

习题 3.1.1 写出表示下列语言的正则表达式:

- 字母表  $\{a, b, c\}$  上包含至少一个  $a$  和至少一个  $b$  的串的集合。
- 倒数第 10 个符号是 1 的 0 和 1 的串的集合。
- 至多只有一对连续 1 的 0 和 1 的串的集合。

3-1-1(a)  $\{a, b, c\}$  上至少 1 个  $a$  和至少 1 个  $b$

$$ab + ba$$

$$\Rightarrow (a+b+c)^* a (a+b+c)^* b (a+b+c)^* + (a+b+c)^* b (a+b+c)^* a (a+b+c)^*$$

$$\Rightarrow (a+b+c)^* [a(a+b+c)^* b + b(a+b+c)^* a] (a+b+c)^*$$

$$(b + (0+1)^2)^* (0+1)^9$$

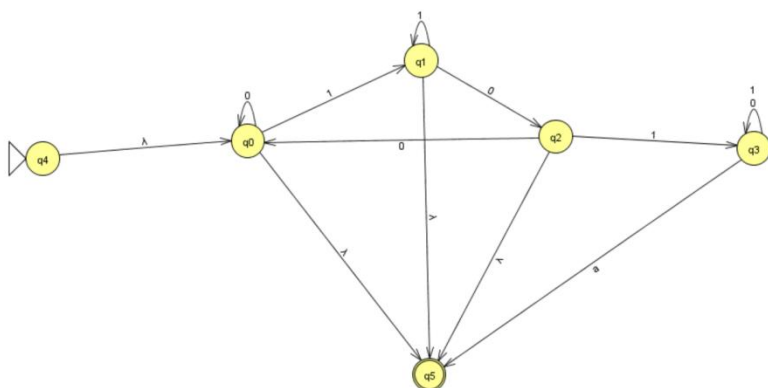
1. c) 连续 1 的串  $\rightarrow$  有 11 或无连续的 1  
 没有 111 (2 对连续的 1)

$$\therefore (0+01)^* (11+1+\epsilon) (0+10)^*$$

习题 3.1.3 写出表示下列语言的正则表达式:

