**北京邮电大学软件学院**

**2016－2017学年第一学期实验报告**

**课程名称： 编译原理**

**项目名称： 语法分析程序的设计和实现**

**项目完成人：**

**姓名：肖逸敏\_\_\_\_\_\_\_\_学号：\_\_2014211990\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**指导教师：\_\_\_\_\_杨正球\_\_\_\_\_\_\_\_\_\_\_\_\_**

**日 期： 2016 年 11 月 25 日**

1. **实验内容**

编写语法分析程序，实现对算法表达式的语法分析。要求所分析算数表达式由如下的文法产生。

E -> E+T | E-T | T

T -> T\*F | T/F | F

F -> id | (E) | num

1. **实验要求**

在对输入表达式进行分析的过程中，输出所采用的产生式。

编写语法分析程序实现自底向上的分析，要求如下。

* + 1. 构造识别所有活前缀的DFA。
    2. 构造LR分析表。
    3. 编程实现算法4.3，构造LR分析程序。

1. **实验环境**

VS2013

1. **实验结果**
   * + 1. 构造识别所有活前缀的DFA
2. S->E
3. E->E+T
4. E->E-T
5. E->T
6. T->T\*F
7. T->T/F
8. T->F
9. F->id
10. F->(E)
11. F->num

























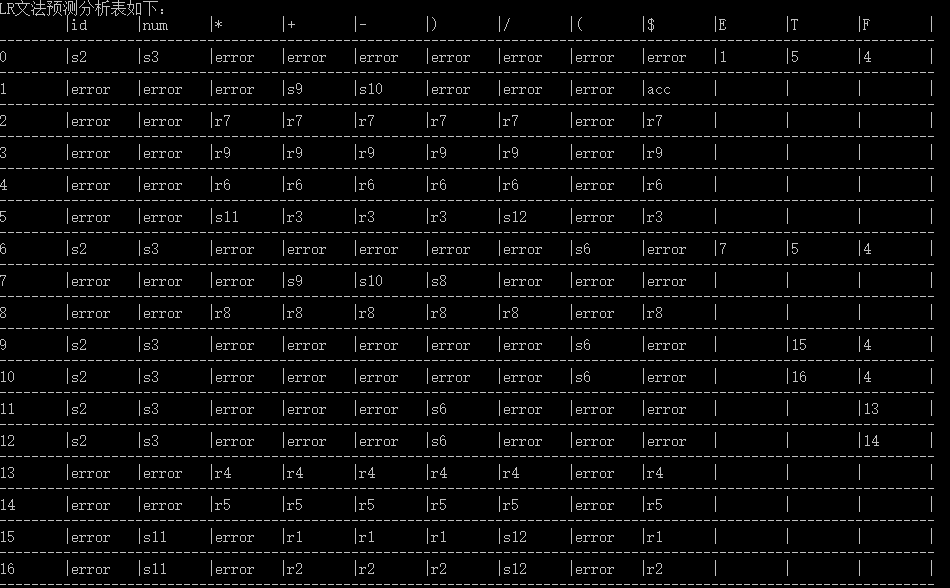




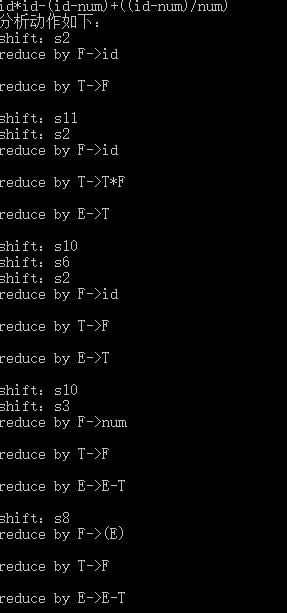


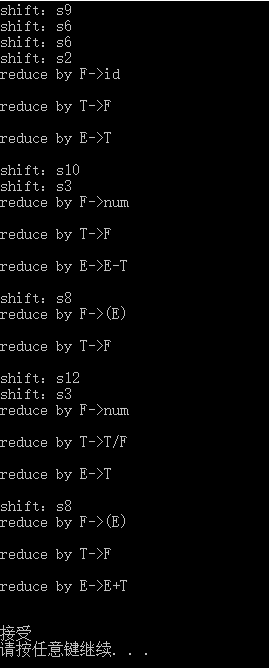


* + - 1. 识别LR分析表



* + - 1. 构造LR分析程序





1. **附录**

1.采用LR文法

2.将预测分析表用二维数组的形式输出

3.对栈顶元素，若该元素不是状态，则通过布尔变量is\_stick来判断该元素不是一个状态，并计算新的状态并压入栈中

代码如下：

#include <stdio.h>

#include <iostream>

#include<stdlib.h>

#include <math.h>

#include <string>

#include <iomanip>

#include <vector>

#include <sstream>

#include<stack>

using namespace std;

