嘛？ 7 个回答

为什么尾递归优化似乎不常出现在非函数式语言的解释器、编译器中？ 9 个回答

为什么不给Python 这样的解释语言写一个编译器？ 7 个回答

Haskell的编译器GHC是Haskell写的吗？ 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](#)

如何科学地学语言
青格乐
★★★★★ 9092 人参与

函数式 Swift
Chris Eidhof (克里斯·安...)
63 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