



杭州电子科技大学
HANGZHOU DIANZI UNIVERSITY

实验题目	有限自动机的确定化和最小化
学号	16051216
姓名	李国夏
学院	卓越学院
专业	理工类实验班
指导老师	黄孝喜

二〇一九年一月一日

1 实验内容

利用状态表和有限自动机的运行原理编写和设计程序，判断输入的自动机是 DFA 还是 NFA，如果是 NFA，利用子集法将其确定化，然后利用求同法或求异法将所得的 DFA 最小化。

2 实验目的

1. 理解有限自动机的作用，进一步理解有限自动机理论
2. 设计有限自动机的表示方式，采用合理的数据结构表示自动机的五个组成部分
3. 掌握 ε 闭包的求法和子集构造算法，以程序实现 NFA 到 DFA 的转换，并且最小化 DFA，提高算法的理解和实现能力

3 实验步骤

1. 可以采用任何语言来完成，如：C、C++或 JAVA 等
2. 建议以文本文件形式来描述自动机，例如：
 第一行：表示状态个数
 第二行开始表示为状态转换表
 最后一行给出接受状态列表
3. 根据读进去的自动机内容，判断其类别（NFA 还是 DFA？）
4. 若是 NFA，利用子集法将其确定化
5. 将 DFA 最小化
6. 输入测试符号串，输出测试结果。

4 实验过程

4.1 程序框图及算法流程

本程序的程序框图及算法流程如图 1 所示。

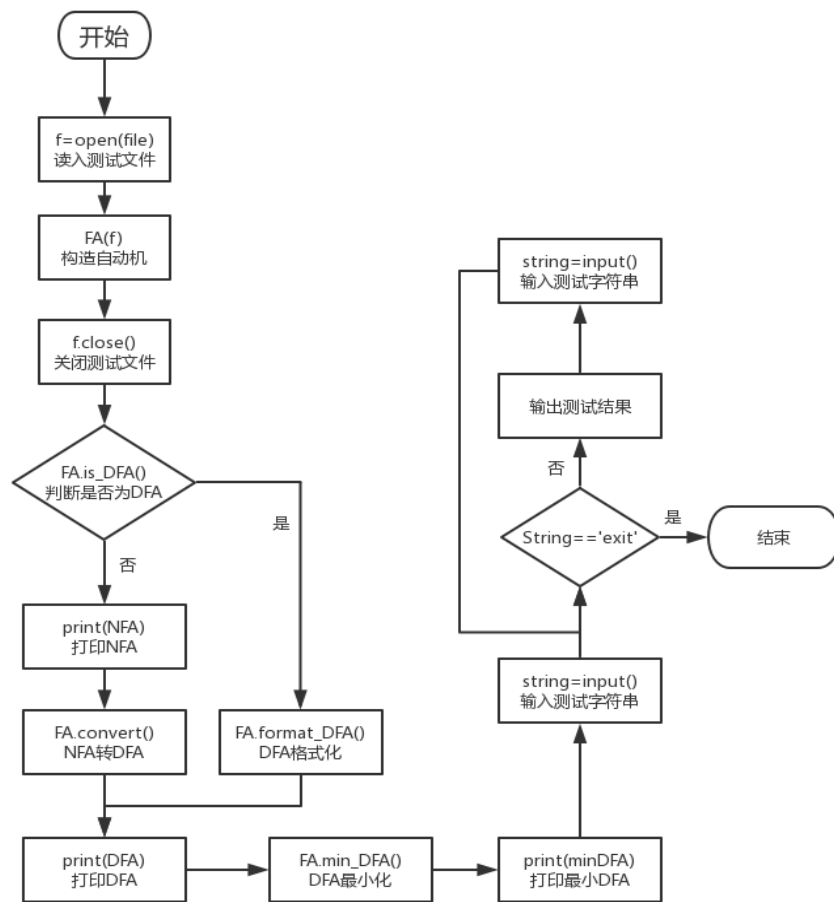


图 1 程序框图及算法流程

4.2 实验结果

测试文件说明：状态编号从 0 开始，第一排表示状态总数，最后一排表示终结状态列表，以空格分割。第 $i + 1$ 排有若干对(字符 c , 状态编号 j)，表示状态 i 接受输入 c , 转移到状态 j , 若第 $i + 1$ 排为空，表示状态 i 没有出边。用字符 '0' 表示 ε 。

测试文件一：

测试样例文件名	NFA.txt
测试样例内容	<pre> NFA.txt - 记事本 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H) 11 0 1 0 7 0 2 0 4 a 3 0 6 b 5 0 6 0 1 0 7 a 8 b 9 b 10 10 </pre>

