Designing DFAs via product construction and designing NFAs.

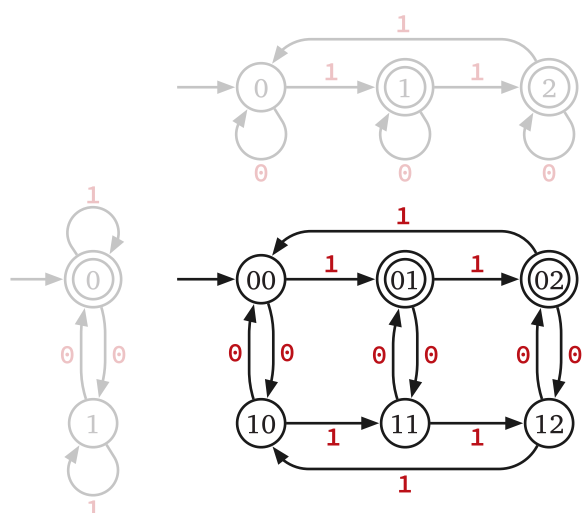
1 DFA for all strings in which the number of 0s is even and the number of 1s is *not* divisible by 3.

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 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 have drawn the two factor DFAs (in gray) to the left and above for reference.



2 DFA for 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:

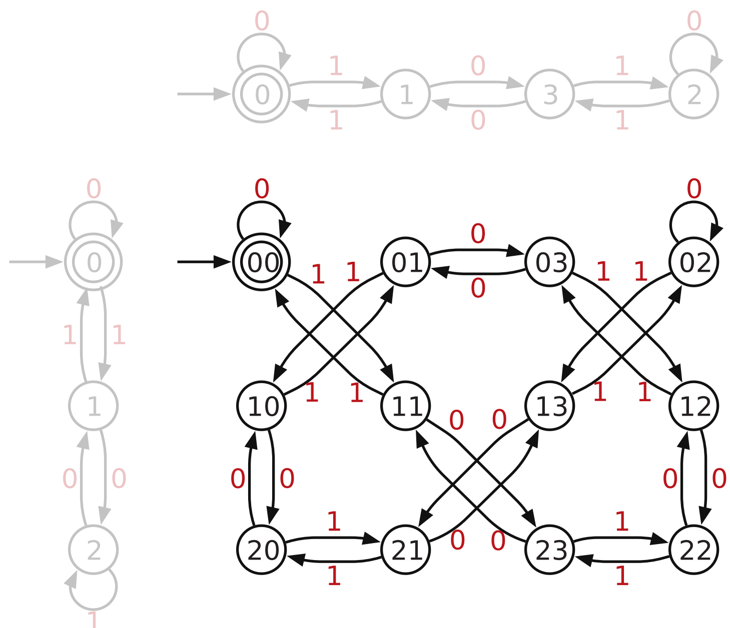
Again, we use a standard product construction of two DFAs, one accepting binary strings divisible by 3, the other accepting ternary strings divisible by 4. The product DFA has twelve states, each labeled with a pair of integers: The binary value read so far modulo 3, and the ternary value read so far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

4.

$$
\begin{aligned}
Q &:= \{0,1,2\} \times \{0,1,2,3\} \\
s &:= (0,0) \\
A &:= \{(0,0)\}
\end{aligned}
$$

$$
\begin{aligned}
\delta((q,r),0) &:= ((2q) \bmod 3, \quad\;\; (3r) \bmod 4) \\
\delta((q,r),1) &:= ((2q+1) \bmod 3, \; (3r+1) \bmod 4)
\end{aligned}
$$

