

Describe deterministic finite-state automata that accept each of the following languages over the alphabet $\Sigma = \{0, 1\}$. You may find it easier to describe these DFAs formally than to draw pictures.

Either drawings or formal descriptions are acceptable, as long as the states Q , the start state s , the accept states A , and the transition function δ are all clear. Try to keep the number of states small.

1. All strings in which the number of 0s is even **and** the number of 1s is *not* divisible by 3.

Solution: We use a standard product construction of two DFAs, one accepting strings with an even number of 0s, and the other accepting strings where the number of 1s is not a multiple of 3.

The product DFA has six states, each labeled with a pair of integers, one indicating the number of 0s read modulo 2, the other indicating the number of 1s read modulo 3.

$$Q := \{0, 1\} \times \{0, 1, 2\}$$

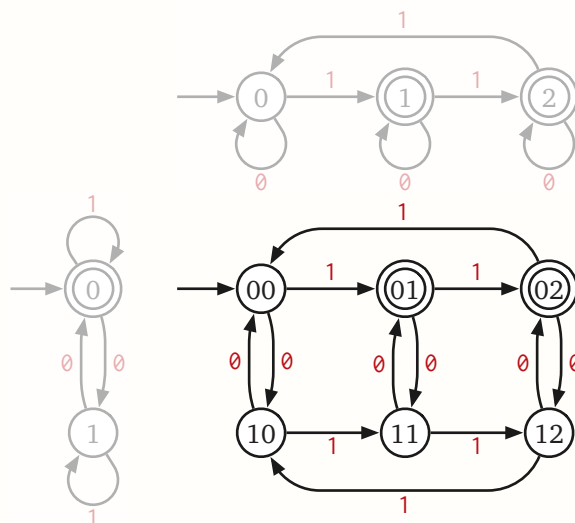
$$s := (0, 0)$$

$$A := \{(0, 1), (0, 2)\}$$

$$\delta((q, r), 0) := (q + 1 \bmod 2, r)$$

$$\delta((q, r), 1) := (q, r + 1 \bmod 3)$$

In this case, the product DFA is simple enough that we can just draw it out in full. I've drawn the two factor DFAs (in gray) to the left and above for reference.



■

2. All strings in which the number of 0s is even **or** the number of 1s is *not* divisible by 3.

Solution: We use a standard product construction of two DFAs, one accepting strings with an even number of 0s, and the other accepting strings where the number of 1s is not a multiple of 3.

The product DFA has six states, each labeled with a pair of integers, one indicating the number 0s read modulo 2, the other indicating the number of 1s read modulo 3.

$$Q := \{0, 1\} \times \{0, 1, 2\}$$

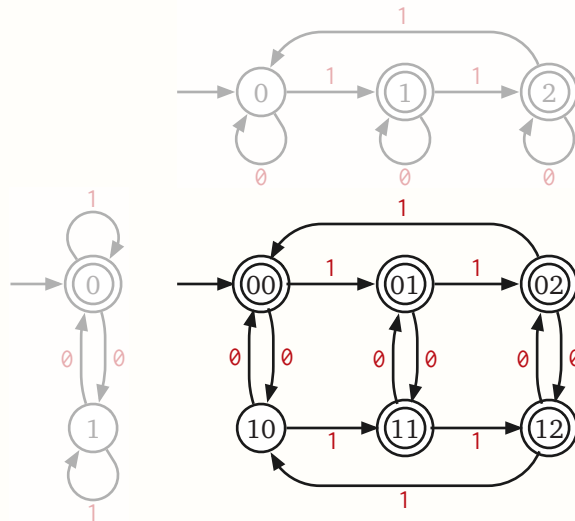
$$s := (0, 0)$$

$$A := \{(0, 0), (0, 1), (0, 2), (1, 1), (1, 2)\}$$

$$\delta((q, r), 0) := (q + 1 \bmod 2, r)$$

$$\delta((q, r), 1) := (q, r + 1 \bmod 3)$$

In this case, the product DFA is simple enough that we can just draw it out in full. I've drawn the two factor DFAs (in gray) to the left and above for reference.



Yes, this is exactly the same DFA as problem 1, except for the accepting states. Similar standard product constructions yield DFAs for any other boolean combination of these two factor languages, including the following:

- $\{w \mid \#(0, w) \text{ is even and } \#(1, w) \text{ is divisible by } 3\}$
- $\{w \mid \#(0, w) \text{ is odd or } \#(1, w) \text{ is divisible by } 3\}$
- $\{w \mid \text{if } \#(0, w) \text{ is even, then } \#(1, w) \text{ is not divisible by } 3\}$
- $\{w \mid \text{if } \#(1, w) \text{ is not divisible by } 3, \text{ then } \#(0, w) \text{ is even}\}$
- $\{w \mid \#(0, w) \text{ is even if and only if } \#(1, w) \text{ is not divisible by } 3\}$
- $\{w \mid \#(0, w) \text{ is odd if and only if } \#(1, w) \text{ is divisible by } 3\}$

This is why, whenever you describe a product construction, especially for a homework or exam problem, you **must** specify the set of accepting states. ■

3. All strings that are **both** the binary representation of an integer divisible by 3 **and** the ternary (base-3) representation of an integer divisible by 4.

