Unless explicitly asked to, you need not prove that any machine you construct defines the language you say it does. * denotes a harder problem.

1. Construct \( \epsilon \)-NFA’s and regular expressions for the following regular languages:

   (a) All words \( w \in \{0,1\}^* \) consisting of either the string 01 repeated some number of times (possibly none), or the string 010 repeated some number of times (possibly none).

   (b) All words \( w \in \{a,b,c\}^* \) consisting of some number of \( a \)'s (possibly none), followed by some number of \( b \)'s (at least one), followed by some number of \( c \)'s (possibly none).

   (c) All words \( w \in \{0,1\}^* \) which contain a 1 somewhere in the last 4 positions. If \( |w| < 4 \), then \( w \) must contain a 1 somewhere.

   (d) All words \( w \in \{a,\ldots,z,0,\ldots,9\}^* \) of the form \( \text{name:}\alpha.\text{address:}\beta \). Where might you use such a machine/expression, and why?

2. Give an \( \epsilon \)-NFA with the same language as that defined by each of the following regular expressions:

   \( (0 + 1)(01) \) \hspace{1cm} \( (a + bb)(ba^* + \epsilon) \) \hspace{1cm} \( (aa^*)b^* + c \)

3. Prove that \( \{w \in \{0,1\}^* \mid w \text{ contains no more than 5 consecutive 0's} \} \) is regular.

4. Let \( R, S, T \) be regular expressions. For each of the following statements, either prove that it is true, or find a specific counterexample.

   \( \mathcal{L}(R(S + T)) = \mathcal{L}(RS) \cup \mathcal{L}(RT) \)

   \( \mathcal{L}((R^*)^*) = \mathcal{L}(R^*) \)

   \( \mathcal{L}((RS)^*) = \mathcal{L}(R^*S^*) \)

   \( \mathcal{L}((R + S)^*) = \mathcal{L}(R^*) \cup \mathcal{L}(S^*) \)

   \( \mathcal{L}((R^*S^*)^*) = \mathcal{L}((R + S)^*) \)

5. Use the pumping lemma to show that none of the following languages are regular:

   \( \{a^n b^n \mid n \geq 0 \} \)

   \( \{a^n b^{2n} \mid n \geq 0 \} \)

   \( \{ww \mid w \in \{0,1\}^* \} \)

\( Date: \) November 10, 2021.
(6) For each of the following languages, determine whether or not they are regular. Justify your answers.

(a) \( \{ xcx \mid x \in \{a, b\}^* \} \)
(b) \( \{ xcy \mid x, y \in \{a, b\}^* \} \)
(c) \( \{ a^n b^m \mid n > m \} \)
(d) \( \{ a^n b^m \mid n \geq m \text{ and } m \leq 1000 \} \)
(e) \( \{ a^n b^m \mid n \geq m \text{ and } m \geq 1000 \} \)
(f) \( \{ 1^p \mid p \text{ is a prime} \} \)

(7) Let \( L, M \) be languages over \( \Sigma \). We define the difference \( L - M \) to be the words that are in \( L \) but not \( M \). That is, \( L - M := (L \cup M) \setminus M \). Show that if \( L, M \) are both regular languages, then \( L - M \) is a regular language over \( \Sigma \).

Is \( \{ a^n b^m \mid n \neq m \} \) a regular language?

(8) Prove that no infinite subset of \( \{0^n 1^n \mid n \geq 0 \} \) is a regular language.

(9) Find minimal DFA’s for each of the following languages. In each case, prove that your DFA is minimal.

(a) \( \{ a^n \mid n \geq 0, n \neq 3 \} \)
(b) \( \{ a^m b^n \mid m \geq 2, n \geq 3 \} \)
(c) \( \{ a^m b \mid m \geq 0 \} \cup \{ b^n a \mid n \geq 0 \} \)

(10) If \( D_1 = (Q, \Sigma, \delta, q_0, F) \) is a minimal DFA, and \( D_2 = (Q, \Sigma, \delta, q_0, Q \setminus F) \) is a DFA for \( \Sigma^* \setminus \mathcal{L}(D_1) \), then is \( D_2 \) necessarily a minimal DFA? Prove your answer.

(11) Find a DFA which accepts \( \{ w \in \{0, 1\}^* \mid w \text{ is a multiple of 3 when interpreted in binary} \} \).
Compute the remainder, mod 3, of the following binary number: 1011011000110101011011. Convert your DFA to a regular expression accepting the same language.

(12) Let \( D \) be a DFA with \( N \) states. Prove the following:
(a) If \( D \) accepts at least one word, then \( D \) accepts a word of length less than \( N \).
(b) If \( D \) accepts at least one word of length \( \geq N \), then \( D \) accepts infinitely many words.

(13) Show that the language \( L := \{ w01^n \mid w \in \{0, 1\}^*, n \in \mathbb{N} \} \cup \{1\}^* \) satisfies the pumping lemma for regular languages. Is \( L \) a regular language?

(14) Give an algorithm that, on input of a DFA \( D \), decides if \( \mathcal{L}(D) = \emptyset \) or not.

(15*) Give an algorithm that, on input of DFA’s \( D_1, D_2 \), decides if \( \mathcal{L}(D_1) \subseteq \mathcal{L}(D_2) \) or not.
(You may appeal to results from the lectures.)

(16*) For any \( X \subseteq \{1\}^* \), show that \( X^* \) is a regular language.