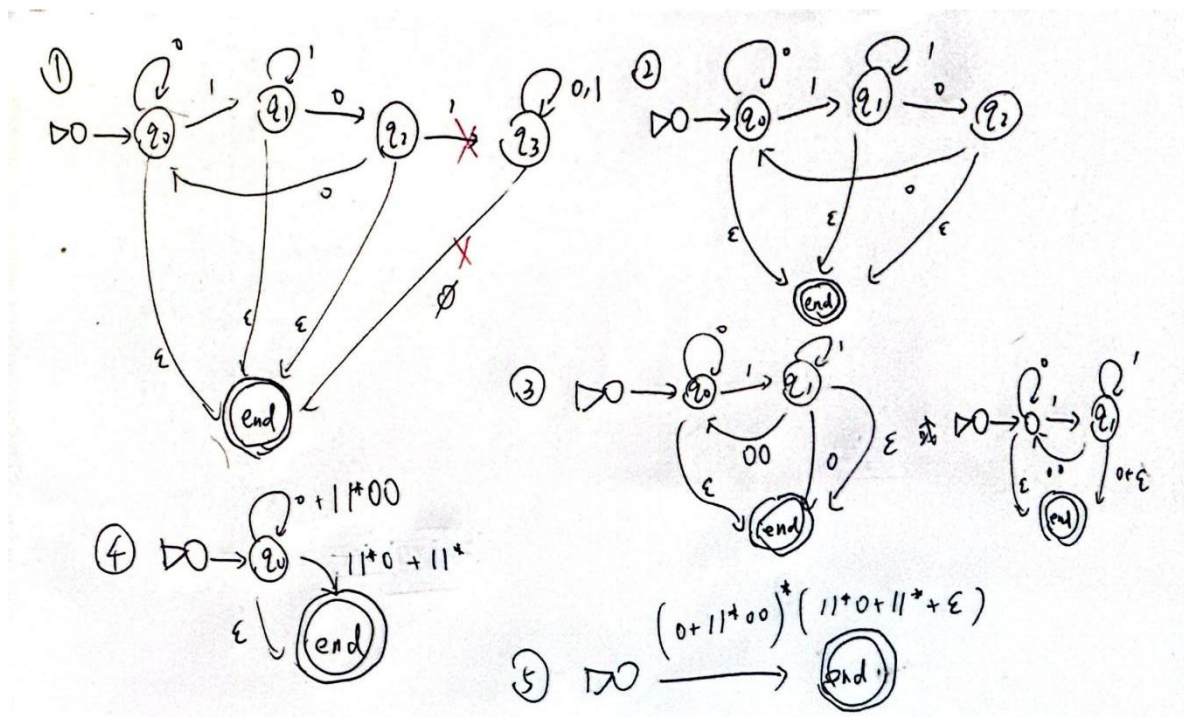
- 不包含 101 作为子串的所有 0 和 1 的串的集合。

采用先画出 DFA 再进行转化的方法: 画出对应的 DFA

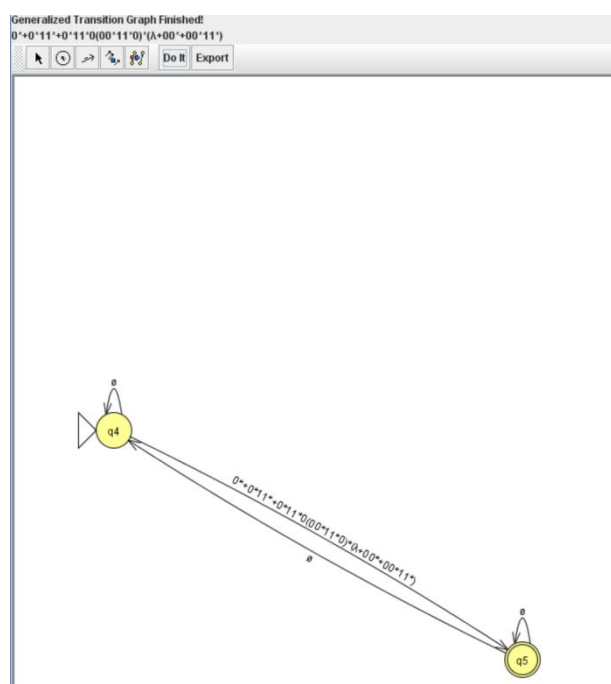


Input	
101	Reject
01010110	Reject
10010	Accept
000010001	Accept
111010	Reject
10010000101	Reject

其中  $q3$  到  $q6$  的是大非, 也就是不可能到达最后的接受态。按照上课讲的第一种方法也就是 GNFA 方法就可以转化出来最终的正则表达式, 转化的过程如下手写所示 (省去  $q3$  因为没有意义): 因此最终的等价的正则表达式为  $(0+11^*00)^* (11^*0+11^*+\epsilon)$



