

# 习题解答HW1

周瓯翔

Fall 2024



# 说明

- 每道题目的解答有多页，在第一页给出给分细则
- 单次作业满分10分

题目	2.1 a	2.3 d	2.4 e	2.4 g	2.7 b	2.15
满分	1'	1'	2'	2'	2'	2'

## 2.1 a

题目 从 C 语言的参考手册确定它们形成输入字母表的字符集（不包括那些只可以出现在字符串或注释中的字符）。

源字符集（c语言源文件所用字符集）

- 26个大写字母: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
- 26个小写字母: a b c d e f g h i j k l m n o p q r s t u v w x y z
- 10个阿拉伯数字: 0 1 2 3 4 5 6 7 8 9
- 特殊字符: ! " # % & ' ( ) \* + , - . / : ; < = > ? [ \ ] ^ \_ { | } ~
- 空白字符: 空格、水平制表符、垂直制表符、换页符、换行符

参见[C \(programming language\) - Wikipedia](https://en.cppreference.com/w/cpp/string/basic/basic_char_traits)

## 2.3 d



题目叙述由下列正规式描述的语言： $0^*10^*10^*10^*$

恰好包含3个1的0和1的串。

## 2.4 e

题目 为下列语言写出正规定义：最多只有一处相邻数字相同的所有数字串

$$S_0 \rightarrow 0$$

$$S_1 \rightarrow S_0 \mid S_0?1(S_01)^*S_0?$$

...

$$S_9 \rightarrow S_8 \mid S_8?9(S_89)^*S_8?$$

$$ANSWER \rightarrow S_9?S_9?$$

另一种可能的写法(不使用?符号)是

$$S_0 \rightarrow 0$$

$$S_1 \rightarrow S_0 \mid (S_0 \mid \varepsilon)1(S_01)^*(S_0 \mid \varepsilon)$$

...

$$S_9 \rightarrow S_8 \mid (S_8 \mid \varepsilon)9(S_89)^*(S_8 \mid \varepsilon)$$

$$ANSWER \rightarrow (S_9 \mid \varepsilon)(S_9 \mid \varepsilon)$$

## 2.4 g

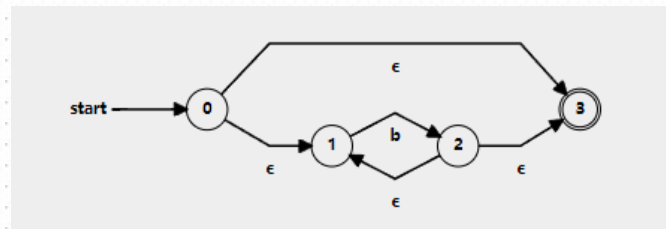
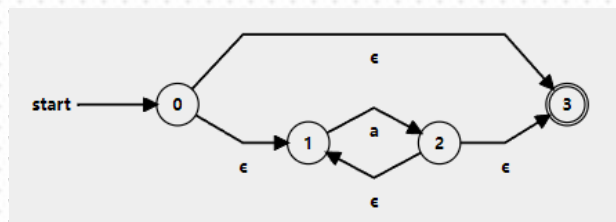
题目 为下列语言写出正规定义：由偶数个0和奇数个1构成的所有0和1的串。

$$\text{even\_0\_even\_1} \rightarrow (00 \mid 11)^* ((01 \mid 10)(00 \mid 11)^*(01 \mid 10)(00 \mid 11)^*)^*$$
$$\text{even\_0\_odd\_1} \rightarrow 1 \text{ even\_0\_even\_1} \mid 0(00 \mid 11)^*(01 \mid 10) \text{ even\_0\_even\_1}$$