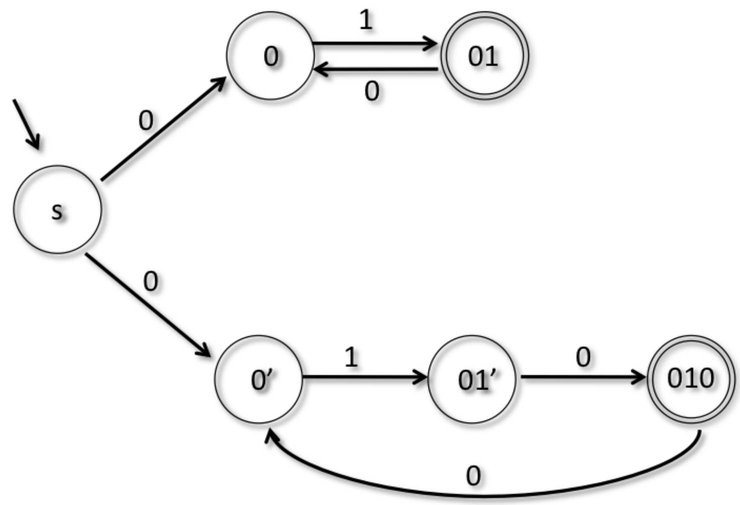
For reference, here is a drawing of the DFA, with the two factor DFAs (in gray) to the left and above; we would not expect you to draw this, especially on exams. More importantly we would expect you ***not*** to draw this, ***especially*** on exams. The states of the factor DFA that maintains ternary-value-mod-4 are deliberately "out of order" to simplify the drawing.



3 Design an NFA for the language $(01)^+ + (010)^+$.

Solution:

The NFA is shown in the figure below.



Note that we've separated the two cases of either repeated 01, or repeated 010. Why would the NFA with states labeled 0 and $0'$ merged be incorrect?

4 DFA for 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.

$$\begin{aligned} Q &:= \{0,1,2,3,4,5\} \times \{0,1,2,3,4,5\} \\ s &:= \{(0,0)\} \\ A &:= \{(4,r) \mid r \in \{0,1,2,3,4,5\}\} \end{aligned}$$

$$\begin{aligned} \delta((q,r),0) &:= (q + r \bmod 6,\ r + 1 \bmod 6) \\ \delta((q,r),1) &:= (q + r \bmod 6,\ r + 1 \bmod 6) \end{aligned}$$

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

Solution:

The language is identical to the set of strings w such that $|w| \bmod 12 \in \{5, 8\}$. This language can be accepted using a 12-state DFA.

5 (**Hard.**) 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 nth Fibonacci number:

$$F_n = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F_{n-1} + F_{n-2} & \text{otherwise} \end{cases}$$

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 0 if we have read nothing yet)

Here is the formal description:

$$
\begin{aligned}
Q &:= \{0,1,2,3,4,5,6,7,8,9\} \times \{0,1,2,3,4,5,6,7,8,9\} \times \{0,1\} \\
s &:= \{(0,1,0)\} \\
A &:= \{(4,r,a) \mid r \in \{0,1,2,3,4,5,6,7,8,9\} \text{ and } a \in \{0,1\}\}
\end{aligned}
$$

$$
\begin{aligned}
\delta((q,r,0),0) &:= (q,r,0) \\
\delta((q,r,1),0) &:= (r,(q+r) \bmod 10, 0) \\
\delta((q,r,0),1) &:= (q,r,1) \\
\delta((q,r,1),1) &:= (q,r,1)
\end{aligned}
$$

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. So this language can be accepted by a DFA with "only" 120 states.

Extra problems [no solutions would be provided]

6 Let $L = \{w \in \{a,b\}^* \mid \text{an } a \text{ appears in some position } i \text{ of } w, \text{ and a } b \text{ appears in position } i+2\}$.

6.A. Create an NFA N for L with at most four states.

6.B. Using the "power-set" construction, create a DFA M from N. Rather than writing down the sixteen states and trying to fill in the transitions, build the states as needed, because you won't end up with unreachable or otherwise superfluous states.

6.C. Now directly design a DFA M' for L with only five states, and explain the relationship between M and M'.