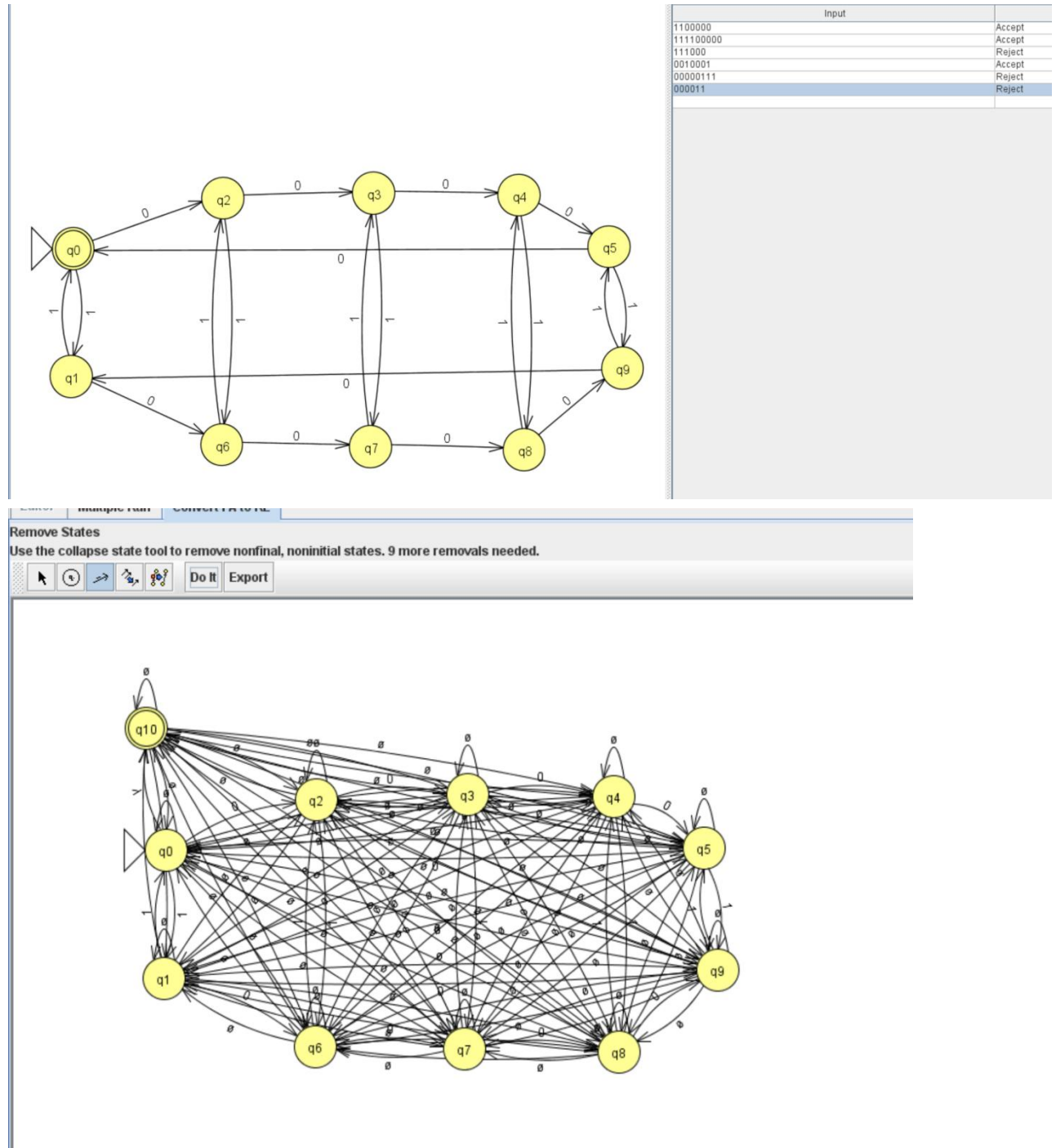
利用 JFLAP 可以给出他转化的正则表达式如下： $0^*0^*11^*0^*11^*0(00^*11^*0)^*(\lambda + 00^*+00^*11^*)$  与我们的表达式其实是等价的。（手写转化之前没有发现这个功能）



b) 具有相同个数的 0 和 1，使得在任何前缀中，0 的个数不比 1 的个数多 2，1 的个数也不比 0 的个数多 2，所有这种 0 和 1 的串的集合。

因为 01 个数相同因此最先先到的就是 10 或者 01 组成的字符串，并且这俩正好满足任何前缀中 0 的个数不比 1 多 2，1 的个数不比 0 多 2，举出例子，比如 1010、0101、1001、0110 等因此直接写就行：  
 $(01+10)^*$

