语言处理程序分为两大类：翻译程序和解释程序。

**翻译程序**的工作方式是把程序设计语言降低到机器水平，即把某一程序设计语言所写的程序(称为源程序)翻译为机器语言程序（称为目标程序），然后由计算机直接执行目标程序。

当程序语言为汇编语言时，翻译程序称为**汇编程序**。

当程序语言为高级语言时，翻译程序称为**编译程序**。

**解释程序**的工作方式是把程序设计语言看作解释器本身的语言。解释器运行时同时取得源程序和输入数据。

翻译方式的程序执行效率比解释方式的要高。解释方式的灵活性要比翻译方式高。



翻译程序和解释程序是可以相互结合运用，如Java源程序就是先通过**编译程序**编译为以java虚拟机的语言——BitCode表示的程序，然后通过不同平台上的Java虚拟机**解释执行**。



有限自动机是更一般化的状态转化图。分为**确定有限自动机**（DFA）和**不确定有限自动机**（NFA）。

**应该如何理解“上下文无关文法”？**

抛砖引玉，试着用汉语来解释一下。比如这个帖子（[用“本来”一词造句，看看谁造的比较有水平。](//link.zhihu.com/?target=http%3A//bbs.hupu.com/6161440.html)）里面让大家用本来造句，结果有这样的句子：  
  
本来这个进球就是违例的，但你不肯承认也没办法  
我有一本来自美国的花花公子杂志  
拿我的笔记本来  
  
如果汉语是上下文无关文法的话，那我们任何时候看见“本来”两个字，都可以把它规约为一个词。可惜汉语不是上下文无关文法，所以“本来”能否归约为一个词，要看它的上下文是什么。上面的三个例子中，第一句里的“本来”可以规约为一个词：  
  
（（本来）（（（这个）（进球））（就）（是）（违例的））），（（但）（（你）（不肯）（承认）（也）（没办法）））  
  
但后面两句都不行。后面的两句大约应该这样规约：  
  
（我）（有）（（（一本）（来自）（美国）（的））（花花公子杂志））  
（拿）（（我的）（笔记本））（来）  
  
看到问题突然想到的思路，不一定对。可以进一步参考这个[Context-free grammar](//link.zhihu.com/?target=http%3A//en.wikipedia.org/wiki/Context-free_grammar)，[Context-sensitive grammar](//link.zhihu.com/?target=http%3A//en.wikipedia.org/wiki/Context-sensitive_grammar)。总之，我觉得不从学编译的角度思考，而是从自然语言的角度思考更容易理解一些。

上下文无关文法（Context-Free Grammar）就是下推自动机（Pushdown Automata）能够识别的文法。跟有限状态自动机（DFA/NFA）相比，多了个栈，有了栈就有了递归能力，可以识别括号表达式（DFA/NFA没有递归能力）