//定义栈

stack<string> stk;

//产生式

char\* S = "S->E";

char\* S1 = "E->E+T";

char\* S2 = "E->E-T";

char\* S3 = "E->T";

char\* S4 = "T->T\*F";

char\* S5 = "T->T/F";

char\* S6 = "T->F";

char\* S7 = "F->id";

char\* S8 = "F->(E)";

char\* S9 = "F->num";

//FIRST集和FOLLOW集

char\* FollowF[100] = { ")", "+", "-", "\*", "/", "$" };

char\* FollowT[100] = { ")", "+", "-", "\*", "/", "$" };

char\* FollowE[100] = { ")", "$", "+", "-", };

int main(int argc, char\* argv[]){

int index = 0;

int size;

int indexi = 0;

int indexj = 0;

string top = "";

bool check\_id = false;

bool check\_num = false;

bool is\_stick = false;

string move = "";

stk.push("0");

string show = "";

string input;

show = show + "$0";

cout << "LR文法预测分析表如下：" << endl;

cout.setf(ios::left);

char\* table[18][13] = {

{ " ", "id", "num", "\*", "+", "-", "}", "/", "(", "$", "E", "T", "F" },

{ "0", "s2", "s3", "error", "error", "error", "error", "error", "error", "error", "1", "5", "4" },

{ "1", "error", "error", "error", "s9", "s10", "error", "error", "error", "acc", " ", " ", " " },

{ "2", "error", "error", "r7", "r7", "r7", "r7", "r7", "error", "r7", " ", " ", " " },

{ "3", "error", "error", "r9", "r9", "r9", "r9", "r9", "error", "r9", " ", " ", " " },

{ "4", "error", "error", "r6", "r6", "r6", "r6", "r6", "error", "r6", " ", " ", " " },

{ "5", "error", "error", "s11", "r3", "r3", "r3", "s12", "error", "r3", " ", " ", " " },

{ "6", "s2", "s3", "error", "error", "error", "error", "error", "s6", "error", "7", "5", "4" },

{ "7", "error", "error", "error", "s9", "s10", "s8", "error", "error", "error", " ", " ", " " },

{ "8", "error", "error", "r8", "r8", "r8", "r8", "r8", "error", "r8", " ", " ", " " },

{ "9", "s2", "s3", "error", "error", "error", "error", "error", "s6", "error", " ", "15", "4" },

{ "10", "s2", "s3", "error", "error", "error", "error", "error", "s6", "error", " ", "16", "4" },

{ "11", "s2", "s3", "error", "error", "error", "s6", "error", "error", "error", " ", " ", "13" },

{ "12", "s2", "s3", "error", "error", "error", "s6", "error", "error", "error", " ", " ", "14" },

{ "13", "error", "error", "r4", "r4", "r4", "r4", "r4", "error", "r4", " ", " ", " " },

{ "14", "error", "error", "r5", "r5", "r5", "r5", "r5", "error", "r5", " ", " ", " " },

{ "15", "error", "s11", "error", "r1", "r1", "r1", "s12", "error", "r1", " ", " ", " " },

{ "16", "error", "s11", "error", "r2", "r2", "r2", "s12", "error", "r2", " ", " ", " " }

};

for (int i = 0; i<18; i++){

for (int j = 0; j<13; j++){

cout.width(8);

cout << table[i][j];

cout.width(1);

cout << "|";

}

cout << endl;

cout << "---------------------------------------------------------------------------------------------------------------------" << endl;

}

cout << "请输入表达式：" << endl;

cin >> input;

input += "$";

cout << "分析动作如下：" << endl;

while (!stk.empty()){

size = input.length();

top = stk.top();

if (move == "acc"){

cout.width(50);

cout << "接受" << endl;//动作分析

return 0;

}

else{

if (!is\_stick){

//cout<<top;

stringstream ss(top);

ss >> indexi;

//cout<<indexi;

indexi++;

}

}

//以上是决定非终结符的，以下为决定终结符的

if (is\_stick){

if (top == "E"){

indexj = 10;

}

else if (top == "T"){

indexj = 11;

}

else if (top == "F"){

indexj = 12;

}

is\_stick = false;

}

else if (input[index] == 'i'){

index++;

if (input[index] == 'd'){//确认id

check\_id = true;

indexj = 1;

index++;

}

else{

cout.width(50);

cout << "出错！" << endl;//动作分析\*/

return 1;

}

}

else if (input[index] == 'n'){

index++;

if (input[index] == 'u'){

index++;

if (input[index] == 'm'){//确认num

check\_num = true;

indexj = 2;

index++;

}

else{

cout.width(50);

cout << "出错！" << endl;//动作分析\*/

return 1;

}

}

else{

cout.width(50);

cout << "出错！" << endl;//动作分析\*/

return 1;

}

}

else if (input[index] == '\*'){

indexj = 3;

}

else if (input[index] == '+'){

indexj = 4;

}

else if (input[index] == '-'){

indexj = 5;

}

else if (input[index] == ')'){

indexj = 6;

}

else if (input[index] == '/'){

indexj = 7;

}

else if (input[index] == '('){

indexj = 8;

}

else if (input[index] == '$'){

indexj = 9;

}

else if (input[index] == 'E'){

indexj = 10;

}

else if (input[index] == 'T'){

indexj = 11;

}

else if (input[index] == 'F'){

indexj = 12;

}

//至此确定使用哪一个动作

move = table[indexi][indexj];

if (move == "error"){

cout << "出错！" << endl;

return 1;

}

if (move[0] == 's'){//shift操作

cout << "shift：" << move;

if (check\_id){

stk.push("id");

stk.push(move.substr(1));

check\_id = false;

}

else if (check\_num){

stk.push("num");

stk.push(move.substr(1));

check\_num = false;

}

else{

if (input[index] == '+'){

stk.push("+");

stk.push(move.substr(1));

index++;

}

else if (input[index] == '\*'){

stk.push("\*");

stk.push(move.substr(1));

index++;

}

else if (input[index] == '-'){

stk.push("-");

stk.push(move.substr(1));

index++;

}

else if (input[index] == ')'){

stk.push(")");

stk.push(move.substr(1));

index++;

}

else if (input[index] == '/'){

stk.push("/");

stk.push(move.substr(1));

index++;

}

else if (input[index] == '('){

stk.push("(");

stk.push(move.substr(1));

index++;

}

}

}

else if (move[0] == 'r'){//reduce操作

if (move[1] == '1'){//使用E->E+T归约

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stringstream ss(stk.top());

ss >> indexi;

indexi++;

stk.push("E");

is\_stick = true;

cout << "reduce by E->E+T";//动作分析

}

else if (move[1] == '2'){//使用E->E -T归约

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stringstream ss(stk.top());

ss >> indexi;

indexi++;

stk.push("E");

is\_stick = true;

cout << "reduce by E->E-T";//动作分析

}

else if (move[1] == '3'){//使用E->T归约

stk.pop();

stk.pop();

stringstream ss(stk.top());

ss >> indexi;

indexi++;

stk.push("E");

is\_stick = true;

cout.width(50);

cout << "reduce by E->T";//动作分析

}

else if (move[1] == '4'){//使用T->T\*F归约

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stringstream ss(stk.top());

ss >> indexi;

indexi++;

stk.push("T");

is\_stick = true;

cout.width(50);

cout << "reduce by T->T\*F";//动作分析

}

else if (move[1] == '5'){//使用T->T/F归约

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stringstream ss(stk.top());

ss >> indexi;

indexi++;

stk.push("T");

is\_stick = true;

cout.width(50);

cout << "reduce by T->T/F";//动作分析

}

else if (move[1] == '6'){//使用T->F归约

stk.pop();

stk.pop();

stringstream ss(stk.top());

ss >> indexi;

indexi++;

stk.push("T");

is\_stick = true;

cout.width(50);

cout << "reduce by T->F";//动作分析

}

else if (move[1] == '7'){//使用F->id归约

stk.pop();

stk.pop();

stringstream ss(stk.top());

ss >> indexi;

indexi++;

stk.push("F");

is\_stick = true;

cout << "reduce by F->id";//动作分析

}

else if (move[1] == '8'){//使用F->(E)归约

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stk.pop();

stringstream ss(stk.top());

ss >> indexi;

indexi++;

stk.push("F");

is\_stick = true;

cout.width(50);

cout << "reduce by F->(E)";//动作分析

}

else if (move[1] == '9'){//使用F->num归约

stk.pop();

stk.pop();

stringstream ss(stk.top());

ss >> indexi;

indexi++;

stk.push("F");

is\_stick = true;

cout.width(50);

cout << "reduce by F->num";//动作分析

}

}

else{//goto 状态

stk.push(move);

}

cout << endl;

}

return 0;

}