## 2.7 c

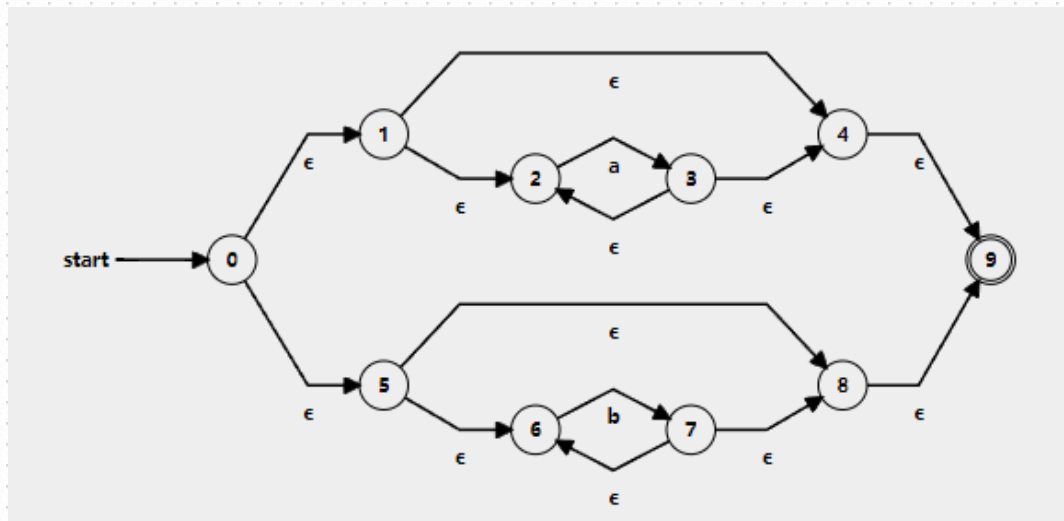
题目 用算法 2.4 为下列正规式构造不确定有限自动机，给出它们处理输入串 **ababbab** 的状态转换序列：  
 $(a^* | b^*)^*$

