编译原理第一次实验报告

151220126 吴杰

实验总体汇报:

我已经完成实验一的所有内容，通过了所有测试样例。我选做的内容是 1.3

词法分析:

词法分析参考 C—文法的 Tokens 部分,简单地写一些正则表达式就可以实现。

语法分析:

我的语法分析分三部分完成，就和参考资料上说的一模一样。

确定语法产生式正确:

在 syntax.y 文件中将参考资料上的语法产生式仔细抄对就没啥大问题了。另外文件的头文件，终结符和非终结符声明按照书上讲的写就可以了。

构建语法分析树:

1、我的语法分析树是一个森林，结点左边连子女，右边连兄弟。

比如 ExtDef : Specifier SEMI

ExtDef 是父节点，它的 Child 结点连着 Specifier，Specifier 的 nextnode 节点连接着 SEMI(SEMI 是 Specifier 的兄弟)

2、我的分析树每个结构点的数据结构如下:

```
struct TreeData {
    double dnum;                    //存储 float 类型的值
    char *value;                    //存储终结符
    char *token;                    //存储终结符类型
    struct TreeData *Child;         //子女节点
    struct TreeData *nextnode;      //兄弟节点
    int lineno;                     //行数
};
```

本来想省去 dnum 的，后来发现对 value 使用 atof 与 atof 定义有冲突，一直没法解决，所以添加了这个中间量。

3、按照语法产生式, 在 syntax.y 文件中将$$,$1,$2…按照父子兄弟连接起来就可。这里我用了 buildChild 和 buildNode 两个函数实现父子，兄弟的连接。

至此，语法分析树基本构建完成

4、错误分析:

参考书上给了几条错误分析时一些语法产生式，不够，还需要加一些产生式，有些在网上可以找找。然后将这些产生式加入到语法树中，和之前的相同，只是 error 不要加进去。

打印语法分析树:

写一个对数的深度遍历，输出一些要求的相关信息就可以了。