L0.2

1

1. 这个集合包含了所有的正奇数
2. 这个集合包含了所有的偶数
3. 这个集合包含了 $\mathbb{N}$ 中所有 6 的倍数
4. 这个集合包含了所有 01 回文字符串
5. 这个集合是一个空集

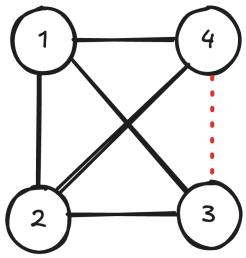
2

1. $\{1, 10, 100\}$
2. $\{n \in \mathbb{N} \mid n > 5\}$
3. $\{0, 1, 2, 3, 4\}$
4. $\{\text{aba}\}$

3

1. A 不是 B 的子集, 但是 B 是 A 的子集.
2. $A \cup B = \{x, y, z\}$, $A \cap B = \{x, y\}$, $A \times B = \{(x, x), (x, y), (y, x), (y, y), (z, x), (z, y)\}$, $\mathcal{P}(B) = \{\emptyset, \{x\}, \{y\}, \{x, y\}\}$

4



每一个顶点的度数为 **1**: 3; **2**: 3; **3**: 2; **4**: 2.

L0.3 - L1.2

1

假设一个含有 N 的顶点的图的每一个顶点的度数都不同, 而一个顶点的度数至少为 0, 最多为 $N - 1$, 所以这 N 个顶点的度数只能是

$0, 1, 2, ..., N-1$ 这 N 个数. 但是如果有一个顶点的度数为 $N-1$, 那么就不可能有一个顶点的度数为 0, 因为这个度数为 $N-1$ 的顶点已经和图中的所有其他顶点相连了. 所以假设不成立, 至少有两个顶点的度数是相同的.

2

如果素数只有有限个, 那么一定有最大的一个素数, 记为 p. 我们考虑所有小于等于 p 的素数 $p_1, p_2, ..., p_k$, 那么 $N = p_1 \times p_2 \times ... \times p_k + 1$ 一定也是素数, 因为 N 不能被任何一个素数 $p_1, p_2, ..., p_k$ 整除. 但是 $N > p$, 这和 p 是最大的素数矛盾, 所以素数有无穷多个.

3

1. 设 $S(n) = \frac{1}{2}n(n+1)$, 当 $n = 1$ 时, $S(1) = 1$, 命题成立. 假设当 $n = k$ 时命题成立, 即 $S(k) = \frac{1}{2}k(k+1)$, 那么当 $n = k+1$ 时,

   $$S(k+1) = S(k) + (k+1) = \frac{1}{2}k(k+1) + (k+1) = \frac{1}{2}(k+1)(k+2)$$

   所以当 $n = k+1$ 时命题也成立. 根据数学归纳法, 对任意的 $n \in \mathbb{N}^+$, 都有 $S(n) = 1 + 2 + ... + n$.

   再设 $C(n) = \frac{1}{4}n^2(n+1)^2$, 当 $n = 1$ 时, $C(1) = 1$, 命题成立. 假设当 $n = k$ 时命题成立, 即 $C(k) = \frac{1}{4}k^2(k+1)^2$, 那么当 $n = k+1$ 时,

   $$\begin{aligned} C(k+1) &= \frac{1}{4}(k+1)^2(k+2)^2 \\ &= \frac{1}{4}(k+1)^2\left(k^2 + 4k + 4\right) \\ &= \frac{1}{4}(k+1)^2k^2 + (k+1)^2k + (k+1)^2 \\ &= C(k) + (k+1)^3 \end{aligned}$$

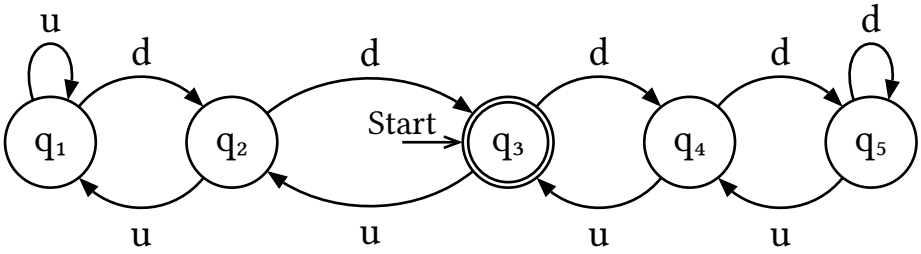
   所以当 $n = k+1$ 时命题也成立. 根据数学归纳法, 对任意的 $n \in \mathbb{N}^+$, 都有 $C(n) = 1^3 + 2^3 + ... + n^3$.