For example, the string **1100** is an element of this language, because it represents $2^3 + 2^2 = 12$ in binary and $3^3 + 3^2 = 36$ in ternary.

Solution: Again, we use a standard product construction of two DFAs, one accepting binary strings divisible by 3, the other accepting ternary strings divisible by 4. The product DFA has twelve states, each labeled with a pair of integers: The binary value read so far modulo 3, and the ternary value read so far modulo 4.

$$Q := \{0, 1, 2\} \times \{0, 1, 2, 3\}$$

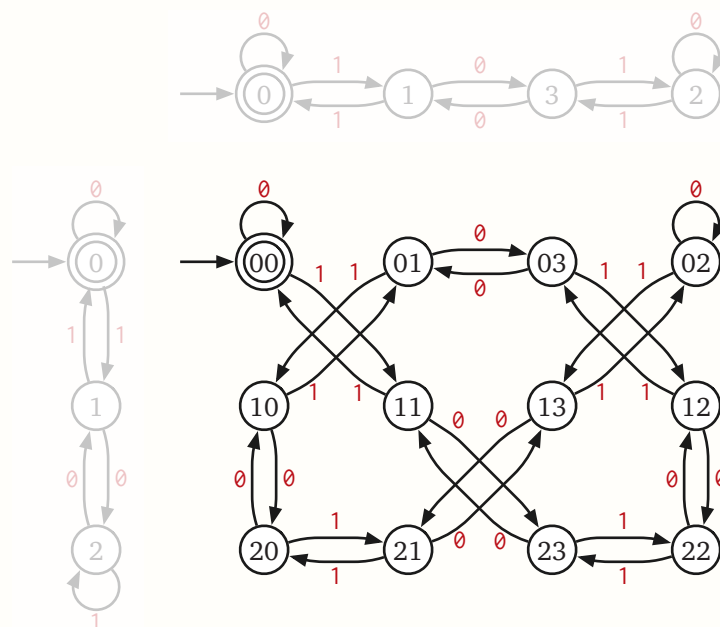
$$s := (0, 0)$$

$$A := \{(0, 0)\}$$

$$\delta((q, r), 0) := (2q \bmod 3, 3r \bmod 4)$$

$$\delta((q, r), 1) := (2q + 1 \bmod 3, 3r + 1 \bmod 4)$$

For reference, here is a drawing of the DFA, with the two factor DFAs (in gray) to the left and above. We would not expect you to draw this, especially on exams. Or more accurately: We would expect you *not* to draw this, *especially* on exams. The states of the factor DFA that maintains ternary-value-mod-4 are deliberately “out of order” to simplify the drawing.



Harder problems to think about later:

4. All strings in which the subsequence 0101 appears an even number of times.

Solution: Our DFA has 16 states, each labeled with a vector (a, b, c, d) of four bits:

- a is the number of times (mod 2) that we have seen 0 .
- b is the number of times (mod 2) we have seen the subsequence 01 .
- c is the number of times (mod 2) we have seen the subsequence 010 .
- d is the number of times (mod 2) we have seen the subsequence 0101 .

$$Q := \{0, 1\} \times \{0, 1\} \times \{0, 1\} \times \{0, 1\}$$

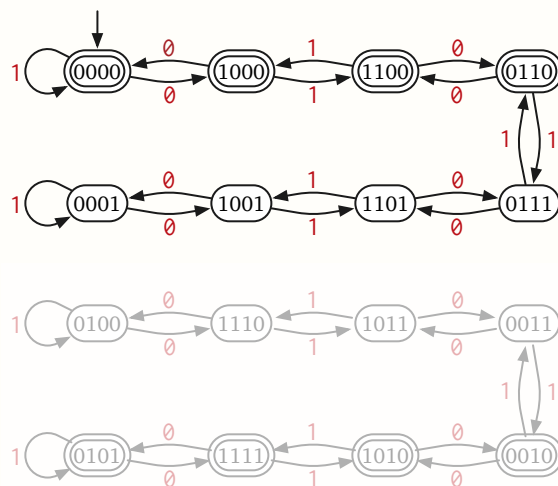
$$s := (0, 0, 0, 0)$$

$$A := \{(a, b, c, 0) \mid a, b, c \in \{0, 1\}\}$$

$$\delta((a, b, c, d), 0) := (a + 1 \bmod 2, b, c + b \bmod 2, d)$$

$$\delta((a, b, c, d), 1) := (a, b + a \bmod 2, c, d + c \bmod 2)$$

This DFA is small enough that we can draw it out in full. It turns out that only eight of the sixteen states are reachable from the start state; the unreachable states are grayed out in the figure below. We would expect you *not* to remove these unreachable states, especially on exams.



