

软件设计文档

20192131018 彭翊桐

一、 软件需求

1. 设计一个应用软件，以实现将正则表达式-->NFA--->DFA-->DFA 最小化-->词法分析程序
1. 正则表达式应该支持单个字符，运算符号有： 连接 选择 闭包 括号
2. 要提供一个源程序编辑界面，让用户输入正则表达式（可保存、打开源程序）
3. 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）
4. 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）
5. 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）
6. 需要提供窗口以使用户可以查看转换得到的词法分析程序（该分析程序需要用 C 语言描述）

二、 总体设计

1. 功能设计

将输入的正则表达式转为 NFA 状态转换表、DFA 状态转换表、最小化 DFA 状态转换表和词法分析程序。

2. 界面设计

- 1) 正则表达式编辑框，显示输入的正则表达式，用户可以直接在编辑框输入正则表达式，也可以点击“读取”按钮选择文件，使用文件中的正

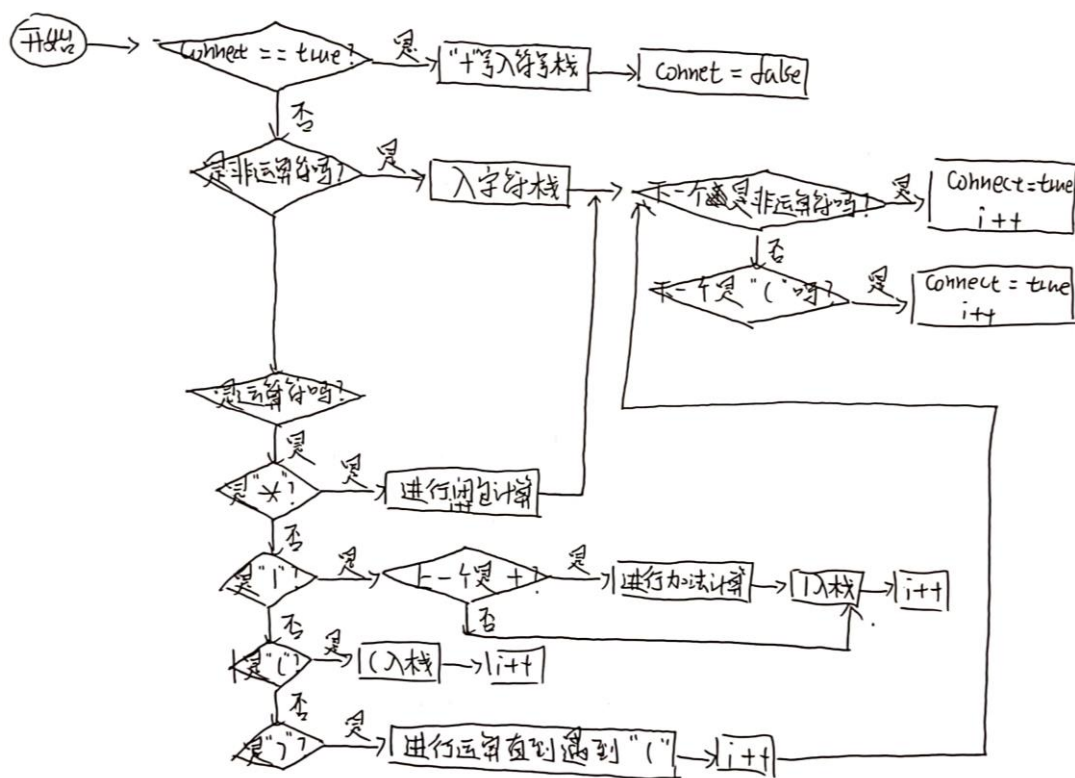
则表达式。

- 2) “读取”按钮，点击后，弹出文件选择对话框，通过对话框可以选择需要打开的文件。选择后，该文件中的正则表达式将被输入正则表达式编辑框中。
- 3) “保存”按钮，点击后，弹出文件选择对话框，通过对话框可以选择需要保存的位置。选择后，正则表达式编辑框中的正则表达式将被保存到指定位置。
- 4) “生成”按钮，点击后，程序将分析正则表达式编辑框中的正则表达式，并将结果显示在 NFA 状态转换表、DFA 状态转换表、最小化 DFA 状态转换表和词法分析程序编辑框中。
- 5) NFA 状态转换表，该表格显示由正则表达式生成的 NFA 状态转换表。
- 6) DFA 状态转换表，该表格显示由正则表达式生成的 DFA 状态转换表。
- 7) 最小化 DFA 状态转换表，该表格显示由正则表达式生成的最小化 DFA 状态转换表。
- 8) 词法分析程序编辑框，该编辑框显示由正则表达式生成的词法分析程序。

