1 DFAs

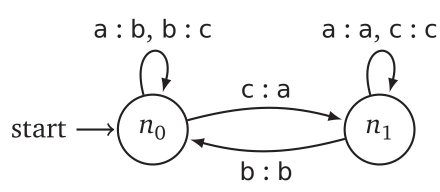
Describe deterministic finite-state automata that accept each of the following languages over the alphabet $\Sigma = \{\texttt{0}, \texttt{1}\}$. Describe briefly what each state in your DFAs *means*.

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 containing the substring **000**.
2. All strings *not* containing the substring **000**.
3. Every string except **000**. *[Hint: Don't try to be clever.]*
4. All strings in which the number of **0**s is even **and** the number of **1**s is *not* divisible by 3.
5. All strings in which the number of **0**s is even **or** the number of **1**s is *not* divisible by 3.
6. Given DFAs M_1 and M_2, all strings in $\overline{L(M_1)} \oplus L(M_2)$.
   Recall that for two sets A and B, their symmetric distance $A \oplus B$ is the set of elements in either A or B, but not both.

2 Other types of automata

1. A *finite-state transducer* (FST) is a type of deterministic finite automaton whose output is a string instead of just *accept* or *reject*. The following is the state diagram of finite state transducer FST_0.



Each transition of an FST is labeled at least an input symbol and an output symbol, separated by a colon (:). There can also be multiple input-output pairs for each transitions, separated by a comma (,). For instance, the transition from n_0 to itself can either take a or b as an input, and outputs b or c respectively.

When an FST computes on an input string $s := \overline{s_0 s_1 \ldots s_{n-1}}$ of length n, it takes the input symbols $s_0, s_1, \ldots, s_{n-1}$ one by one, starting from the starting state, and produces corresponding output symbols. For instance, the input string abccba produces the output string bcacbb, while cbaabc produces abbbca.

   (a) Each of the following strings is the input of FST_0. Give the sequence of states entered and the output produced.
      - aaca
      - cbbc
      - bcba

- acbbca

(b) Describe a formal model of FST. Specifically, describe the 5-tuple that defines an FST. *Hint: An* FST *has no accepting states, but it has the output alphabet* Γ. *Its transition function is of the form* $\delta : Q \times \Sigma \to Q \times \Gamma$.

(c) Give a formal description of FST_0.

(d) Give a state diagram of an FST with the following behavior. Its input and output alphabets are $\{\mathsf{T}, \mathsf{F}\}$. Its output string is inverted on the positions with indices divisible by 3 and is identical on all the other positions. For instance, on an input TFTTFTFT it should output FFTFFTTT.

Work on these later:

Describe deterministic finite-state automata that accept each of the following languages over the alphabet $\Sigma = \{\mathtt{0}, \mathtt{1}\}$. Describe briefly what each state in your DFAs *means*.

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.

7. All strings w such that *in every prefix of* w, the number of **0**s and **1**s differ by at most 1.

8. All strings containing at least two **0**s and at least one **1**.

9. All strings w such that *in every prefix of* w, the number of **0**s and **1**s differ by at most 2.

*10. All strings in which the substring **000** appears an even number of times.
(For example, **0001000** and **0000** are in this language, but **00000** is not.)

11. 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.

*12. All strings w such that $F_{\#(\mathtt{10}, w)} \bmod 10 = 4$, where $\#(\mathtt{10}, w)$ denotes the number of times **10** appears as a substring of w, and F_n is the nth Fibonacci number:

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