

Lab1 实验报告

学生信息

姓名：李子沐；学号：221180029；邮箱：221180029@smail.nju.edu.cn

实验环境和编译方法

实验环境与OJ要求相同，即

- Ubuntu 20.04 64bit
- GCC 7.5.0
- Flex 2.6.4
- Bison 3.6.1

使用makefile编译即可。

使用命令编译方法如下：

在目录 李子沐_221180029/Code/ 下，输入

```
flex lexical.l
bison -d syntax.y
gcc main.c mynode.c syntax.tab.c -ldl -o parser
```

实现功能

完成实验要求的必做和选做部分，即：

1. 使用flex和bison完成对C--的词法分析和语法分析
2. 输入一个.cmm文件，若无错误，输出要求格式的语法树；若有错误，输出错误类型与行号
3. 输入INT型数据支持八进制，十六进制输入，FLOAT类型数据支持指数形式输入，并处理为十进制数/标准浮点数输出。
4. 可以识别并跳过行注释 // 和块注释 /*...*/

注：同一行最多处理一个错误。代码中定义全局的errorline，与当前的yylineno对比判断多的错误是否输出。

在识别错误类型（Type A或Type B）外，额外实现了一些具体的错误类型：

1. 词法错误：通过定义错误的整型和浮点型词法单元，可以检测错误的整型、浮点型输入
2. 词法错误：可以检测不在定义中的字符（参考实验指导的实现）

3. 语法错误：可以识别（待更新）

注：测试文件全部来自实验指导的样例

实验细节细节

实现步骤主要参考实验指导，关键代码与问题如下。

多叉树

节点定义如下，包含当前词法/语法单元所在行号 `lineno` 等信息。

为了方便操作，这里定义子节点数量最大为 `MAX_CHILDREN_NUM`，未来若有动态分配的需求再修改。

测试遇到的bug：要求产生式生成空串时不输出，由于我在语法分析中为空串分配了语法单元，因此直接打印节点仍然会输出对应的非终结符。通过增加 `nodetype` 成员，通过 `nodetype == SYN_NODE && childnum == 0` 判断产生空串

```
typedef struct node {
    int lineno;                /* the line num of this node*/
    char name[32];             /* e.g. TYPE , make sure that name is less than 32'
    /* Attention: the INT and FLOAT is stored as a char array (string) */
    char value[32];            /* e.g. int , make sure that value is less than 32'
    struct node* parent;       /* only one parent */
    struct node* childs[MAX_CHILDREN_NUM]; /* possibly many childs, but must be less than MAX_
    int childnum;              /* the number of childs */
    int nodetype;
} Node;
```

添加多个子节点

利用 `<stdarg.h>` 的可变参数，实现了 `add_many_childs` 函数，方便产生式使用。

块注释忽略

通过 `input` 和 `while` 实现块注释忽略，同时可以记录跳过行数（尽管没有用）

总结

笨人编程能力较差，上一次独立写代码还是在大一上的二层次程序设计课。构建多叉树的过程遇到了许多弱智问题，例如字符数组要用 `strcpy()` 赋值，总之最后还是磕磕绊绊地完成了实验！

实验过程与想象中不太相同，我以为要手写DFA和LALR分析器，但是在实验指导的帮助下整体体验很棒（比电子的实验课强多了），感谢助教！感谢老师！