**语法分析器实验报告**

姓名：姚梦园

班级：20级计基

学号：320200918701

1. 源语言

（2）文法定义：

PROG        →    ****{****  DECLS  STMTS  ****}****

DECLS       →    DECLS  DECL    ****|****empty

DECL         →    ****int****NAMES  ****;  |  bool****NAMES  ****;****

NAMES     →    NAMES ****,****  NAME  ****|****  NAME

NAME       →    ****id****

STMTS    →    STMTS  STMT  ****|****   STMT

STMT      →    ****id  =****  EXPR ****;    |   id :=****BOOL****;****

STMT      →    ****if  id****  ****then****STMT

STMT      →    ****if   id****  ****then****STMT  ****else**** STMT

STMT      →    ****while   id**** ****do****STMT

STMT      →    ****{****  STMTS   STMT  ****}****

STMT      →    ****read  id  ;****

STMT      →    ****write  id  ;****

EXPR    →    EXPR  ADD  TERM  ****|****  TERM

ADD     →   ****+ | -****

TERM    →    TERM  MUL NEGA ****|**** NEGA

MUL     →    ****\* | /****

NEGA   ****→****FACTOR ****|  -**** FACTOR

FACTOR→    ****(****  EXPR ****) |  id  |**** ****number****

BOOL    →    BOOL  ****||****  JOIN    ****|****    JOIN

JOIN     →    JOIN   ****&&****   NOT ****|****  NOT

NOT      →    REL   ****|  !****REL

REL       →    EXPR   ROP  EXPR

ROP      →     ****>  |  >=  |  <  |  <=  |  ==  |   !=****

1. 程序设计流程

lex(source\_file,Buffer);//词法分析，将分析后的词素写入结构数组Buffer

    prog(node);  //进行语法分析

    write\_in(root,des\_file); //将语法分析的结果写入目标文件

将词法分析产生的token写入内存，进行语法分析

关键点1：左递归的消除

Eg.

TREE \* decls(){

    printf("invoking decls!\n");

    TREE \* l,\*r,\*f,\*root;

    save = p;

    if((l = decl())!=NULL)              //left node

    {

        while(1)

        {

            save = p;

            if((r = decl()) != NULL)  // right node exist!

            {

                TREE \* fnode = (TREE \*)malloc(sizeof(TREE)); //father node

                f = fnode;

                f->left = l;

                f->right = r;

                f->third = NULL;

                strcpy(f->value,"decls");

                l = f;

            }

            else    //right node does not exist!

            {

                p = save;

                break;

            }

        }

    }

    else  //left node does not exist!

    {

        p = save;

        root = NULL;

    }

    root = l;

    return root;

}

关键点2：

函数返回该函数生成的抽象语法树的根节点，向上传递，以此建立一颗完整的抽象语法树

关键点3：

注释的消除，注释语句直接在词法分析过程中忽略，并不进入语法分析

为方便下一步语义分析，给抽象语法树中出现的序列编码

prog 38

decls 39

stmts 40

names 41

if\_then 42

if\_then\_else 43

while\_do 44

block\_stmt 45

1. 代码实现和测试用例

此部分见于附件