## 编译原理开题报告

# 设计课题的目的及意义

### 设计课题的目的：

编译原理作为计算机专业的一门重要的必修课程，但是由于这门课程的理论性过强，原理概念抽象且复杂，导致大多数学生望而却步，没有兴趣和动力去学习这门课程。编译原理的课程内容即涉及复杂深奥的理论原理知识，也有大量的算法实现细节。语言和文法，词法分析，语法分析，属性文法和语言制导，语义分析和中间代码产生，代码优化及目标代码生成是其主要内容。枯燥的理论知识加上传统的学习模式很容易使学生失去学习这门课的信心和兴趣，因此急需一套编译原理核心算法的示教软件，将编译原理的主要步骤及核心算法用可视化界面演示出来，让学生能够直观的了解计算机的编译器将源代码编译解析为目标代码的过程。本课题的目的是为学生学习编译原理提供帮助，降低学生学习编译原理的难度，激发学生学习编译原理的兴趣，提高学生的动手实践能力及问题分析能力。

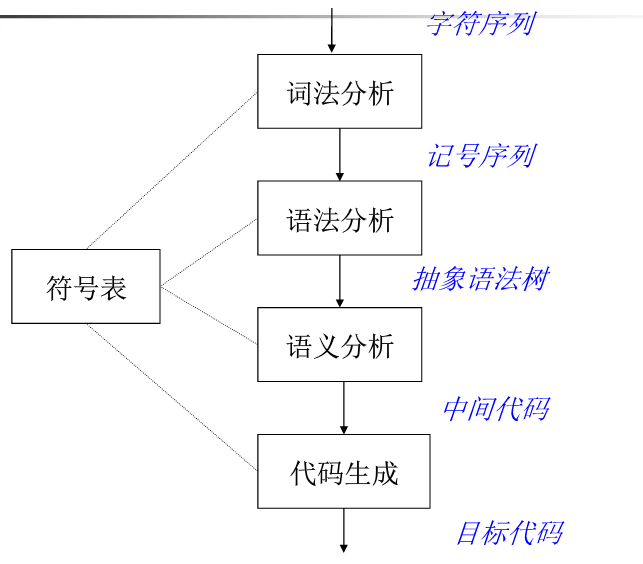
设计课题的意义：

在演示设计一个简单的程序语言的编译器的过程中，让学生能够更加深刻的理解词法分析器，语法分析器，语义分析器与中间代码生成器，代码优化器和代码生成器的构造原理和实现技术，提高学生对成程序设计语言的理解和正确使用程序设计语言的能力，在以后的学习工作中，提高开发大型软件的能力，提高抽象思维能力及形式化描述能力；掌握编译技术有助于加深学生对计算机的组织结构，操作系统及指令系统的理解；编译器作为操作系统最基本的系统软件，对本身的性能要求较高，因此涉及到一些经典的算法。这些算法，设计模式和实现思想不仅可以应用到其他软件的开发，还能对其他问题的解决提供启发。

# 预期目标和成果形式

# 预期目标：

本软件模块按照编译原理的模块来划分，主要分为“词法分析”，“语法分析”，“语义分析和中间代码产生”，“代码优化”和“目标代码生成”。预期目标是完成主要模块的核心算法及其可视化界面演示。编译器结构如下



词法分析：输入字符序列，输出字符序列

语法分析：输入记号序列（单词流），定义语法规则，输出抽象语法树

语义分析：更具抽象语法树生成中间代码

代码生成：输入中间代码，经过优化生成目标代码

## 成果形式

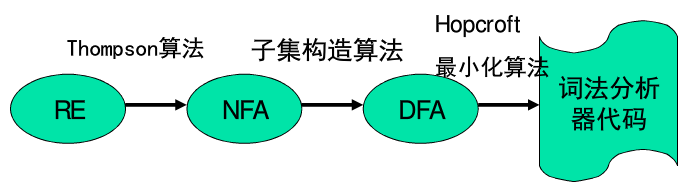
使用java语言的GUI组件完成词法分析，语法分析，语义分析及中代码生成，目标代码生成等模块的核心算法的动态显示。

需要完成的模块及希望达成的成果形式如下

# 拟采用的设计方法：

主要采用的设计方法就是各模块原理+核心算法+可视化界面三部分结合，从整体结构到具体细节演示此软件的工作流程。

词法分析器：读入源程序输入的输入字符流，按照定义的词法规则输出记号流（字符串流）。由于目前网络上已有成熟的词法分析器，所以不采取手工定义词法分析器的方式，采用自动的词法分析器（flex,flex,lex等），主要工作是采用正则表达式，定义声明式规则，在根据正则表达式构造有限自动机。其中构造确定的有限自动机分为三个步骤，每个步骤对应一个算法，本软件将详细演示此过程，流程图如下

