

第一个项目任务要求：

- (1) 以文本文件的方式输入某一高级程序设计语言的所有单词对应的正则表达式，系统需要提供一个操作界面，让用户打开某一语言的所有单词对应正则表达式文本文件，该文本文件的具体格式可根据自己实际的需要进行定义。
- (2) 正则表达式应该可以支持命名操作，运算符有：转义符号(\)、连接、选择(|)、闭包(*)、正闭包(+)、[]、可选(?)、括号()
- (3) 要提供一个源程序编辑界面，让用户输入一行（一个）或多行（多个）正则表达式
(可保存、打开正则表达式文件)
- (4) 需要提供窗口以使用户可以查看转换得到的 NFA（用状态转换表呈现即可）
- (5) 需要提供窗口以使用户可以查看转换得到的 DFA（用状态转换表呈现即可）
- (6) 需要提供窗口以使用户可以查看转换得到的最小化 DFA（用状态转换表呈现即可）
- (7) 将最小化得到 DFA 图转换得到的词法分析源程序（该分析程序需要用 C/C++ 语言描述，而且只能采用讲稿中的转换方法一或转换方法二来生成源程序），系统需提供窗口以使用户可以查看转换得到的词法分析源程序

- (8) 对要求(7)得到的源程序进行编译生成一个可执行程序，并以该高级程序设计语言的一个源程序进行测试，输出该该源程序的单词编码。需要提供窗口以使用户可以查看该单词编码。
- (9) 对系统进行测试：（A）先以 TINY 语言的所有单词的正则表达式作为文本来测试，生成一个 TINY 语言的词法分析源程序；（B）接着对这个词法分析源程序利用 C/C++ 编译器进行编译，并生成可执行程序；（C）以 sample.tny 来测试，输出该 TINY 语言源程序的单词编码文件 sample.lex。
- (10) 要求应用程序为 Windows 界面
- (11) 书写完善的软件文档

补充说明：

- (1) 通过命名中加下划线(_)来表示该正则表达式需要生成 DFA 图。
- (2) 由于整数中的正号 (+) ,在系统已经被用作正比闭包运算符，所以需要通过引入转义符号\来区分。

第二个项目任务要求

- (1) 以文本文件的方式输入某一高级程序设计语言的所有语法对应的 BNF 文法，因此系统需要提供一个操作界面，让用户打开某一语言的所有语法对应的 BNF 文法的文本文件，该文本文件的具体格式可根据自己实际的需要进行定义。
- (2) 求出文法的每个非终结符号的 First 集合和 Follow 集合，并需要提供窗口以使用户可以查看该结果（可用两张表格的形式分别进行呈现）

(3)需要提供窗口以便用户可以查看文法对应的 LR(1)DFA 图。(可以用画图的方式呈现,也可用表格方式呈现该图点与边数据)

(4)需要提供窗口以便用户可以查看文法对应的 LALR(1)DFA 图。(可以用画图的方式呈现,也可用表格方式呈现该图点与边数据)

(5)需要提供窗口以便用户可以查看文法对应的 LALR(1)分析表。(如果该文法为 LALR(1) 文法时)【LALR(1)分析表采用表格的形式呈现】

(6) 打开一个与上述所输入 BNF 文法对应的程序设计语言的源程序,并采用 LALR(1)语法分析方法对这个源程序中的每个语句进行语法分析,需要提供窗口以便用户可以查看对应的语法分析过程。【可以使用表格的形式逐行显示分析过程】

(7) 在 (6) 的分析过程中同时生成相应的语法树,需要提供窗口以便用户可以查看对应的语法分析过程(语法树需要采用树状形式进行呈现)。【每个语句的语法树结构可根据实际的需要进行定义。】

(8) 以 TINY 语言的所有语法以及第一项任务的测试结果 sample.lex 作为测试,并生成对应的语法树并呈现出来。

(9) 要求应用程序为 Windows 界面

(10) 书写完善的软件文档

(11) 选做内容:可以生成某种中间代码【具体的中间代码形式可以自定】。

补充说明:输入的文法均为 2 型文法(上下文无关文法)。文法规则为了处理上的简单,输入时均默认输入的第一个非终结符就是文法的开始符号,用@表示空串。因此下面的文法均为正确的输入。

例 1:

$$E \rightarrow E + T$$
$$T \rightarrow a$$

由于第一个非终结符为 E，因此 E 是文法的开始符号。

例 2:

$$\text{exp} \rightarrow \text{exp addop term} \mid$$
$$\text{term addop} \rightarrow + \mid -$$
$$\text{term} \rightarrow \text{term mulop}$$
$$\text{factor} \mid \text{factor mulop} \rightarrow$$
$$* \mid /$$
$$\text{factor} \rightarrow (\text{exp}) \mid n$$

由于第一个非终结符为 exp，因此 exp 是文法的开始符号。

第三个项目任务要求

mini-c 语言作为测试

- (1) 以 mini-c 的词法进行测试，并以至少一个 mini-c 源程序进行词法分析的测试（该

mini-c 源程序需要自己根据 mini-c 词法和语法编写出来，类似于 sample.tny）。

- (2) 以 mini-c 的语法进行测试，并以测试步骤（1）的源程序所生成的单词编码文件进行语法分析，生成对应的语法树。

二、选做内容：

1.采用 LALR(1)语法分析方法进行语法分析并生成相应的中间代码，中间代码可以自选（可以是四元组、三元组、伪代码等中间代码），中间代码生产结果可以在屏幕上显示或以文件的形式保存