# 跟vczh看实例学编译原理——三：Tinymoe与无歧义语法分析

文章中引用的代码均来自<https://github.com/vczh/tinymoe>。

看了前面的三篇文章，大家应该基本对Tinymoe的代码有一个初步的感觉了。在正确分析“print sum from 1 to 100”之前，我们首先得分析“phrase sum from (lower bound) to (upper bound)”这样的声明。Tinymoe的函数声明又很多关于block和sentence的配置，不过这里并不打算将所有细节，我会将重点放在如何写一个针对无歧义语法的递归下降语法分析器上。所以我们这里不会涉及sentence和block的什么category和cps的配置。

虽然“print sum from 1 to 100”无法用无歧义的语法分析的方法做出来，但是我们可以借用对“phrase sum from (lower bound) to (upper bound)”的语法分析的结果，动态构造能够分析“print sum from 1 to 100”的语法分析器。这种说法看起来好像很高大上，但是其实并没有什么特别难的技巧。关于“构造”的问题我将在下一篇文章《跟vczh看实例学编译原理——三：Tinymoe与有歧义语法分析》详细介绍。

在我之前的博客里我曾经写过《[如何手写语法分析器](http://www.cppblog.com/vczh/archive/2008/06/15/53373.html)》，这篇文章讲了一些简单的写递归下降语法分析器的规则，尽管很多人来信说这篇文章帮他们解决了很多问题，但实际上细节还不够丰富，用来对编程语言做语法分析的话，还是会觉得复杂性太高。这篇文章也同时作为《[如何手写语法分析器](http://www.cppblog.com/vczh/archive/2008/06/15/53373.html)》的补充。好了，我们开始进入无歧义语法分析的主题吧。