

# 系统使用说明书

## 一、功能介绍

该程序是一个用于分析正则表达式，并分析生成对应的 NFA、DFA、最小化 DFA 和词法分析程序的编程语言分析程序，对于输入的编程语言的源程序，程序可以根据该语言的正则表达式生成源程序的编码。

## 二、正则表达式

### 1.1 格式说明

输入正则表达式时，一个表达式占据一行。左侧是正则表达式的名称，然后是一个空格+一个“=”再加上一个空格，即“ = ”，紧接着右侧是表达式的内容。以下是正则表达式的一个格式示例：

`number = DIGITDIGIT*`

表达式支持的表达式有：连接（不使用符号）、选择（|）、可选（?）、闭包（\*）和圆括号。支持对这四种符号使用转义字符，如若想将符号“\*”表示为表达式的一个转换而不是表达式的运算，则可以用“\\*”表示。下图是使用转义的一个实例：



图 1 转义使用实例

正则表达式的命名中，有两种特殊情况：①关键词统一使用“KEYWORD”命名，若关键词是大小写不敏感的，则使用“KEYWORD\*”表示；②对于注释的正则表达式，只需要表示一个注释的开始与结束符号，并且注释的开始符号统一使用命名“L\_COMMENT”，注释的结束符号统一使用命名“R\_COMMENT”。可以允许多种注释方式，但开始符和对应的结束符的表达式必须相邻。特别地，若注释不允许跨行，则表达式表示为“R\_COMMENT = \n”。以下是 TINY 语言中关键字和注释的正则表达式示例：

KEYWORD\* = IF|THEN|ELSE|END|REPEAT|UNTIL|READ|WRITE

L\_COMMENT = {

R\_COMMENT = }

## 1.2 表达式设计建议

需要针对 BNF 文法的每一个终结符号单独编写正则表达式。如 TINY 文法中的“ADDOP -> + | -”，终结符号是“+”和“-”，则需分别对这两个符号编写正则表达式。

表达式的命名最好统一大小写，而名称和使用大写还是使用小写需要根据相

应的 BNF 文法进行设计。如在 TINY 文法中标识符使用小写英文“identifier”表示，则在正则表达式命名中需要保持一致，使用“identifier = ...”；在 mini-c 中标识符是“id”，则 mini-c 的标识符正则表达式则设计为“id = ...”。对于“+”、“-”这类有固定形式的，命名可以任意，但不能和 BNF 文法中的符号相同。

### 三、词法分析程序

生成的词法分析程序使用 c++ 语言，可以单独在 c++ 编译器上运行。允许输入一个源程序代码，使用回车+“EOF”表示程序结束。

### 四、使用方法说明

程序的 UI 界面如下图所示：

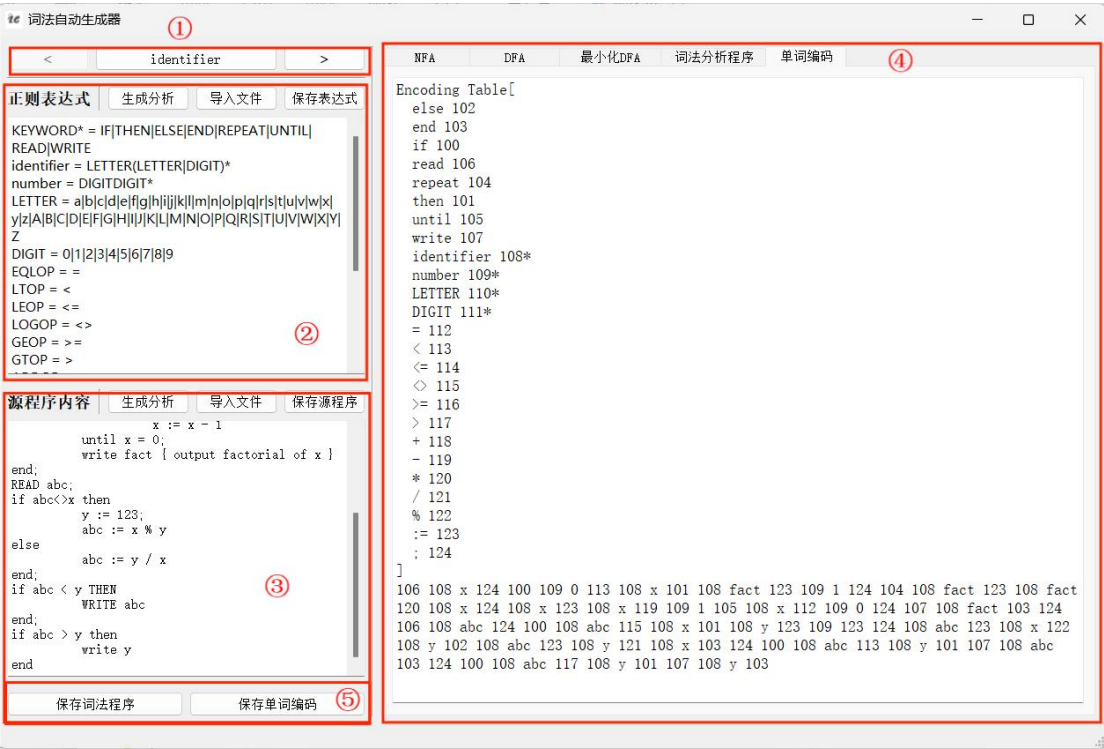


图 2 UI 界面

程序可以分成五个模块来说明。

①当生成正则表达式的分析之后，在模块①处会显示当前表达式的名称，可以点击左右键进行切换，以查看不同表达式的 NFA、DFA 和最小化 DFA（关键字、注释的正则表达式不显示）；

②这是正则表达式的输入模块。可以直接在编辑框输入，也可以点击“导入

文件”按钮打开文件导入。可以通过“保存表达式”按钮保存文本框内的内容。在输入表达式后，点击“生成分析”，即可生成对正则表达式的分析；

③这是源程序的输入模块。可以直接在编辑框输入，也可以点击“导入文件”按钮打开文件导入。可以通过“保存源程序”按钮保存文本框内的内容。在输入表达式后，点击“生成分析”，即可生成对源程序的分析（需要先对正则表达式进行分析，然后才能进行源程序分析）；

④在该模块中，可以查看分析的结果。在上方分为五个分页，可以通过点击分页来切换查看 NFA、DFA、最小化 DFA、词法分析程序和单词编码。

⑤该模块有两个按钮。“保存词法程序”可以保存分析生成的词法分析程序，而“保存单词编码”可以保存生成的编码。