

编译原理实验报告

词法及语法分析

121220100 王梓轩

完成的功能点：

1. 基本功能（识别 A 类或 B 类错误，打印语法树）。
2. 额外功能：八进制及十六进制数的识别、普通浮点数及科学记数法浮点数的识别、两种注释的识别。

编译方法：

依次输入：（XXX 为测试用的 C 文件）

```
make lex
make syn
make compile
./parser XXX
```

结点结构：

语法树结点的结构及意义：

```
struct Node
{
    unsigned int lineno; //结点所在的行号
    char* name; //结点的名称
    char* value; //结点的类型
    struct Node** sons; //结点所指向的所有子结点，构成动态指针数组
    unsigned int sonNum; //结点子结点的个数
    unsigned int isLeaf; //结点是否为叶结点
};
```

亮点：

1. 创建结点的函数使用的是变参函数，这样使得创建任意多个结点事较为简洁。
2. 结点指向的子结点个数由变参函数的参数传进来，并申请一块存放该数目子结点的空间，并赋相应的值。
3. 结点在打印的时候并没有选择空间开销较大的递归方式，而是自己建立了一个链表当作栈用，实现深度优先搜索。
4. 链表中的每个结点的层数由其父结点加一得到，这样可以根据层数方便地打印出缩进。
5. 结构体中用 isLeaf 判断是否为叶结点，也就是判断是否为终结符。
6. “/*...*/”方式的注释参见了 <http://bbs.csdn.net/topics/40224740>，实现了对嵌套注释的报错。
7. 通过添加一些特殊错误的正则表达式，可以将这些错误处理成词法错误（如错误的八进制、十六进制、浮点数）。
8. 利用“%error-verbose”选项自动判断错误类型，判断更加准确。