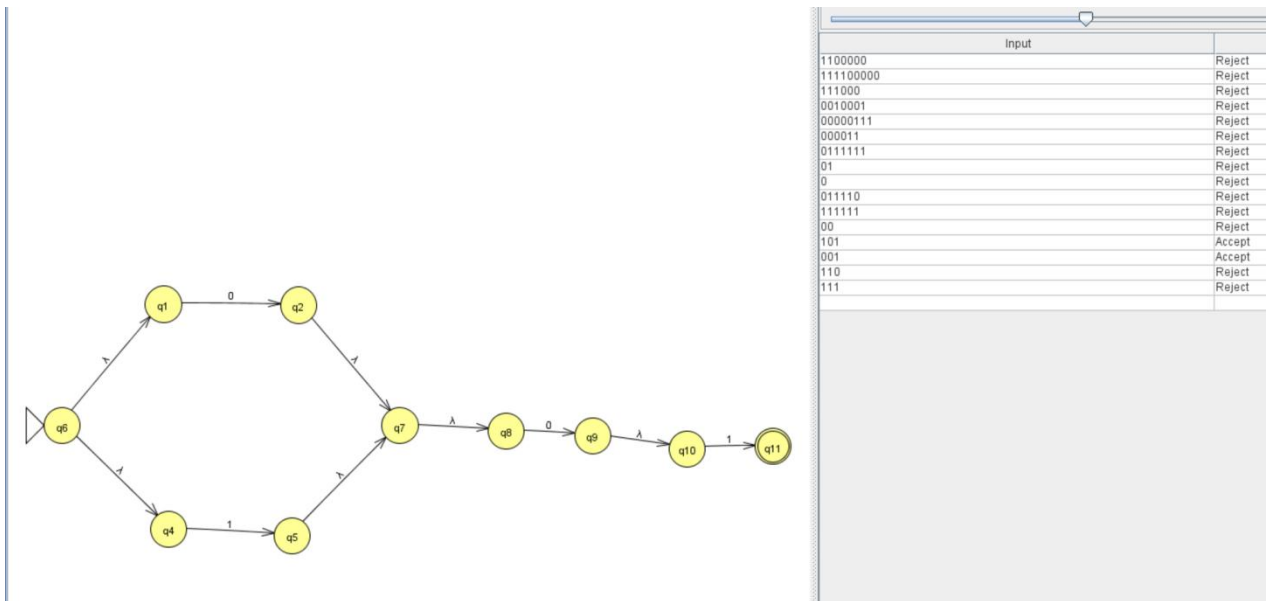
b) 0 的个数被 5 整除且 1 的个数是偶数的所有 0 和 1 的串的集合。



