

第三章作业

第一次作业:

练习 3.2.1, 本题中我们把所有的符号归结为 **op**, 实际当中是可以再细分的, 同学们只要回答的有道理即可

```
<float, >
<id, 指向 limitedSquare 符号表项指针>
<(<, >
<id, 指向 x 符号表指针>
<)>, >
<{, >
<float, >
<id, 指向 x 符号表指针>
<return, >
<(<, >
<id, 指向 x 符号表指针>
<op, "<=">
<num, -10.0>
<op, "||">
<id, 指向 x 符号表指针>
<op, ">=">
<num, 10.0>
<)>, >
<op, "?">
<num, 100>
<op, ":">
<id, 指向 x 符号表指针>
<op, "*">
<id, 指向 x 符号表指针>
<}>, >
```

练习 3.3.2

1) $a(a|b)^*a$

表示以 a 开头以 a 结尾且至少包含两个字符的由 a 和 b 构成的字符串的集合

2) $((\epsilon|a)b^*)^*$

表示由 a 和 b 构成的字符串的集合

5) $(aa|bb)^*((ab|ba)(aa|bb)^*(ab|ba)(aa|bb)^*)^*$

表示含有偶数个 a 和偶数个 b 的由 a 和 b 构成的字符串的集合

练习 3.3.5

提示：8、9 可以先画出状态转换图，再根据状态转换图写正则表达式（1、8 的状态图见第二次作业）。8、9 注意“子串”跟“子序列”的区别。“子串”要求连续，“子序列”不要求连续，只需要保证单调性。e.g. “aabab”不包含子串 abb 但是包含子序列 abb

注意正则表达式中的几个特殊符号：

“*”表示出现任意次，包含 0 次，

“+”表示至少出现一次，

“?”表示至多出现一次

1) 包含 5 个元音的所有小写字母串，这些串中的元音按顺序出现

str \rightarrow other* a (other|a)* e (other|e)* i (other|i)* o (other|o)* u (other|u)*

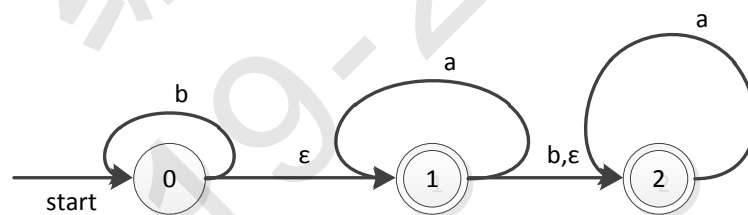
other \rightarrow [bcdfghijklmnpqrstvwxyz]

8) 所有由 a 和 b 组成且不含子串 abb 的串

str $\rightarrow b^*(a|ab)^*$ 或者 str $\rightarrow b^*(a+ab?)^*$

9) 所有由 a 和 b 组成且不含子序列 abb 的串

ANS:思路如下，不包含子序列 abb，可以尝试着枚举几种典型的情况，状态图如下：



$b^*a^*(b|\epsilon)a^*$ 和 $b^*a^*(a|b)a^*|\epsilon$

即

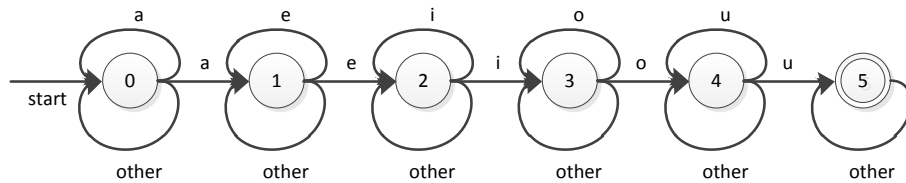
str $\rightarrow b^* | b^*a^+ | b^*a^+ba^*$

第二次作业:

1.D FA/NFA 设计

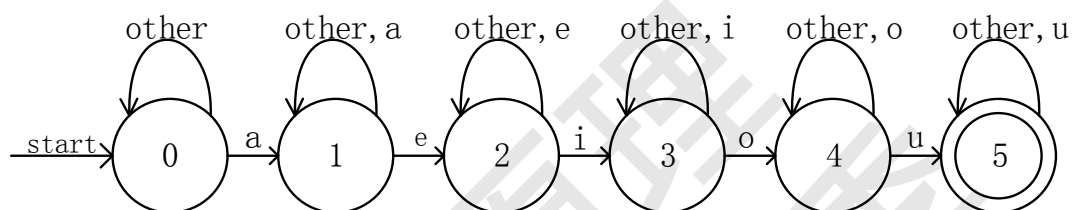
为下面的语言设计一个 DFA 或 NFA。

- 1) 包含 5 个元音的所有小写字母串，这些串中的元音按顺序出现
NFA 的一种:



上图中 other 表示除去元音字母以外剩下的 21 个字母。

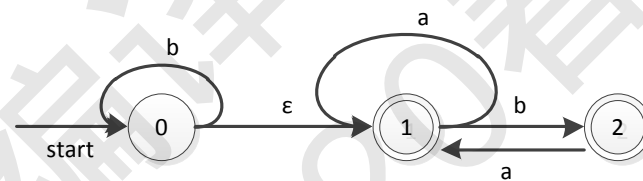
DFA 的一种:



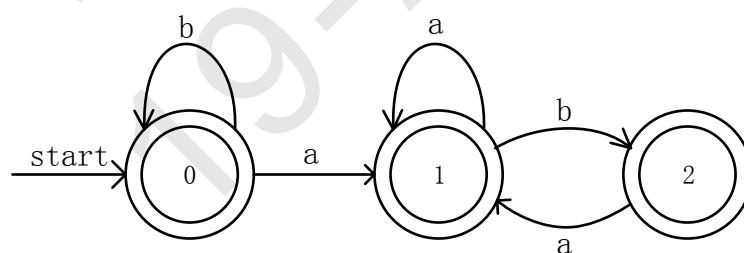
上图中 other 表示除去元音字母以外剩下的 21 个字母。

- 2) 所有由 a 和 b 组成且不含子串 abb 的串

ANS:枚举几种典型的情况，不难画出如下 NFA:



等价的 DFA 如下图所示。



2.NFA 模拟输入

用算法 3.22 模拟图 3-29 中的 NFA 在处理输入 aabb 时的过程。(算法 3.22 见龙书第三版 P99)

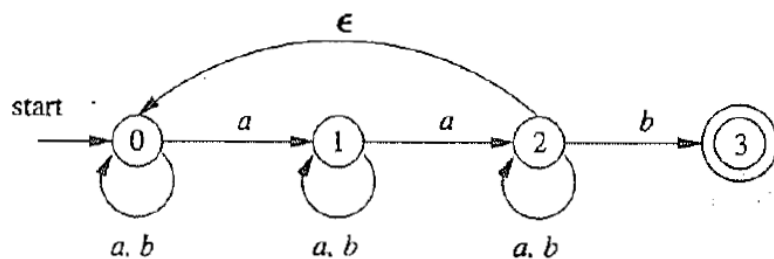


图 3-29 练习 3.6.3 的 NFA

ANS:

$F=\{3\}, S=\epsilon\text{-closure}(0)=\{0\}, c='a'$

$S=\epsilon\text{-closure}(\text{move}(\{0\}, 'a'))=\{0,1\}, c='a'$

$S=\epsilon\text{-closure}(\text{move}(\{0,1\}, 'a'))=\{0,1,2\}, c='b'$

$S=\epsilon\text{-closure}(\text{move}(\{0,1,2\}, 'b'))=\{0,1,2,3\}, c='b'$

$S=\epsilon\text{-closure}(\text{move}(\{0,1,2,3\}, 'b'))=\{0,1,2,3\}, c=\text{eof}$

