中山大学数据科学与计算机学院本科生实验报告

课程名称：编译器构造实验 任课教师：陈炬桦

|  |  |  |  |
| --- | --- | --- | --- |
| 学年度 | 2018-2019 | 学期 | 第二学期 |
| 年级 | 16级 | 专业（方向） | 计算机科学与技术 |
| 学号 | 16337341 | 姓名 | 朱志儒 |
| 电话 | 15989184223 | Email | 739741104@qq.com |
| 开始日期 | 2019/3/8 | 完成日期 | 2019/3/8 |

1. 实验题目

输入输出文法

* 1. Description

输入开始符号，非终结符，终结符，产生式；

按非终结符顺序输出产生式；

* 1. Input

输入开始符号；  
非终结符个数，非终结符，空格符分隔；  
终结符个数，终结符，空格符分隔；  
产生式的个数，各产生式的左边和右边符号，空格符分隔；

* 1. Output

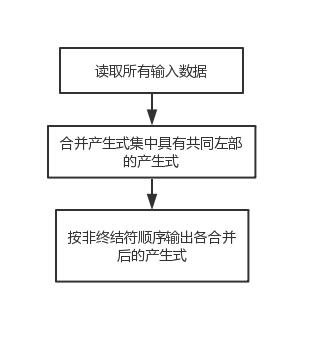
G[开始符号]：

按非终结符顺序输出各产生式；

1. 算法描述（介绍程序模块功能，流程图）
2. 算法：

* 读取所有输入数据
* 合并产生式集中具有共同左部的产生式，如合并成
* 按非终结符顺序输出各合并后的产生式

1. 流程图：



1. 测试数据（2组）
2. 测试数据：

输入：

Z

8 Z E F P G T Q S

3 + \* i

18

Z E+T

E E

P G

F F

P G

G G

T T\*i

Q E

S i

E S+F

F FP

G GG

Q E+F

E T

F P

G F

Q T

Q S

输出：

G[Z]:

Z::=E+T

E::=E | S+F | T

F::=F | FP | P

P::=G | G

G::=G | GG | F

T::=T\*i

Q::=E | E+F | T | S

S::=i

1. 测试数据：

输入：

E

3 E T F

4 + \* ( )

6

E E+T

T T\*F

E T

T F

F (E)

F i

输出：

G[E]:

E::=E+T | T

T::=T\*F | F

F::=(E) | i

1. 程序清单
2. #include <iostream>
3. #include <string>
4. #include <vector>
5. #include <algorithm>
6. **using** **namespace** std;
8. **struct** VN{
9. **int** Nv;
10. string VN[100];
11. };
12. VN Vns = { 0 };
14. **struct** VT{
15. **int** Nt;
16. string VT[100];
17. };
18. VT Vts = { 0 };
20. **struct** PS {
21. **int** Np;
22. string PL[200], PR[200];
23. };
24. PS ps = { 0 };
26. string S;
28. **int** main() {
29. cin >> S;
30. cin >> Vns.Nv;
31. **for** (**int** i = 0; i < Vns.Nv; ++i)
32. cin >> Vns.VN[i];
33. cin >> Vts.Nt;
34. **for** (**int** i = 0; i < Vts.Nt; ++i)
35. cin >> Vts.VT[i];
36. cin >> ps.Np;
37. **for** (**int** i = 0; i < ps.Np; ++i)
38. cin >> ps.PL[i] >> ps.PR[i];
40. vector<string> tmp;
41. PS output = { 0 };
42. **for** (**int** i = 0; i < ps.Np; ++i)
43. **if** (find(tmp.begin(), tmp.end(), ps.PL[i]) == tmp.end()) {
44. tmp.push\_back(ps.PL[i]);
45. output.PL[output.Np] = ps.PL[i];
46. output.PR[output.Np] = ps.PR[i];
47. **for** (**int** j = i + 1; j < ps.Np; ++j)
48. **if** (ps.PL[i] == ps.PL[j])
49. output.PR[output.Np] += " | " + ps.PR[j];
50. output.Np++;
51. }
53. cout << "G[" << S << "]:" << endl;
54. **for** (**int** i = 0; i < Vns.Nv; ++i) {
55. **for** (**int** j = 0; j < output.Np; ++j) {
56. **if** (output.PL[j] == Vns.VN[i]) {
57. cout << output.PL[j] << "::=" << output.PR[j] << endl;
58. **break**;
59. }
60. }
61. }
62. **return** 0;
63. }