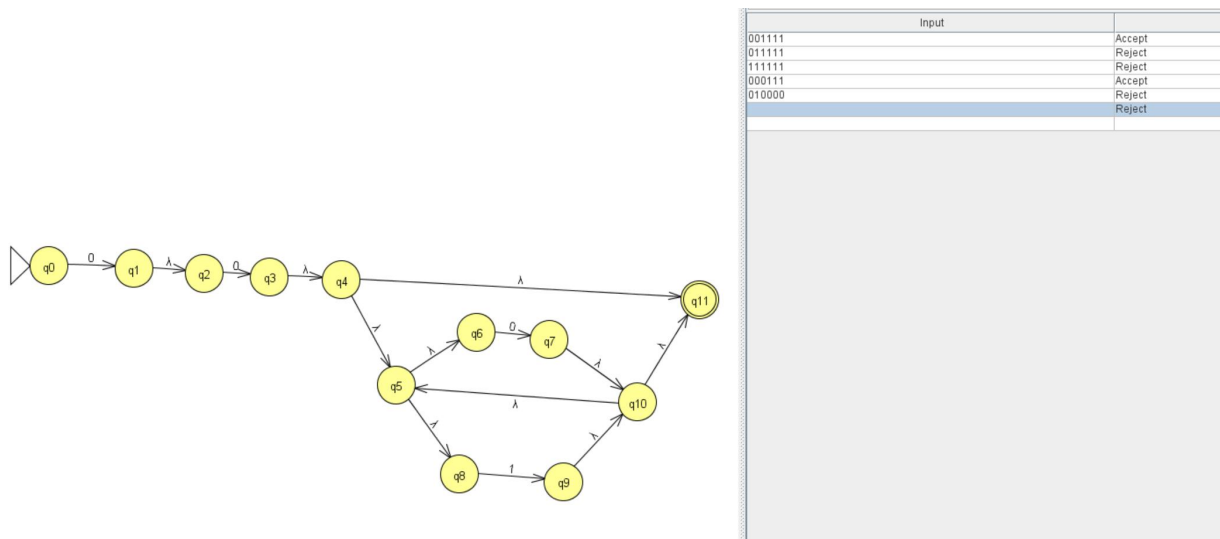
通过 JFLAP 自带的转化为 RE 的工具，转化出来如下的正则表达式：

$$(11+00000+(10+01)(11)^*10000+(001+(10+01)(11)^*(0+101))(11)^*1000+(0001+(10+01)(11)^*1001+(001+(10+01)(11)^*(0+101))(11)^*(0+101))(11)^*(0+101))(11)^*100+(00001+(10+01)(11)^*10001+(001+(10+01)(11)^*(0+101))(11)^*1001+(0001+(10+01)(11)^*1001+(001+(10+01)(11)^*(0+101))(11)^*(0+101))(11)^*(0+101))(11)^*00(11)^*10001+00(11)^*(0+101)(11)^*1001+(00(11)^*1001+00(11)^*(0+101)(11)^*(0+101))(11)^*(0+101))* (01+10+00(11)^*10000+00(11)^*(0+101)(11)^*1000+(00(11)^*1001+00(11)^*(0+101)(11)^*(0+101))(11)^*100))^*$$





c)  $00(0 + 1)^*$



**习题 3.2.8** 给出一个算法：输入一个 DFA  $A$ ，对于给定的  $n$ （与  $A$  的状态个数无关），计算出  $AA$  所接受的长度为  $nm$  的串的个数。这个算法应当对于  $n$  和  $A$  的状态数来说都是多项式的。提示：使用定理 3.4 的构造所提示的技术。





在 <https://regex101.com/>测试结果如下：可见给出的样例均可以识别匹配：

```
:/ ^((?i)(?:[$€£¥]|USD|EUR|GBP|CNY)?\s*(?:((?:\d{1,3}(?:,\d{3})+|\d+)(?:\.\d+)?\s*[KMB]?)\s*(?:\s*(?:to|-|-)\s*(?:[$€£¥]|USD|EUR|GBP|CNY)?\s*(?:((?:\d{1,3}(?:,\d{3})+|\d+)(?:\.\d+)?\s*[KMB]?)?)\s*(?:\/|bper\b|\ba(?:n)?\b)?\s*(?:hour(?:ly)?|hr|h|day|d|week|wk|month|mo|mth|year|yr|annum|annual|daily|weekly|monthly|yearly))?$ /gm
```

TEST STRING

\$20-\$30/hr

CNY\*12500\*per\*month

\$50,000\*per\*year

\$25/hour

€12.5/hr

80k\*a\*year

£400\*per\*week

\$20-\$30/hr

\$1.2m\*annual

EUR\*30-35\*per\*hour

2000\*daily

45k\*monthly

Salary:\*\$100,000

75k

\$20-\$30

\$40\*per

12-to-18\*students

20-30%

+1-202-555-0100

€,500\*per\*year

CNY\*12,,000\*per\*month

12\*to\*18\*students

20-30%

11:00