本文介绍了如何使用 C# 实现一个简化 Scheme——iScheme 及其解释器。

http://zh.lucida.me/blog/how-to-implement-an-interpreter-in-csharp/#more

实现以下目标：

* 实现基本的词法分析，语法分析并生成抽象语法树。
* 实现嵌套作用域和函数调用。
* 解释器的基本原理。
* 以及一些 C# 编程技巧。

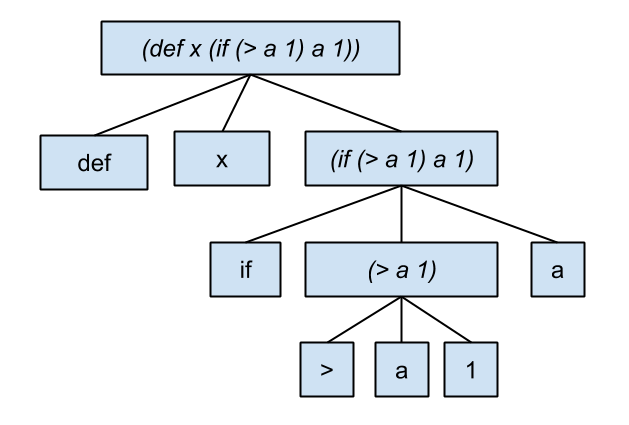
从头开始构造 iScheme 的解释程序。

**解释器构造**

iScheme 解释器主要分为两部分，解析（Parse）和求值（Evaluation）：

* 解析（Parse）：解析源程序，并生成解释器可以理解的中间（Intermediate）结构。这部分包含词法分析，语法分析，语义分析，生成语法树。
* 求值（Evaluation）：执行解析阶段得到的中介结构然后得到运行结果。这部分包含作用域，类型系统设计和语法树遍历。

程序即语法树



### 词法分析

词法分析负责把源程序解析成一个个词法单元（Lex），以便之后的处理。

iScheme 的词法分析极其简单——由于 iScheme 的词法元素只包含括号，空白，数字和变量名，因此C#自带的 String#Split 就足够。