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 教学助理（TA）：

|  |  |  |  |
| --- | --- | --- | --- |
| 学年度 | 2018-2019 | 学期 | 第二学期 |
| 年级 | 2016 | 专业（方向） | 计算机科学与技术 |
| 学号 | 16337052 | 姓名 | 杜尔鑫 |
| 电话 | 15626281204 | Email | duerx@mail2.sysu.edu.cn |
| 开始日期 | 2019-03-08 | 完成日期 | 2019-03-08 |

1. 实验题目

**1.1 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；

此题需要提交实验报告；实验报告用“学号+姓名+22”

**1.2 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；

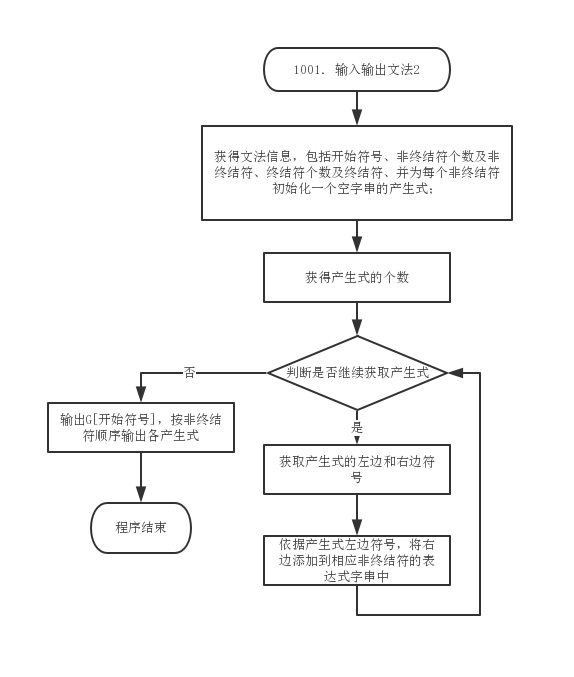
**1.3 Output**

G[开始符号]：

按非终结符顺序输出各产生式；

2. 算法描述(介绍程序模块功能；流程图)

1. 输入文法信息并储存，包括开始符号、非终结符个数及非终结符、终结符个数及终结符；并为每个非终结符初始化一个空字串的产生式；
2. 输入产生式的个数，各产生式的左边和右边符号；并根据左边将右边添加到相应非终结符的表达式中，且使其格式化；
3. 输出G[开始符号]，按非终结符顺序输出各产生式；



3. 测试数据(2组)

**3.1 Standard Example**

**3.1.1 Input**

|  |
| --- |
| 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 |

**3.1.2 Output**

|  |
| --- |
| 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 |

**3.2 Addition Test （由题1000改编）**

**3.2.1 Input**

|  |
| --- |
| E  3 E T F  7 + - \* / ( ) i  8  E E+T  T T\*F  E T  T F  F (E)  F i  E E-T  T T/F |

**3.2.2 Output**

|  |
| --- |
| G[E]:  E::=E+T | T | E-T  T::=T\*F | F | T/F  F::=(E) | i |

4. 程序清单

|  |
| --- |
| #include <iostream>  using namespace std;  struct Vns\_Struct { int Nv; string VN[10];} ;  struct Vts\_Struct { int Nt; string VT[10];} ;  struct ps\_Struct { int Np; string PL[20], PR[20];} ;  string S;  int main(int argc, char const \*argv[])  {      Vns\_Struct Vns;      Vts\_Struct Vts;      ps\_Struct ps;      // 开始符号      cin >> S;      // 非终结符      cin >> Vns.Nv;      string out[Vns.Nv];      for (int i = 0; i < Vns.Nv; ++i)      {          cin >> Vns.VN[i];          out[i] = "";      }      // 终结符      cin >> Vts.Nt;      for (int i = 0; i < Vts.Nt; ++i)          cin >> Vts.VT[i];      // 产生式      cin >> ps.Np;      for (int i = 0; i < ps.Np; ++i)      {          cin >> ps.PL[i] >> ps.PR[i];          int j = 0;          for (; j < Vns.Nv; ++j)              if (Vns.VN[j] == ps.PL[i])                  break;          if (out[j] != "")              out[j] += " | ";          out[j] += ps.PR[i];      }      // 输出部分      cout << "G[" << S << "]:" << endl;      for (int i = 0; i < Vns.Nv; ++i)          cout << Vns.VN[i] << "::=" << out[i] << endl;      return 0;  } |