**《编译原理》课程**

**设计报告**

**专业 软件工程**

**班级 100685501**

**学号 1006840526**

**姓名 梁懿**

**指导教师 朱保平**

**南京理工大学计算机学院**

**2013年7月**

## 需求实现

1. 词法分析，关键字识别与变色
2. 语法分析，实现了LALR(1)算法。包括文法的识别，文法的分析，包括建立项目集族，计算Action表格；对于给定的串分析其语法，判断是否为文法的句子。
3. 四元组的优化，建立DAG树，实现删除公共子表达式等。

## 关键算法

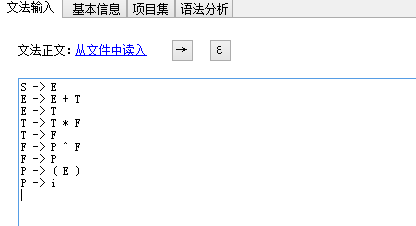
1. DFA识别关键字
2. 项目集闭包的运算
3. 计算向前搜索符
4. 计算项目集的Action表并拓展项目族
5. First集的计算
6. 语法测试
7. DAG树的创建

## 功能与界面

1. 输入源代码识别关键字



1. 输入文法产生式



1. 计算文法的基本信息



1. 计算项目集与分析表：



1. 输入句子进行分析，对于不属于该文法的句子，程序将以弹窗的形式报错



## 编程中遇到的问题与思考

1. 在开发的一开始，我想学习lex 和yacc，或者在自己的代码中嵌入它们，为此我查阅了许多资料，在linux下实现了这两个工具，最后发现这个方案并不可行，它们生成的c语言代码基本可以说是不可阅读的。同时，我采用的是C#语言，如果只是是在自己的程序中调用人家的dll，感觉就不像是自己做的了。另外，对于老师给出的要求，我自己想要的实现时更为复杂的，我想要输入语言，文法，生成对应的四元式，结果后来发现工程量实在太大，所以我只实现了老师要求的一些功能。

2. 词法分析实现，一开始我画了一个特别复杂的dfa，按照画出的dfa实现，将是一个非常复杂庞大的工程，我参看了一些别人的成语，有的但是dfa就写上了几百行，而且用数字表示状态，按照软件工程的思路，这样的代码是难以维护的。我进行思考后，决定在初始状态下输入两个字符后进行状态的转移，同时建立一个缓冲区，将缓冲区的字符附加到带识别的串中，将下一字符作状态转化同时放入缓冲区。实现了代码的优化，思路清晰，便于理解与维护。

3. 我采用了c#语言编程，在语法分析的实现过程中，我努力按照c#的语言规范，实现结构优雅，易于理解的代码，在代码的设计中，我费了很大的心思，包括终结符，项目集都封装的很好。同时，因为这是自己第一次采用C#语言写工程，这也给我的开发带来了一定的难度，我的代码也够明朗，易懂，同时也是很庞大的，超过了千行。

4. 还有一些乱七八糟的小问题，就不在这里一一陈述了。

## 心得和体会

做这个课程设计花了我很多的心思，恰好时间在期末，考试一门接着一门，很难抽出一个完整的时间，经常都是编程都晚上一两点。

同时，关于自己编译原理的课程，感觉自己的学习能力有所欠缺，到了编程的时候，还觉得自己很多课堂上的知识理解的还不够透彻，从而感觉无从下手，只能耐着性子再温习下书本。

另外，自己第一次采用C#编写工程，感觉上这是一个上手很容易的语言，但是按照它的规范进行编程，却并没有想象中的容易，编程的过程的中出了很多基础的问题，我一边查阅文档，一边解决。另外也受自己完美主义的影响，不断的编程，还不断的重构，或许我的程序所实现的功能不是最强大的，但是我的代码绝对是经过了精心的设计的。

时间仓促，感觉这是自己过着最累，也最充实的一个期末。写完代码后，这几天一直头晕晕的，做梦都是代码，如果有不足之处，还请老师指教，谢谢！