构造NFA的过程：



$a^*$

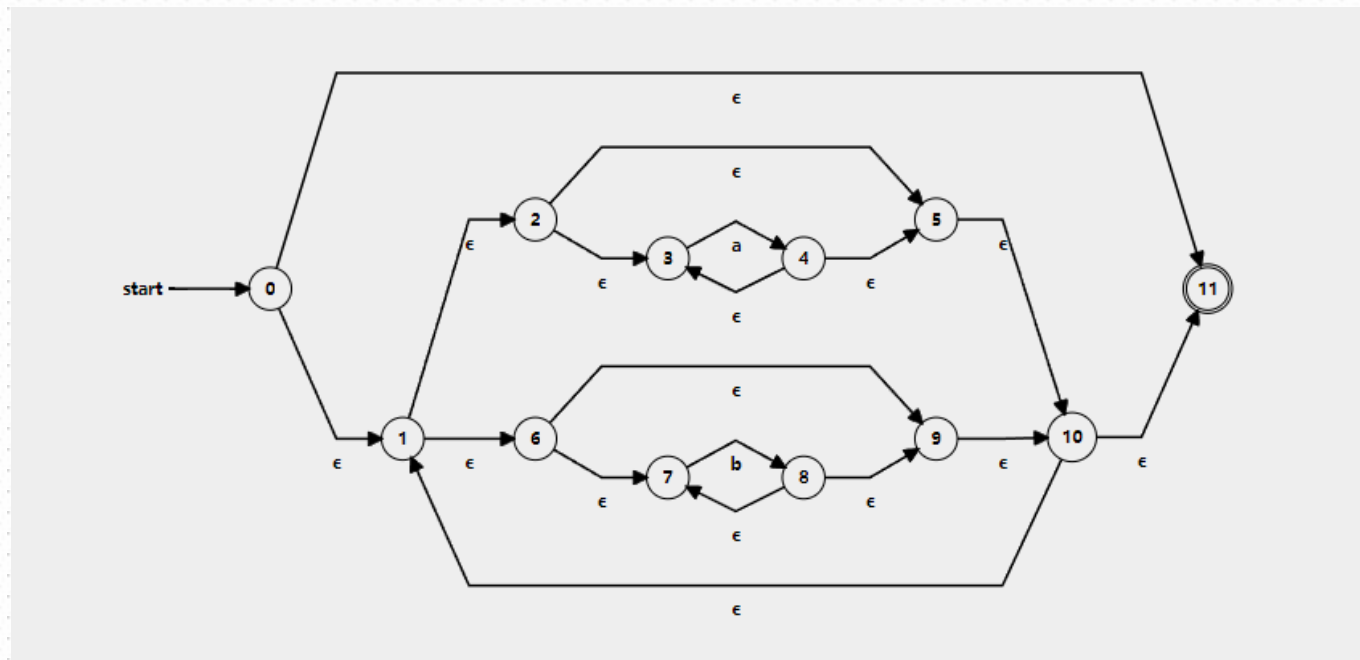
$b^*$



$a^* | b^*$

## 2.7 c (Cont)

题目 用算法 2.4 为下列正规式构造不确定有限自动机，给出它们处理输入串 **ababbab** 的状态转换序列： $(a^* | b^*)^*$



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

说明：根据题目要求，需要按照算法 2.4 构造，考试中如果未要求，可以构造更简单的NFA

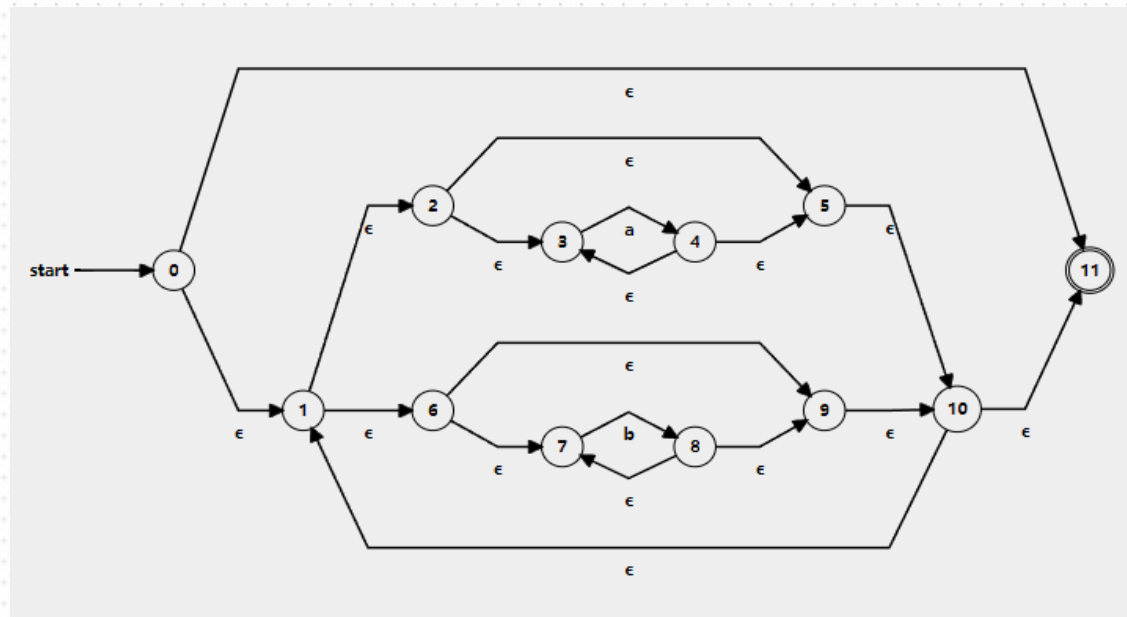


## 2.7 c (Cont)

题目 用算法 2.4 为下列正规式构造不确定有限自动机，给出它们处理输入串 **ababbab** 的状态转换序列：  
 $(a^* | b^*)^*$

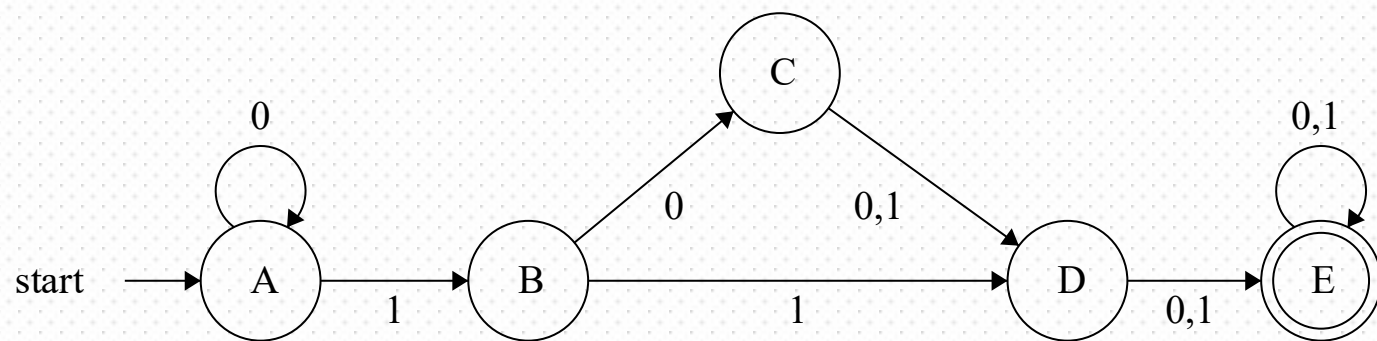
输入 ababbab 时的状态转换序列：

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 10$	// 识别 a
$\rightarrow 1 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10$	// 识别 b
$\rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 10$	// 识别 a
$\rightarrow 1 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10$	// 识别 bb
$\rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 10$	// 识别 a
$\rightarrow 1 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10$	// 识别 b
$\rightarrow 11$	



## 2.15

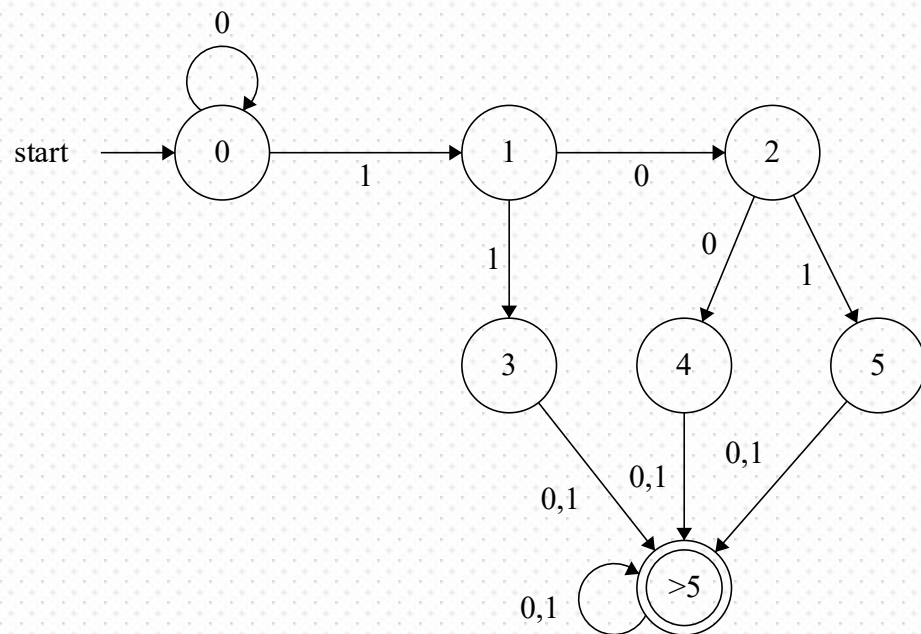
题目 构造一个最简的 DFA，它接受所有大于101的二进制整数。



## 2.15(Cont)

题目 构造一个最简的 DFA，它接受所有大于101的二进制整数。

根据输入，可以判断当前得到的数字，  
手工构造如下DFA，状态名代表了当前得到的数字：



## 2.15 (Cont)

题目 构造一个最简的 DFA，它接受所有大于101的二进制整数。

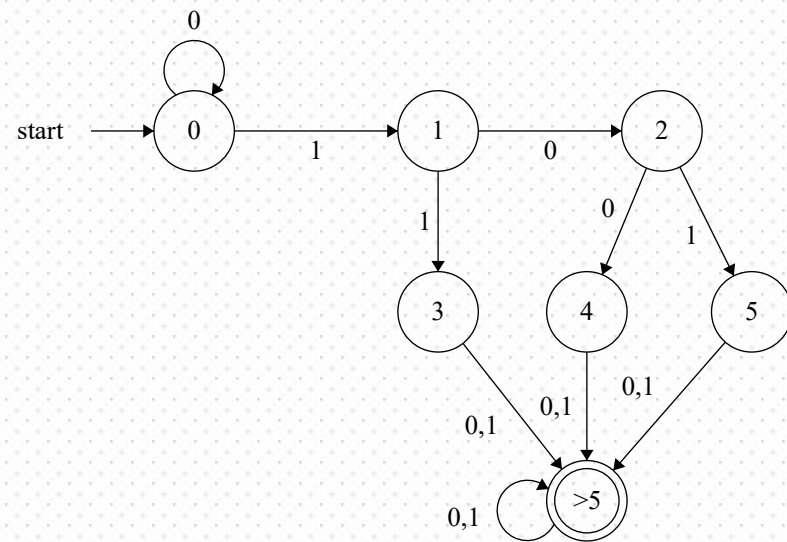
然后根据算法 2.3 得到极小DFA

1. 首先检查状态转换函数是否为**全函数**:

任意状态都存在对 0 或 1 的转换

2. 划分状态子集, 得到 **F** 和 **S - F**

$F = \{ ">5" \}$   $S - F = \{ 0, 1, 2, 3, 4, 5 \}$



## 2.15 (Cont)

题目 构造一个最简的 DFA，它接受所有大于101的二进制整数。

然后根据算法 2.3 得到极小DFA

### 3. 构造新的划分

F 中只有一个状态，不可再分: { ">5" }

考察  $S - F = \{ 0, 1, 2, 3, 4, 5 \}$

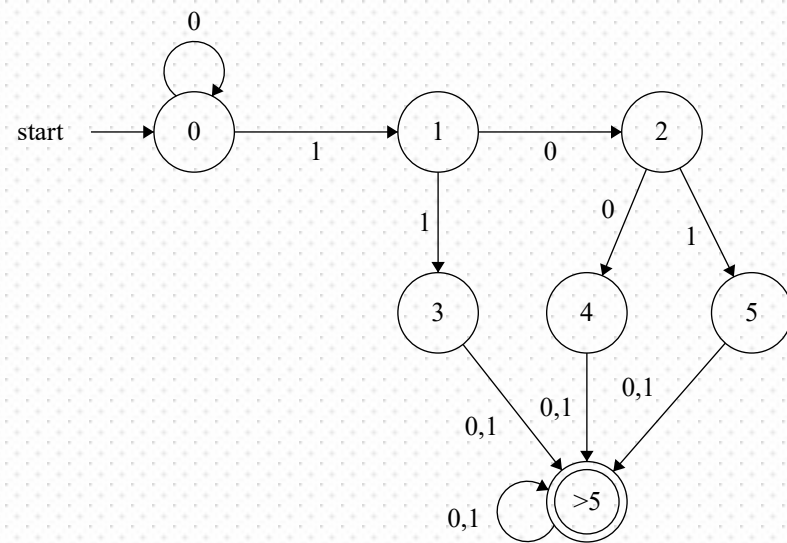
对于输入 0, 状态 0、1、2 分别转换到 0、2、4

而状态 3、4、5 均转换到 ">5"

所以将  $S - F$  划分为: { 0 } { 1 } { 2 } { 3, 4, 5 }

状态 3、4、5 的 0 转换与 1 转换都相同 (均到达状态 ">5" )

故子集 { 3, 4, 5 } 不可再分



## 2.15 (Cont)

题目 构造一个最简的 DFA，它接受所有大于101的二进制整数。

然后根据算法 2.3 得到极小DFA

### 4. 极小 DFA 的状态转换表

状态	输入符号	
	0	1
A ({0})	A	B
B ({1})	C	D
C ({2})	D	D
D ({3, 4, 5})	E	E
E ({6})	E	E

## 2.15 (Cont)

题目 构造一个最简的 DFA，它接受所有大于101的二进制整数。

然后根据算法 2.3 得到极小DFA

5. 极小 DFA 的状态转换图