测试自动机图示	
预计的最小化 DFA 图示	
测试正则	$(a b)^*abb$
实际程序运行结果	<pre> D:\Anaconda3\python.exe This is New DFA converted from NFA (5, {0: {'a': 1, 'b': 2}, 1: {'a': 1, 'b': 3}, 2: {'a': 1, 'b': 2}, 3: {'a': 1, 'b': 4}, 4: {'a': 1, 'b': 2}}, [4]) [[4], [3], [1], [0, 2]] This is minDFA (4, {0: {'a': 1, 'b': 0}, 1: {'a': 1, 'b': 3}, 3: {'a': 1, 'b': 4}, 4: {'a': 1, 'b': 0}}, [4]) 输入测试字符串(输入exit则结束):abb abb is YES 输入测试字符串(输入exit则结束):aabb aabb is YES 输入测试字符串(输入exit则结束):babb babb is YES 输入测试字符串(输入exit则结束):abaabbabb abaabbabb is YES 输入测试字符串(输入exit则结束):ababa ababa is NO 输入测试字符串(输入exit则结束):exit >>> </pre>

测试文件二：

测试样例文件名	NFA2.txt
测试样例内容	<pre> 10 a 1 0 2 0 8 0 3 0 5 a 4 0 7 b 6 0 7 0 2 0 8 b 9 9 </pre>

测试自动机图示	
预计的最小化 DFA 图示	
测试正则	$a(a b)^*b$
实际程序运行结果	<pre> D:\Anaconda3\python.exe This is NFA {0: {'a': [1]}, 1: {'0': [2, 8]}, 2: {'0': [3, 5]}, 3: {'a': [4]}, 4: {'0': [7]}, 5: {'b': [6]}, 6: {'0': [7]}, 7: {'0': [2, 8]}, 8: {'b': [9]}, 9: {}} This is New DFA converted from NFA (4, {0: {'a': 1}, 1: {'a': 2, 'b': 3}, 2: {'a': 2, 'b': 3}, 3: {'a': 2, 'b': 3}}, [3]) [[3], [1, 2], [0]] This is minDFA (3, {0: {'a': 1}, 1: {'a': 1, 'b': 3}, 3: {'a': 1, 'b': 3}}, [3]) 输入测试字符串(输入exit则结束):ab ab is YES 输入测试字符串(输入exit则结束):aaaab aaaab is YES 输入测试字符串(输入exit则结束):aaa aaa is NO 输入测试字符串(输入exit则结束):b b is NO 输入测试字符串(输入exit则结束):aab aab is YES 输入测试字符串(输入exit则结束):b b is NO 输入测试字符串(输入exit则结束):a a is NO 输入测试字符串(输入exit则结束):exit </pre>

测试文件三：

测试样例文件名	DFA.txt
测试样例内容	<pre> 5 a 1 b 2 a 1 b 3 a 1 b 2 a 1 b 4 a 1 b 2 4 </pre>

测试自动机图 示	
预计的最小化 DFA 图示	
测试正则	$(a b)^*abb$
实际程序运行 结果	<pre> D:\Anaconda3\python.exe This is DFA (5, {0: {'a': 1, 'b': 2}, 1: {'a': 1, 'b': 3}, 2: {'a': 1, 'b': 2}, 3: {'a': 1, 'b': 4}, 4: {'a': 1, 'b': 2}}, [4]) [[4], [3], [1], [0, 2]] This is minDFA (4, {0: {'a': 1, 'b': 0}, 1: {'a': 1, 'b': 3}, 3: {'a': 1, 'b': 4}, 4: {'a': 1, 'b': 0}}, [4]) 输入测试字符串(输入exit则结束):ab ab is NO 输入测试字符串(输入exit则结束):abb abb is YES 输入测试字符串(输入exit则结束):aabb aabb is YES 输入测试字符串(输入exit则结束):abba abba is NO 输入测试字符串(输入exit则结束):aabbb aabbb is NO 输入测试字符串(输入exit则结束):aabbabb aabbabb is YES 输入测试字符串(输入exit则结束):exit </pre>

5 实验总结

本次实验的收获主要在以下几个方面: 1. 选用合适的数据结构(Python 中的字典和列表)来表示自动机。2. NFA 转 DFA 中利用广度优先遍历实现闭包的运算。3. 在实现最小化 DFA 时采用求异法, 实际上可以类比树的遍历, 加深了对求异法的理解和使用。

其他都是按部就班地通过课程 PPT 实现相应算法, 收获较小, 仅仅增加了熟练度。

另外采用 Python 语言可以实现快速开发, 然后用面向对象的思想来封装程序, 使程序更加简洁, 这是编程方面的收获。

最后, 从心理上来说, 通过本次实验, 大大增加了学习编译原理的信心和成就感, 提高了对这门课后续学习和研究的兴趣。