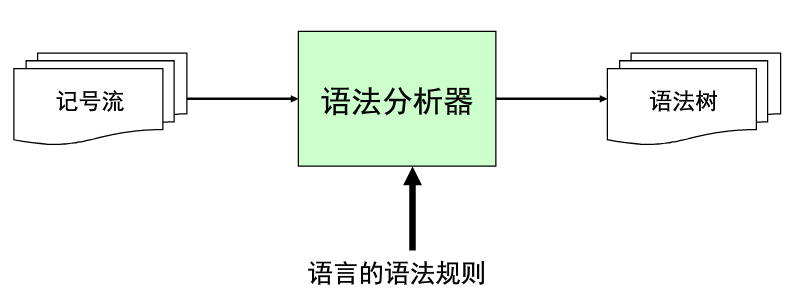


1）正则表达式（RE）-->非确定的有限自动机（NFA） 采用的是Thompson算法

2）非确定的有限自动机（NFA）->确定的有限自动机（NFA） 采用的是子集构造算法

3）确定的有限自动机（DFA）->最小化的DFA 采用的是Hopcroft最小化算法

语法分析器：语法分析是编译过程的核心部分，其任务是在词法分析识别出单词流的基础上，分析并判定程序的语法结构是否符合语法规则。语言的语法结构是根据上下文无关无法（用四元组G=(T,N,B,S)来表示）定义。因此语法分析的工作原理就是根据文法产生式，读入词法分析器生成的记号流，检查是记号流是否为一个句子，满足的话，输出抽象语法树。其工作结构图如下：



语法分析器分为两类分别为

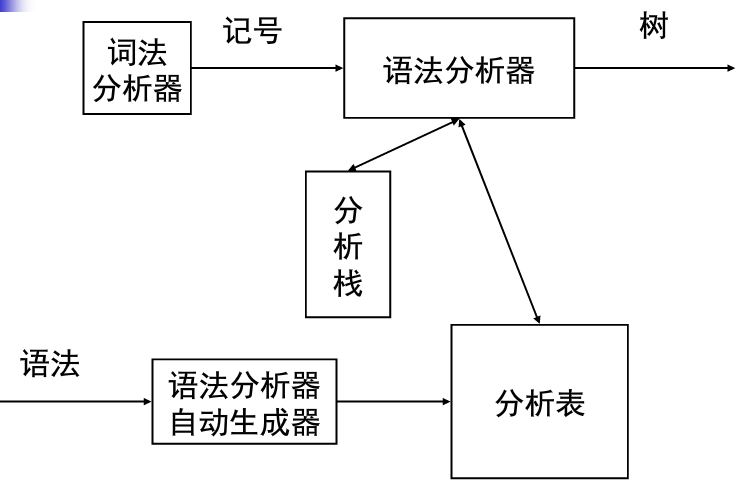
1）自上而下分析

1>预测分析算法（递归下降分析法）

原理：采用回溯的思想，减少算法中重复的次数

2>LL（1）分析算法

原理：消除左递归，找到克服回溯的充分必要条件，采取表驱动的分析算法，原理图如下：



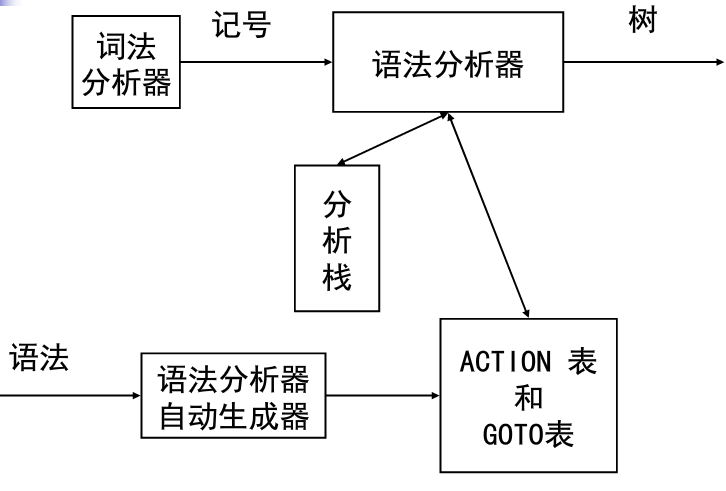
2）自下而上分析 ：

1>算符优先分析法 原理：定义算符之间（终结符）之间的某种优先关系，借助于这种优先关系寻找“可规约串”，进行规约。

2>LR分析法

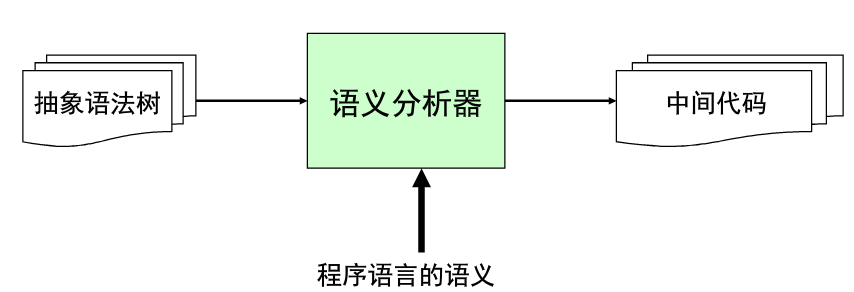
原理：规范规约（最左规约-最右推倒的逆过程）的关键问题是寻找句柄。LR的基本思想是，在规范规约的过程中，一方面记住已移进和规约出的整个符号串，另一方面根据所用的产生式推测未来可能会碰到的输入符号。总结：记住“历史”，展望“未来”。

