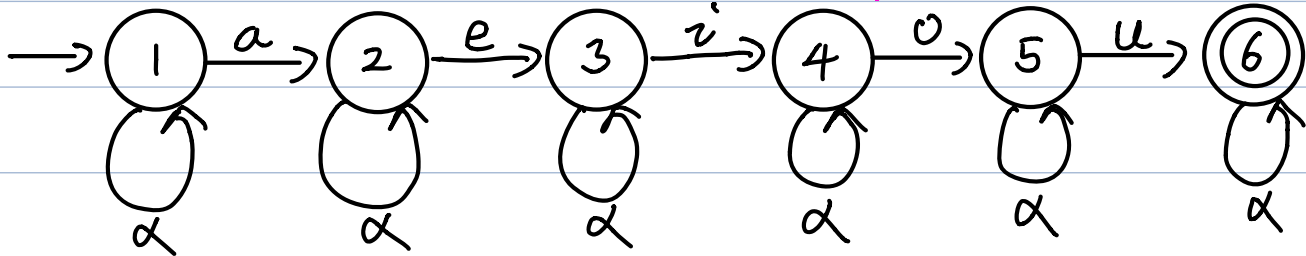


1. 解: 为下面的语言设计一个 DFA 或 NFA.

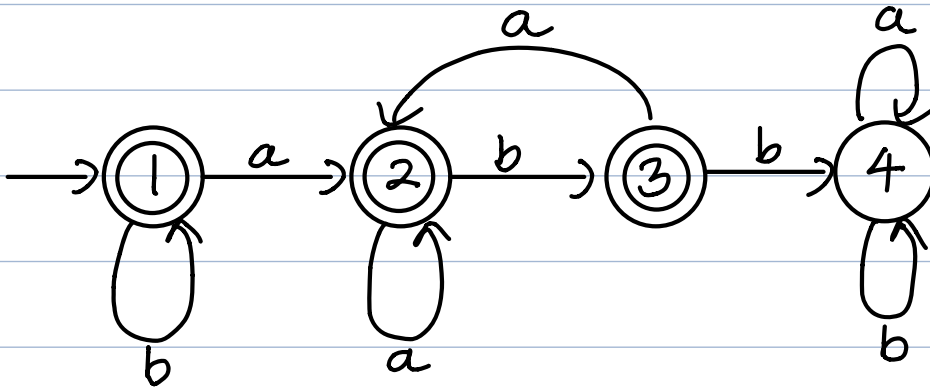
1) 包含 5 个元音的所有小写字母串, 这些串中的元音按顺序出现.

DFA: 其中 α 为任意小写字母串.



2) 所有由 a 和 b 组成且不含子串 abb 的串.

DFA:



2. 解: 用算法 3.22 模拟图 3-29 中的 NFA 在处理输入 aabb 时的过程.

