**实 验 报 告（五）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | LR(0)分析 | | | 专 业 | 计算机科学与技术 |
| 课程名称 | 编译原理 | 指导老师 | 赵智超 | 班 级 | 03 |
| 姓 名 | 孔伟恒 | 学 号 | 18100140316 | 评 分 |  |
| 实验地点 | 1C06217 | 实验日期 | 2019/11/20 |
| 一、实验目的  构造LR(0)分析程序，利用它进行语法分析，判断给出的符号串是否为该文法识别的句子，了解LR（K）分析方法是严格的从左向右扫描，和自底向上的语法分析方法。 | | | | | |
| 二、实验内容（含实验原理介绍）  程序输入/输出示例：  对下列文法，用LR（0）分析法对任意输入的符号串进行分析：  （1）E->E+T  （2）E->E—T  （3）T->T\*F  （4）T->T/F  （5）F->(E)  （6）F->i | | | | | |
| 1. 实验过程及步骤（包含使用软件或实验设备等情况）   #include <iostream> #include <stack> #include <cstdlib> #include <string> using namespace std;  //初始化预测分析表 string action[12][6] = {  {"S5", "error", "error", "S4", "error", "error"},  {"error", "S6", "error", "error", "error", "acc"},  {"error", "r2", "S7", "error", "r2", "r2"},  {"error", "r4", "r4", "error", "r4", "r4"},  {"S5", "error", "error", "S4", "error", "error"},  {"error","r6", "r6", "error", "r6", "r6"},  {"S5", "error", "error", "S4", "error", "error"},  {"S5", "error", "error", "S4", "error", "error"},  {"error", "S6", "error", "error", "S11", "error"},  {"error", "rl", "r7", "error", "r1", "rl"},  {"error", "r3", "r3", "error", "r3", "r3"},  {"error", "r5", "r5", "error", "r5", "r5"} };  string go[12][3] = {  {"1", "2", "3"},  {"error", "error", "error"},  {"error", "error", "error"},  {"error", "error", "error"},  {"8", "2", "3"},  {"error", "error", "error"},  {"error", "9", "3"},  {"error", "error", "10"},  {"error", "error", "error"},  {"error", "error", "error"},  {"error", "error", "error"},  {"error", "error", "error"}, };  char Vt[6] = {'i', '+', '\*', '(', ')', '#'}; //终结符表 string LR[6] = {"E->E+T", "E->T", "T->T\*F", "T->F", "F->(E)", "F->i"}; //LR文法 stack <int> S; //状态栈 stack <char> X; //符号栈 char input[10]; //输入字符  int num (string s) { //判断字符串中的数字  int i;  string str = "";  for (int j = 0; j < s.length(); j++) {  if (s[j] >= '0' && s[j] <= '9')  str = str + s[j];  }  i = atoi(str.c\_str());  return i; }  void preview (char \*c) {  for (int j = 0; c[j] != '#'; j++) {  cout << c[j];  }  cout << endl; }  void print (int i, char \*c) {  for (int j = i; c[j - 1] != '#'; j++) {  cout << c[j];  }  cout << '\t'; }  int same (char a) { //用于查找终结符  for (int i = 0; i < 6; i++) {  if (a == Vt[i]) return i;  }  return -1; }  void analyse () { //分析程序  bool flag = true; //循环条件控制  int step = 1, point = 0, state = 0; //步骤、指针、状态   char ch1, ch2;  int m, n, l; //用于判断终结符，分析表,表达式右部的长度  string str1; //用于判断对应分析表中的符号  string str2 = "#", str3 = "0"; //记录符号栈的所有元素  cout << "姓名:孔伟恒 学号: 18100140316 班级:计算机三班\n\n";  cout << "请输入要规约的字符串，并以#号结束: "<<endl;  cin >> input;  cout << "您输入的字符串为: ";  preview (input);  X.push ('#');  S.push (0);  cout<< "步骤" << '\t' << "状态栈" << '\t' << "符号栈" << '\t' << "输入串"<< '\t' << "动作" << endl;  cout << step++ <<'\t'<< str3 << '\t' << str2 << '\t';  print (point, input);  cout << "初始化" << endl; //初始化  while (flag) { ///进入循环/  state = S.top();  ch1 = input[point];  m = same(ch1);  str1 = action[state][m];  //移进动作/  if (str1[0] == 'S') {  n = num(str1);  S.push(n);  X.push(ch1);  str2 = str2 + ch1;  ch2 = n + 48;  str3 = str3 + ch2;  point++;  cout << step++ << '\t' << str3<< '\t' <<str2 << '\t';  print (point, input);  cout << str1 << ":" << "移进"<< endl;  } else if (str1[0] == 'r') {//归约动作  n = num(str1);  l = LR[n - 1].length() - 3;  for (int i = 1; i <= l; i++) {  S.pop();  str3 = str3.substr(0, str3.length() - 1);  X.pop();  str2 = str2.substr(0, str2.length() - 1);  }  X.push(LR[n - 1][0]);  str2 = str2 + LR[n - 1][0];  state = S.top();  if (LR[n - 1][0] == 'E') {  S.push(num(go[state][0]));  ch2 = num(go[state][0]) + 48;  str3 = str3 + ch2;  } else if (LR[n - 1][0] == 'T') {  S.push (num (go[state][1]));  ch2 = num(go[state][1]) + 48;  str3 = str3 + ch2;  } else if (LR[n - 1][0] == 'F') {  S.push(num(go[state][2]));  ch2 = num(go[state][2]) + 48;  str3 = str3 + ch2;  }  cout << step++ << '\t' << str3 << '\t' << str2 << '\t';  print (point, input);  cout << str1 << ":" << LR[n - 1] << "归约\n";// << endl;  } else if (str1 == "error") { //出错:  cout << "Error!\n" << "程序错误，分析结束! "<< endl;  flag = false;  } else if (str1 == "acc") { //分析成功  cout << "Acc\n" << "分析成功，终止程序! " << endl;  flag = false;  }  } }   int main () {   analyse();   return 0; } | | | | | |
| 四、实验结果（含算法说明、程序、数据记录及分析等，可附页）  数据测试 | | | | | |
| 五、实验思考题  １、LR的分析的优点是什么？  ２、LR分析器的组成？ | | | | | |
| 六、实验总结（含实验心得体会，收获与不足等） | | | | | |

注：双面打印