

Designing DFAs via product construction and designing NFAs.

- 1** Describe a DFA that accepts the following language over the alphabet  $\Sigma = \{0, 1\}$ .  
All strings in which the number of **0**s is even and the number of **1**s is *not* divisible by 3.
- 2** 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.
- 3** Design an NFA for the language  $(01)^+ + (010)^+$ .

**Work on these later:**

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.

- 4** All strings  $w$  such that  $\binom{|w|}{2} \bmod 6 = 4$ . (**Hint:** Maintain both  $\binom{|w|}{2} \bmod 6$  and  $|w| \bmod 6$ .)
- 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  $n$ th Fibonacci number:

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