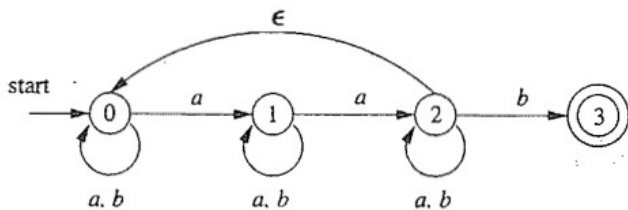


图 3-29 练习 3.6.3 的 NFA

```

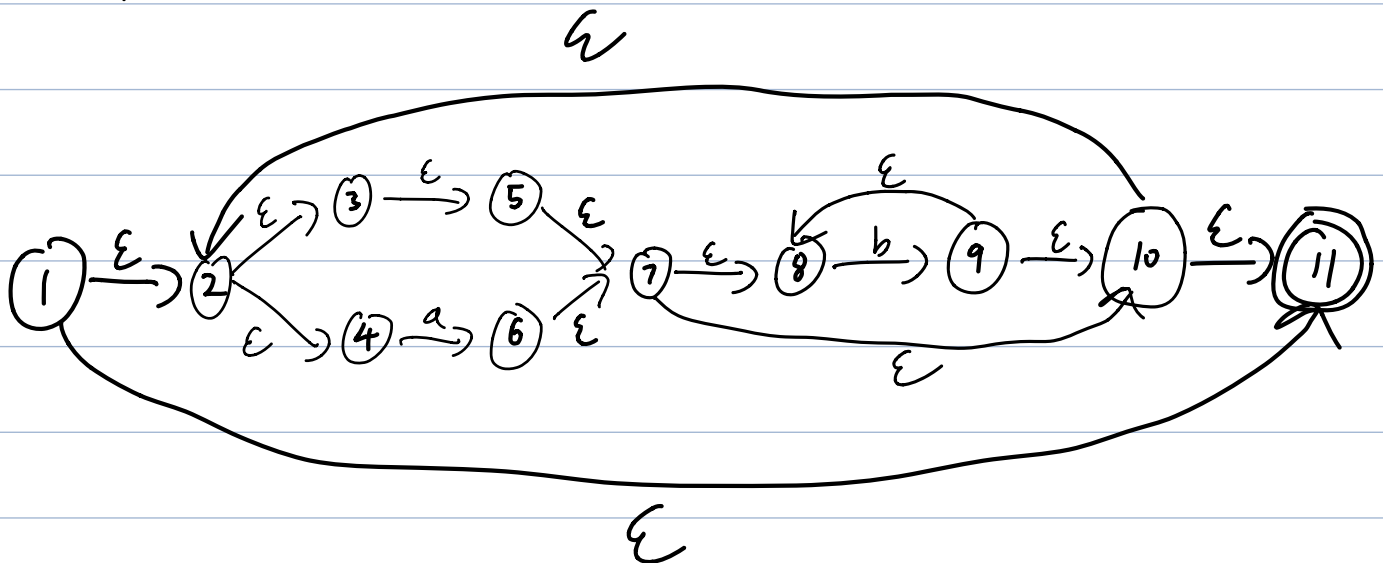
1) S =  $\epsilon$ -closure( $s_0$ );
2) c = nextChar();
3) while ( c != eof ) {
4)     S =  $\epsilon$ -closure(move(S, c));
5)     c = nextChar();
6) }
7) if ( S  $\cap$  F  $\neq \emptyset$  ) return "yes";
8) else return "no";
  
```

- ① $S = \epsilon\text{-closure}(\{0\}) = \{0\};$
- ② $c = a;$
- ③ $S = \epsilon\text{-closure}(\text{move}(S, c)) = \{0, 1\};$
- ④ $c = a;$
- ⑤ $S = \epsilon\text{-closure}(\text{move}(S, c)) = \{0, 1, 2\};$
- ⑥ $c = b;$
- ⑦ $S = \epsilon\text{-closure}(\text{move}(S, c)) = \{0, 1, 2, 3\};$
- ⑧ $c = b;$
- ⑨ $S = \epsilon\text{-closure}(\text{move}(S, c)) = \{0, 1, 2, 3\};$
- ⑩ $c = \text{EOF};$
- ⑪ $\{3\} \in S, \text{ return "yes"};$

3. 解: 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) $((\epsilon|a)b^*)^*$

NFA:



NFA状态

$\{1, 2, 3, 4, 5, 7, 8, 10, 11\}$

$\{2, 3, 4, 5, 6, 7, 8, 10, 11\}$

$\{2, 3, 4, 5, 7, 8, 9, 10, 11\}$

DFA状态

A

B

C

a

B

B

B

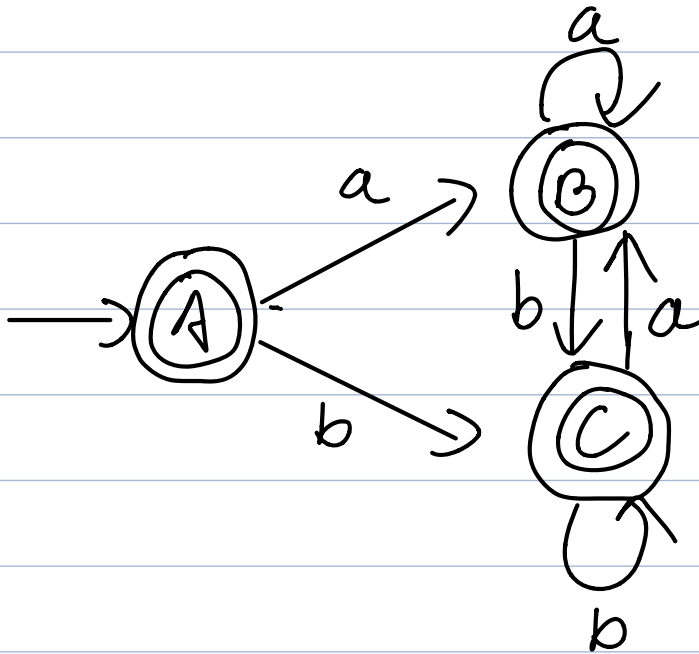
b

C

C

C

DFA:



2) $(a|b)^*abb(a|b)^*$

DFA:

