词法和语法分析

151160018 郝凯龙 hao-kailong@qq.com

一、实现功能

借助词法分析工具 GNU Flex 和语法分析工具 GNU Bison，使用 C 语言来完成实验。可以生成语法树。对于错误的程序，可以提示出词法和语法错误。

二、实验环境

GNU Linux Release: Ubuntu 16.04 LTS
GCC version: 5.4.0
GNU Flex version: 2.6.4
GNU Bison version: 3.0.4

三、编译方法

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

四、重点难点

1.多叉树的构造

将每一个终结符和非终结符作为多叉树的一个节点。多叉树节点定义为 struct Node*类型。为每一个节点用 malloc 分配空间。

多叉树采用链表实现。节点定义如下。

```
struct Node{
    union{
        int valInt;
        float valFloat;
        double valDouble;
    };//值
    int line;//行号
    char type[64];//类型
    char name[64];//名字
    struct Node* child;//孩子
    struct Node* next;//兄弟
};
```

通过以下函数构造。

```
void addNext(struct Node *a, struct Node *b);为节点a添加兄弟节点b
```

struct Node* generateNode(double val,int line,char *type,char *name,struct Node *child,struct Node *next);开辟空间生成新节点，返回指针

2.错误恢复

在理解程序含义的基础上，在可能出错的地方进行错误恢复。

添加了多条规则。举例如下。

```
Stmt: error SEMI {hasError=1;}
| Exp error {hasError=1;}
| RETURN Exp error {hasError=1;}
| IF LP error RP Stmt {hasError=1;}
| IF LP error RP Stmt ELSE Stmt {hasError=1;}
| WHILE LP error RP Stmt {hasError=1;};
```

在出现语法错误时，不再打印语法树。

3.注释识别

在词法规则中使用以下正则表达式识别注释。

```
SHORTNOTE \/\/.*\n
LONGNOTE \/\*[^(\/\*)(\*\/)]*\*\/
```

在识别出注释后直接抛弃，因为语法分析时并不需要。