- *5. All strings w such that $\binom{|w|}{2} \bmod 6 = 4$. [Hint: Maintain both $\binom{|w|}{2} \bmod 6$ and $|w| \bmod 6$.]

Solution: Our DFA has 36 states, each labeled with a pair of integers representing $\binom{|x|}{2} \bmod 6$ and $|x| \bmod 6$, where x is the prefix of the input read so far.

$$Q := \{0, 1, 2, 3, 4, 5\} \times \{0, 1, 2, 3, 4, 5\}$$

$$s := (0, 0)$$

$$A := \{(4, r) \mid r \in \{0, 1, 2, 3, 4, 5\}\}$$

$$\delta((q, r), 0) := (q + r \bmod 6, r + 1 \bmod 6)$$

$$\delta((q, r), 1) := (q + r \bmod 6, r + 1 \bmod 6)$$

The transition function exploits the identity $\binom{n}{2} = \binom{n-1}{2} + n - 1$. ■

Solution: This language is the set of all strings w such that $|w| \bmod 12 \in \{5, 8\}$. So this language is accepted using a 12-state DFA, whose states represent the number of symbols read so far, modulo 12.

$$Q := \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$

$$s := 0$$

$$A := \{5, 8\}$$

$$\delta(q, 0) := (q + 1) \bmod 12$$

$$\delta(q, 1) := (q + 1) \bmod 12$$

To prove this DFA is correct, we need to argue, for all non-negative integers n , that $\binom{n}{2} \bmod 6 = 4$ if and only if $n \bmod 12 = 5$ or $n \bmod 12 = 8$. The simplest way to prove this claim is to prove stronger claim: The function $n \mapsto \binom{n}{2} \bmod 6$ has period 12, and its value depends on $n \bmod 12$ according to the following table. Notice that the residue 4 appears twice in each period, at positions 5 and 8.

$n \bmod 12$	0	1	2	3	4	5	6	7	8	9	10	11
$\binom{n}{2} \bmod 6$	0	0	1	3	0	4	3	3	4	0	3	1

This stronger claim follows by induction from the recurrence $\binom{n}{2} = \binom{n-1}{2} + n - 1$. ■

- ★6. All strings w such that $F_{\#(10,w)} \bmod 10 = 4$, where $\#(10,w)$ denotes the number of times 10 appears as a substring of w , and F_n is the n th Fibonacci number.

Solution: Our DFA has 200 states(!), each labeled with three values:

- $F_k \bmod 10$, where k is the number of times we have seen the substring 10 .
- $F_{k+1} \bmod 10$, where k is the number of times we have seen the substring 10 .
- The last symbol read (or \emptyset if we have read nothing yet)

Here is the formal description:

$$Q := \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \times \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \times \{\emptyset, 1\}$$

$$s := (0, 1, \emptyset)$$

$$A := \{4\} \times \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \times \{\emptyset, 1\}$$

$$\delta((q, r, \emptyset), \emptyset) := (q, r, \emptyset) \quad \delta((q, r, 1), \emptyset) := (r, q + r \bmod 10, \emptyset)$$

$$\delta((q, r, \emptyset), 1) := (q, r, 1) \quad \delta((q, r, 1), 1) := (q, r, 1)$$

The transition function exploits the recursive definition $F_{k+1} = F_k + F_{k-1}$. ■

Solution: The Fibonacci numbers modulo 10 define a repeating sequence with period 60, and the residue 4 appears three times in each period:

0, 1, 1, 2, 3, 5, 8, 3, 1, 4, 5, 9, 4, 3, 7, 0, 7, 7, 4, 1, 5, 6, 1, 7, 8, 5, 3, 8, 1, 9,
0, 9, 9, 8, 7, 5, 2, 7, 9, 6, 5, 1, 6, 7, 3, 0, 3, 3, 6, 9, 5, 4, 9, 3, 2, 5, 7, 2, 9, 1, ...

(We can verify this pattern for all Fibonacci numbers using a straightforward but tedious induction proof.) It follows that our language can be accepted by the following DFA with “only” 120 states, each labeled with two values:

- The number of times we’ve seen the substring 10 , modulo 60
- The last symbol read (or \emptyset if we have read nothing yet)

Here is the formal description:

$$Q := \{0, 1, \dots, 59\} \times \{\emptyset, 1\}$$

$$s := (0, \emptyset)$$

$$A := \{9, 18, 51\} \times \{\emptyset, 1\}$$

$$\delta((r, \emptyset), \emptyset) := (r, \emptyset)$$

$$\delta((r, 1), \emptyset) := (r + 1 \bmod 60, \emptyset)$$

$$\delta((r, \emptyset), 1) := (r, 1)$$

$$\delta((r, 1), 1) := (r, 1)$$

■