原理图如下：

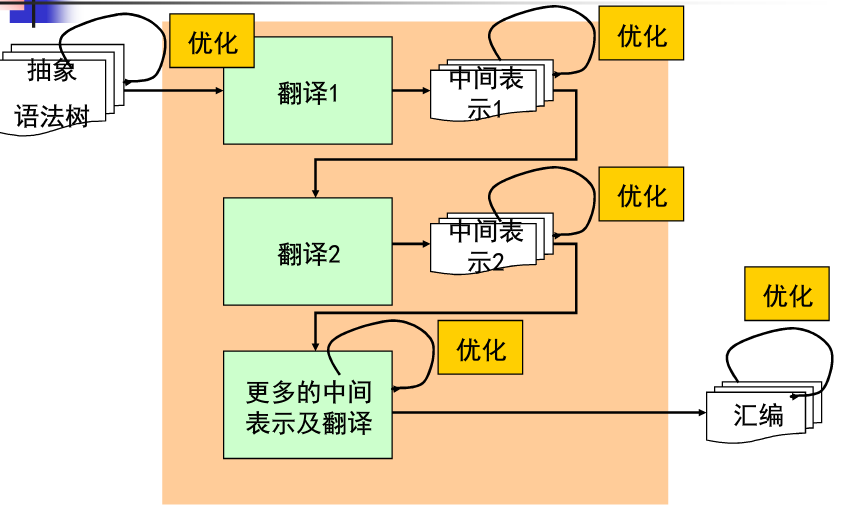


语义分析及中间代码生成：使用抽象语法树和符号表中的信息，依据语言定义来检查语义的一致性，以保证各部分能有意义的结合在一起，实现一种抽象的程序。此部分主要演示中间语言的几种形式，如后缀式，三地址代码，四元式。其主要任务是检查程序（抽象语法树）的上下文相关的属性

其工作结构图如下：



代码优化及目标代码生成：对生成的中间代码进行优化，产生较好的目标代码。代码优化是对被优化的程序进行的一种语义保持的变换。语义保持是让程序的可观察行为不能改变；变化的目的是让程序能够比变换前：更小，更快，cache行为更好，更节能，等等。代码优化的位置图如下：



本软件结合自身能力和时间要求主要完成词法分析器和语法分析器，其他部分尽力完成。

# 开发环境及工具：

操作系统 windows7

开发工具 eclipse4.5 javaSE

# 进度计划

2017年12月1日-2017年12月15日 完成开题报告

2018年1月1日-2018年3月31日 公司实习，查阅课题资料，了解编译原理各模块的主要原理

2018年4月1日-2018年4月28日 实现编译原理各模块的核心算法，完成部分论文

2018年5月1日-2018年5月15日 完成编译原理的核心算法的动态可视化演示工作，完成部分论文

2018年5月16日-2018年6月 完成系统整合，完成论文初稿

# 参考文献：

【1】《编译原理教程》（第四版）主编：胡元义 出版社：西安电子科技大学出版社

【2】《编译原理》作者：胡伦俊、徐兰芳、骆婷 出版社：电子工业出版社

【3】《程序设计语言编译原理》作者：陈火旺、刘春林、谭庆平 出版社：国防工业出版社

【4】《编译原理》 作者：[lfred V. Aho](https://book.douban.com/search/Alfred%20V.%20Aho)、 [Monica S.Lam](https://book.douban.com/search/Monica%20S.Lam) 、 [Ravi Sethi](https://book.douban.com/search/Ravi%20Sethi) 、 [Jeffrey D. Ullman](https://book.douban.com/search/Jeffrey%20D.%20Ullman) 出版社：机械工业出版社

【5】《算法导论》作者：[Thomas H.Cormen](https://book.douban.com/search/Thomas%20H.Cormen) 、[Charles E.Leiserson](https://book.douban.com/search/Charles%20E.Leiserson) 、 [Ronald L.Rivest](https://book.douban.com/search/Ronald%20L.Rivest) 、[Clifford Stein](https://book.douban.com/search/Clifford%20Stein)出版社：机械工业出版社

【6】《java核心思想》  作者：Bruce Eckel 出版社：机械工业出版社

【7】《数据结构和算法》 作者：严蔚敏 出版社：清华大学出版社