4

1. M_1 的起始状态为 q_1, 接受状态集为 $\{q_2\}$; M_2 的起始状态为 q_1, 接受状态集为 $\{q_1, q_4\}$.

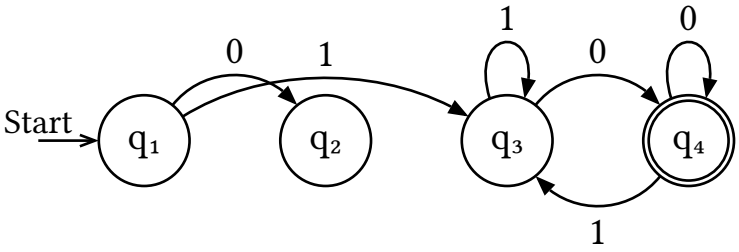
2. M_1 的状态序列为 $q_1q_2q_3q_1q_1$; M_2 的状态序列为 $q_1q_1q_1q_2q_4$.
3. M_1 不接受, M_2 接受.
4. M_1 不接受, M_2 接受.

5

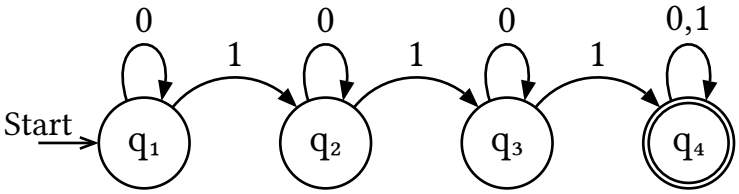


6

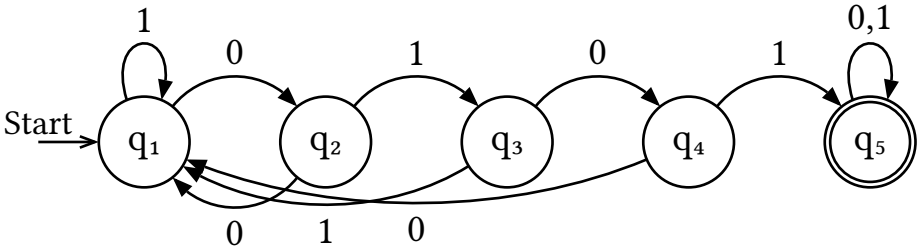
1. w 从 1 开始且以 0 结束



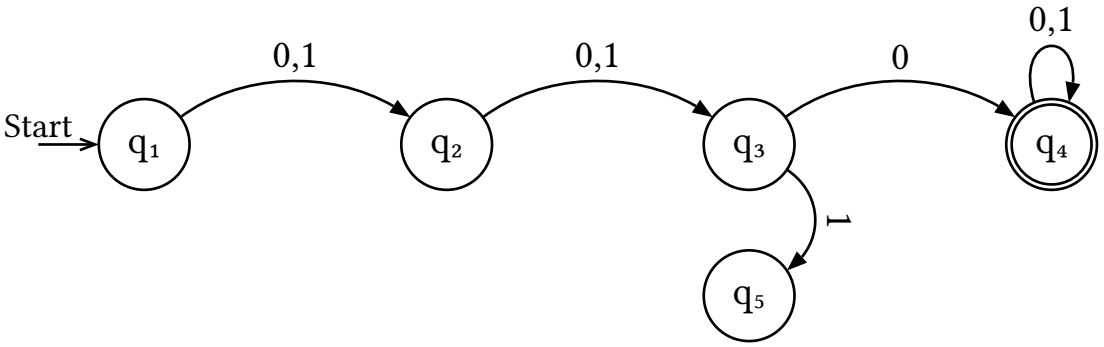
2. w 含有至少 3 个 1



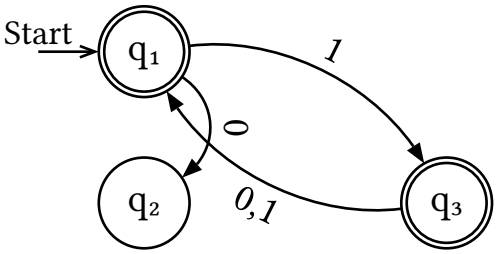
3. w 含有子串 0101



4. w 的长度不小于 3 且第 3 个符号为 0



5. w 的奇数位置均为 1

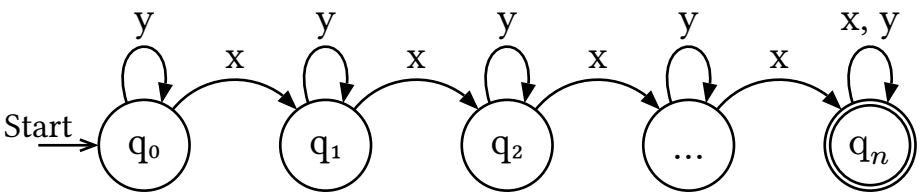


5

下面每个语言都是两个简单语言的交. 构造识别这些语言的 DFA:

1. w 至少含有 3 个 a 和至少 2 个 b

先构建一个 DFA M_{count} 来识别含有至少 n 个某个符号的字符串, 假设这个符号为 x, y 代表字母表中除 x 以外的其他符号, 那么 M_{count} 的状态转移图为:



然后构建两个 DFA, M_a 识别含有至少 3 个 a 的字符串, M_b 识别含有至少 2 个 b 的字符串. 最后通过笛卡尔积构造法构造出识别这两个语言交的 DFA M, 具体而言:

- $Q = Q_a \times Q_b$
- $q_0 = (q_{0a}, q_{0b})$
- $F = \{(q_a, q_b) \mid q_a \in F_a, q_b \in F_b\}$
- $\delta((q_a, q_b), x) = \left(\delta_{a(q_a,x)}, \delta_{b(q_b,x)}\right)$

2. w 含有偶数个 a 和 1 个或 2 个 b

同样先构建一个 DFA M_{mod} 来识别含有 $n \pmod 2$ 个某个符号的字符串, 假设这个符号为 x, y 代表字母表中除 x 以外的其他符号, 那么 M_{mod} 的状态转移图为:

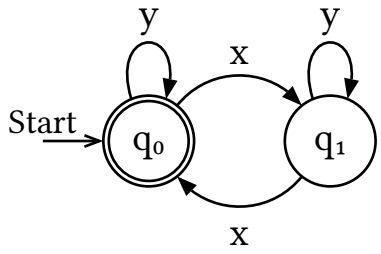


Figure 1: $n = 0 \pmod 2$

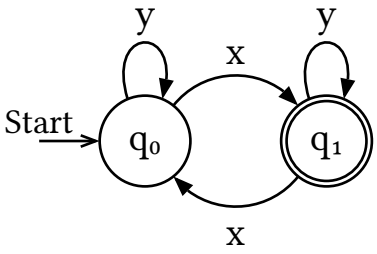


Figure 2: $n = 1 \pmod 2$

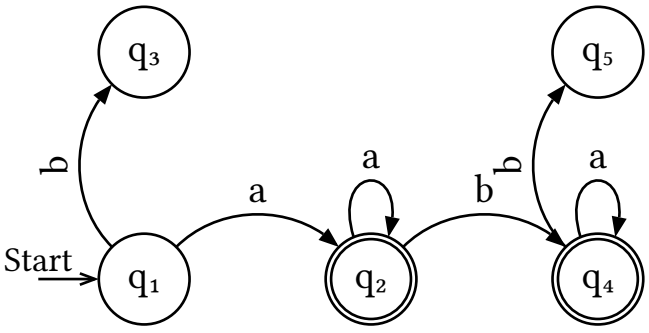
假设我们已经基于 M_{count} 构造了 M_{b1} 和 M_{b2} 来分别识别含有 1 个和 2 个某个符号的字符串, 那么我们可以通过并集构造法来构造出识别含有 1 个或 2 个某个符号的字符串的 DFA M_{b1or2}, 具体而言:

- $Q = Q_{\text{b1}} \times Q_{\text{b2}}$
- $q_0 = (q_{0b1}, q_{0b2})$
- $F = \{(q_{b1}, q_{b2}) \mid q_{b1} \in F_{b1} \vee q_{b2} \in F_{b2}\}$
- $\delta((q_{b1}, q_{b2}), x) = (\delta_{b1}(q_{b1}, x), \delta_{b2}(q_{b2}, x))$

然后我们可以通过 **1.** 中的方式构造 M_{mod} 和 M_{b1or2} 的笛卡尔积来识别这两个语言交的 DFA.

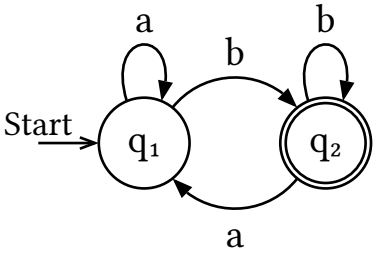
3. w 从 a 开始且最多有 1 个 b

直接构造



4. w 含有奇数个 a 并且以 b 结尾

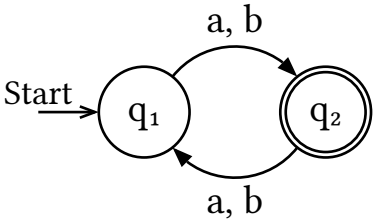
含有奇数个 a 的 DFA 可以通过 **2.** 中的 M_{mod} 构造, 以 b 结尾的 DFA 可以直接构造:



然后我们可以通过 **1.** 中的方式构造这两个语言交的 DFA.

