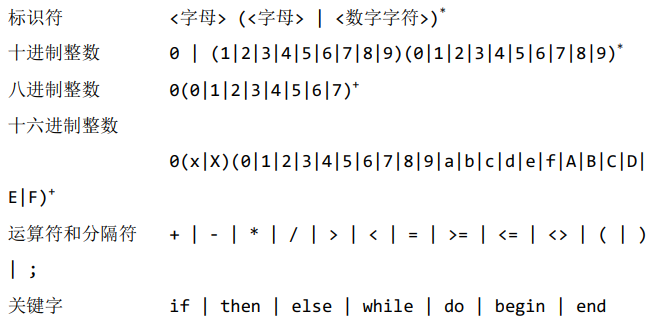
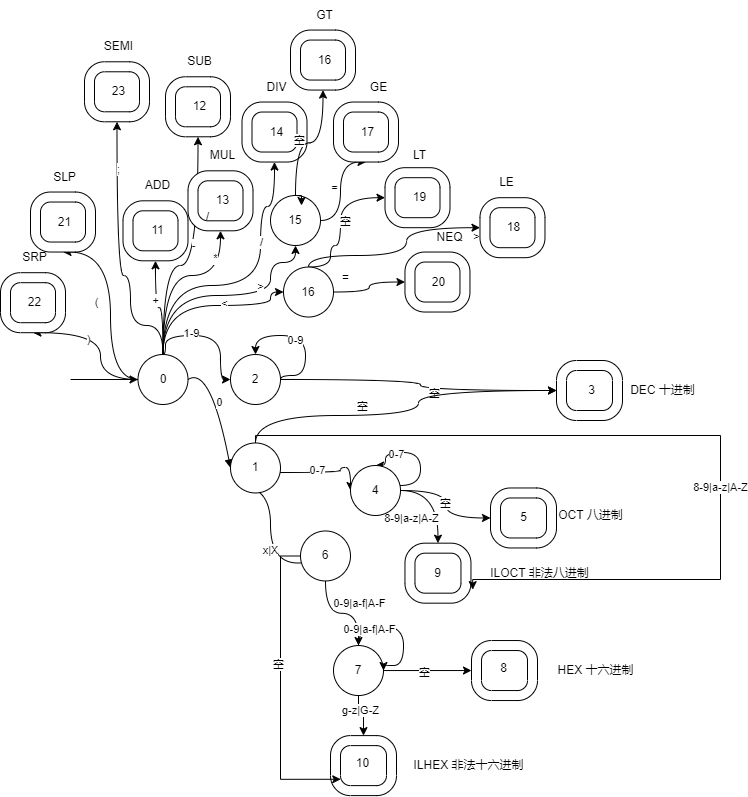
编译原理实验报告

1. 词法分析子系统
2. 词法的正规式描述



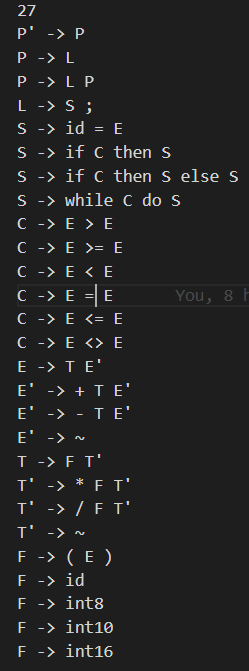
1. 变换后的正规文法
2. 状态图



1. 词法分析程序的主要数据结构与算法
   1. 数据结构
      1. class symbolTableNode 存储语法单元基本属性
         1. string type 存储种别值
         2. int typeCode 以数字形式存储种别值
         3. string props 存储字符串属性值
         4. int prop 存储数值类型属性值
      2. 全局变量
         1. 与tellType函数配合进行字符类型判断
            1. IS\_LETTER 0
            2. IS\_DIGIT 1
            3. IS\_SYMBOL 2
         2. enum TOKEN枚举类型用于配合typeCode以数字形式存储种别值
   2. 算法
      1. queue<string> sep\_words ( string in )
         1. 输入：从用户端直接获取到的字符串
         2. 输出：分割好的序列
         3. 功能：将从用户端直接获取到的字符串进行分割，按顺序存放在队列里返回给上层函数
         4. 实现方式简述：逐字符遍历字符串，调用tellType函数判断当前字符的类型，通过标志位（初值为输入字符串第一个字符的类型）记录上一次类型发生变化后的类型。当类型不发生改变且当前类型不为symbol时，在临时字符串t尾部附加当前字符；当类型发生改变时，记录变化后的类型值，将当前的t字符串push到队列里。  
            有几个特例进行特殊处理：
            1. 为16进制数字设置标志位，当前一字符为0，当前字符为x或X时，设置flag为1，当flag为1时，进入特殊处理方式，不再按上文处理方式进行处理，持续读入字符直到分隔符或symbol类型截止并压栈。
            2. 当前一输入字符为<或>时，判断当前字符是否为可以与前一符号构成整体的=或>，若可以构成整体，将当前字母附加到字符串尾部。
      2. symbolTableNode scan\_digit ( std::string in, int\* error )
      3. symbolTableNode scan\_letter ( std::string in )
      4. symbolTableNode scan\_else ( std::string in, int\* error )
2. 语法分析子系统
3. LR(1)分析法

选择SLR(1)，发现冲突过多之后选择完成LR(1)。

完成消除左递归，并添加拓广。



1. 语法分析子系统结构
2. 语法分析子系统的主要数据结构与算法
   1. 数据结构
      1. Struct TreeNode 语法树节点
      2. Struct Production 产生式
      3. Struct Goto goto表相关数据结构
      4. Struct Action action表相关数据结构
      5. Class CFG\_LR1 CFG文法
      6. Class PredictTable 预测表
   2. 算法
      1. 构造项目集
      2. 构造预测表
      3. 分析
3. 三地址代码生成器

• 语法制导定义

• 算法基本思想

1. 实验体会（可选）