# 跟vczh看实例学编译原理——一：Tinymoe的设计哲学

## 系列文章的目标

词法分析

语法分析

语义分析

符号表

全文CPS变换

编译生成高效的其他语言的代码

编译生成自己的指令集

带GC的虚拟机

类型推导（intersection type，union type，concept mapping）

跨过程分析（inter-procedural analysis）

## Tinymoe设计的目标

处理Tinymoe需要的知识点多，用于编译原理教学

容易嵌入其他语言的程序，可以调用宿主程序提供的功能

语法优化于描述复杂的逻辑，而不是优化与复杂的数据结构和算法（虽然也可以）

动态类型语言

## Tinymoe的设计哲学

小规模的语言核心+大规模的标准库

扩展后的东西跟原生的东西外观一致

## Tinymoe的实现难点

需要三次Parsing操作，动态生成文法，但是仍然按照手写递归下降的方法来写

CPS变换后没有特殊处理直接执行容易导致callstack积累的没用的东西过多

动态类型语言的类型推导

跨过程分析和对程序的控制流的化简（譬如continuation转状态机等）