5. w 的长度为偶数且含有奇数个 a

含有奇数个 a 的 DFA 可以通过 **2.** 中的 M_{mod} 构造, 长度为偶数的 DFA 可以直接构造:



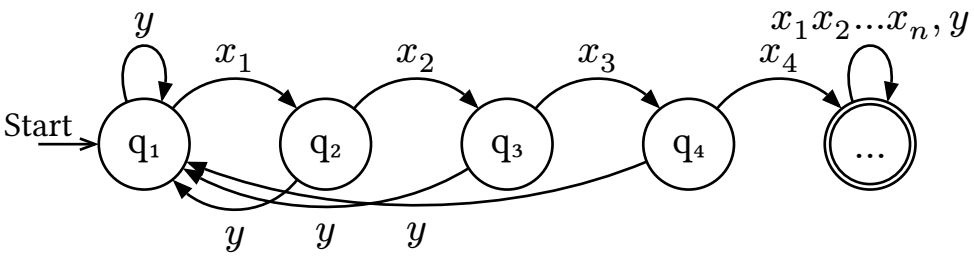
然后我们可以通过 **1.** 中的方式构造这两个语言交的 DFA.

6

下述每个语言都是一个简单语言的补. 构造识别这些语言的 DFA:

1. w 不含有 ab 子串

先构造识别含有任意子串 $x_1x_2...x_n$ 的 DFA M_{sub}



然后我们只需要令新的 DFA 的接受状态集为原 DFA 的非接受状态集, 即 $F = Q - F_{\text{sub}}$, 就可以识别不含有 ab 子串的语言.

2. w 既不含有 ab 子串, 也不含有 ba 子串

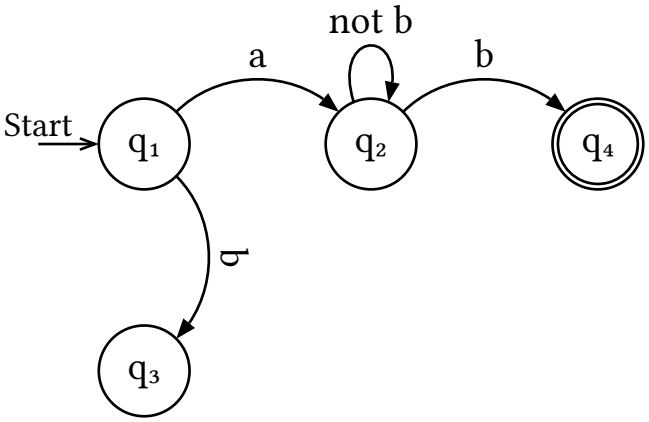
先通过 M_{sub} 构造识别含有 ab 子串的 DFA M_{ab}, 再通过 M_{sub} 构造识别含有 ba 子串的 DFA M_{ba}. 然后通过并集构造法构造出识别这两个语言并的 DFA M_{aborba}, 具体而言:

- $Q = Q_{\text{ab}} \times Q_{\text{ba}}$
- $q_0 = (q_{0\ \text{ab}}, q_{0\ \text{ba}})$
- $F = \{(q_{\text{ab}}, q_{\text{ba}}) \mid q_{\text{ab}} \in F_{\text{ab}} \vee q_{\text{ba}} \in F_{\text{ba}}\}$
- $\delta((q_{\text{ab}}, q_{\text{ba}}), x) = (\delta_{\text{ab}}(q_{\text{ab}}, x), \delta_{\text{ba}}(q_{\text{ba}}, x))$

再利用 **1.** 中的方法构造出识别这两个语言的并的补的 DFA.

3. w 不是在 a^*b^* 中的字符串

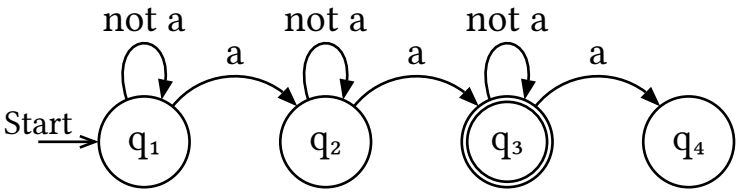
先构造识别在 a^*b^* 中的字符串的 DFA:



然后再利用 **1.** 中的方法构造出识别这个语言的补的 DFA.

4. w 是恰好不含 2 个 a 的任意串

我们先构造一个识别恰好含有 2 个 a 的任意串的 DFA:



然后再利用 **1.** 中的方法构造出识别这个语言的补的 DFA.