$S \cap F \neq \text{null}, \therefore \text{return "yes"}$

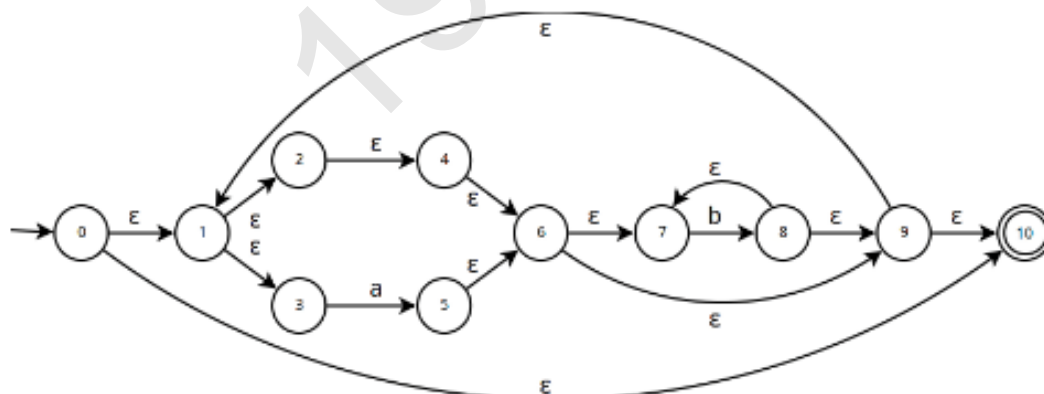
3.DFA 化简

使用算法 3.23 和 3.20 将下述正则表达式转换为 DFA，并尝试化简该 DFA

1) $((\epsilon \mid a)b^*)^*$

ANS:

NFA 如下:

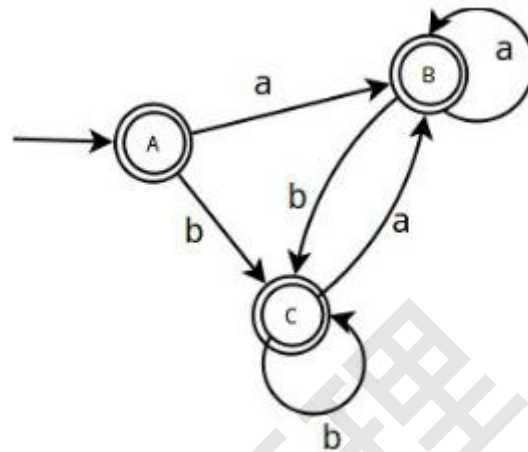


转换表如下:

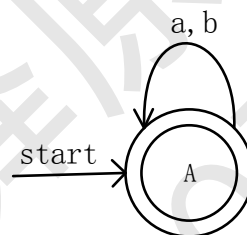
NFA 状态	DFA 状态	输入符号	
		a	b

{0,1,2,3,4,6,7,9,10}	A	B	C
{1,2,3,4,5,6,7,9,10}	B	B	C
{1,2,3,4,6,7,8,9,10}	C	B	C

转换后的 DFA 如下：



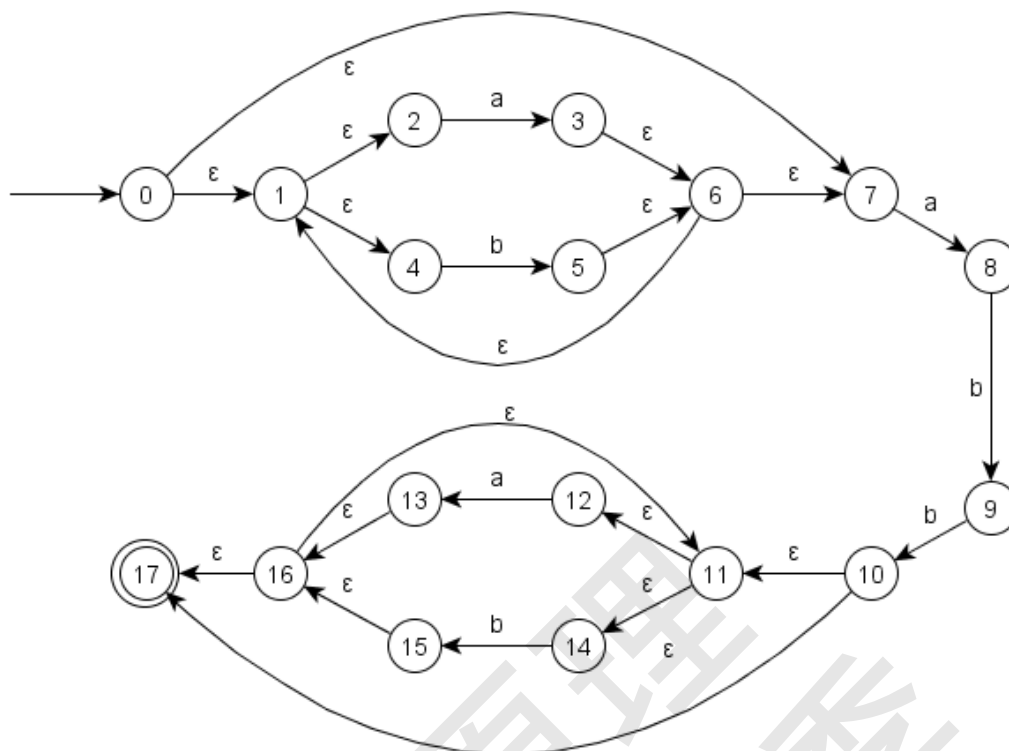
经过化简，最终可得化简后 DFA 如下图所示。



2) $(a \mid b)^*abb(a \mid b)^*$

ANS:

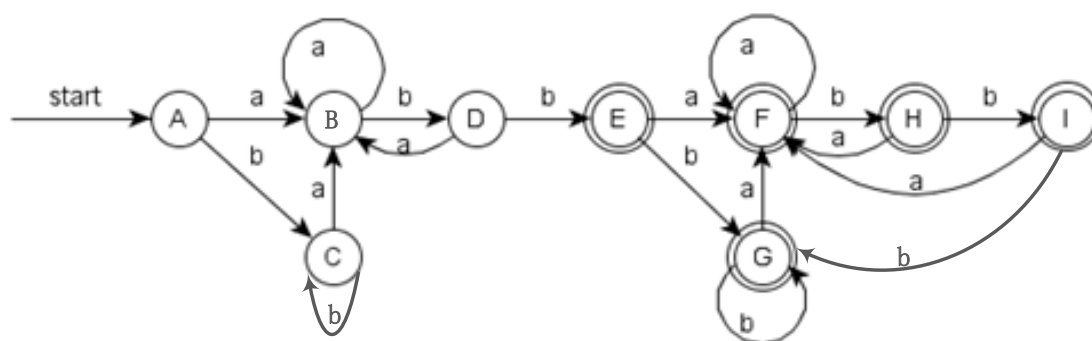
NFA 如下：



转换表如下：

NFA 状态	DFA 状态	输入符号	
		a	b
{0,1,2,4,7}	A	B	C
{1,2,3,4,6,7,8}	B	B	D
{1,2,4,5,6,7}	C	B	C
{1,2,4,5,6,7,9}	D	B	E
{1,2,4,5,6,7,10,11,12,14,17}	E	F	G
{1,2,3,4,6,7,8,11,12,13,14,16,17}	F	F	H
{1,2,4,5,6,7,11,12,14,15,16,17}	G	F	G
{1,2,4,5,6,7,9,11,12,14,15,16,17}	H	F	I
{1,2,4,5,6,7,10,11,12,14,15,16,17}	I	F	G

DFA 如下：



经过化简，最终可得化简后 DFA 如下图所示。