3. 逻辑设计

本程序的所有转换过程均基于状态转移表实现。

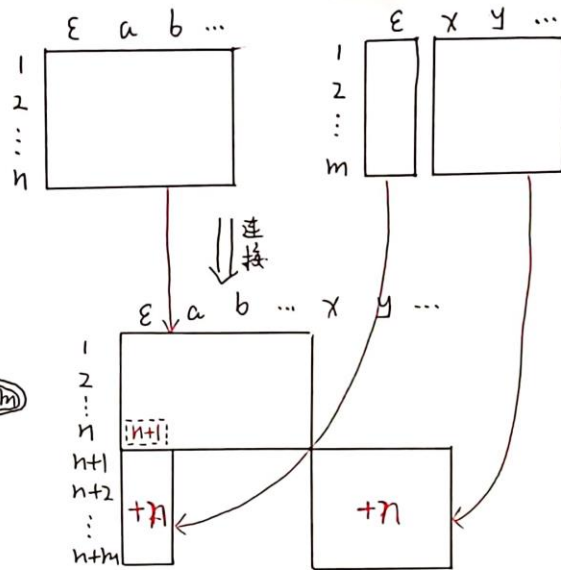
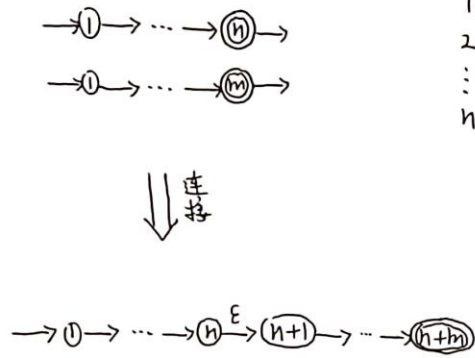
- 1) 正则表达式转 NFA
 - a) 解析算数表达式，过程如下：



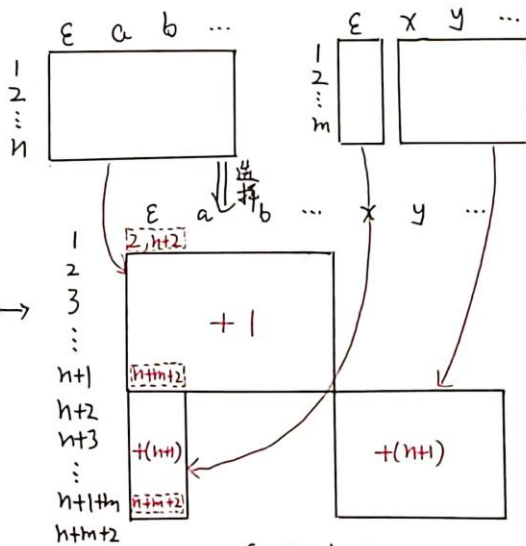
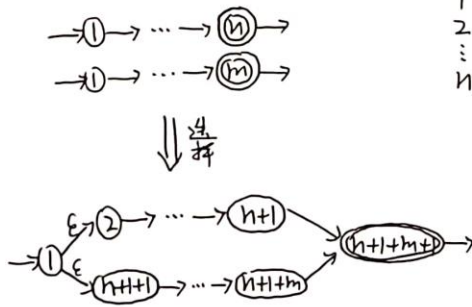
b) 进行选择、连接、闭包计算

选择、连接、闭包运算均通过 NFA 状态转移表进行，规律如下：

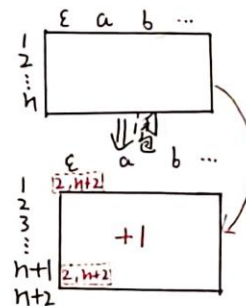
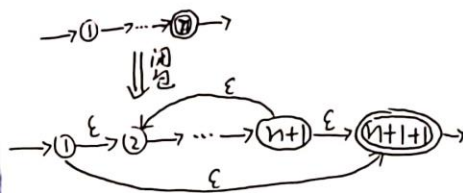
连接运算



选择运算



闭包运算



2) NFA 转 DFA

首先，计算出初态的闭包，并将该状态集加入队列。取出队列头的元素，计算该状态集转换后的状态集，将全新的状态集入队，并将队列头元素和转移后的状态集作为 DFA 状态转移表的新的一行，重复上

述过程直到队列为空，即可得到 DFA 状态转移表。

3) DFA 最小化

建立一个临时表，该表存储状态集简化为单个状态后的 DFA 状态转移表。建立两个集合，分别存储简化后的终态和非终态。根据 DFA 最小化的规则，在临时表中判断并合并相同的状态。最后将结果存储最小化 DFA 状态转移表中。

4) DFA 转词法分析代码

根据 DFA 转词法分析代码的规则，使用最小化 DFA 状态转移表，非终态和终态集合，自动生成由 C++ 编写的 lexicalAnalysis 函数。该函数的参数类型为 String 类型，返回值为 bool 类型，返回 true 表示接收，而返回 false 表示出错。

三、 开发环境和工具

Visual Studio 2019 with C++ MFC