Problem 0. Document how much time you spend on each of the following problems and cite any resources you received help from.

Problem 1. A common variant of an NFA, call it an a TFA, allows multiple starting states and does not support epsilon transitions. Other than these two minor variations, a TFA is identical to an NFA.

Formally, a TFA is a tuple $T = (Q, \Sigma, \delta, I, F)$ where Q is a finite set of states, Σ is a finite alphabet, $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ is the transition function, $I \subseteq Q$ is the set of start states, and $F \subseteq Q$ is the set of accepting states.

Prove that a language is regular if and only if there exists a TFA that recognizes it.

Problem 2. When parsing text files, it can be useful to ignore whitespace characters when searching for a string. If we have a string such as `"professor"`, we might want to match strings like `" p r o f es s or"`. We say that strings of this form *padded* with the whitespace character. However, in this problem we will use the $\sqcup$ symbol to represent whitespace (since a whitespace character is hard to draw).

More formally, if A is a language over the alphabet Σ, then the *padding* of A with $\sqcup$ is the language

$$\begin{aligned}\{w \in \Sigma^* \mid w = \sqcup^{k_1}\, a_1 \sqcup^{k_2} a_2 \cdots \sqcup^{k_n} a_n \sqcup^{k_{n+1}} \\ \text{where } a_1 a_2 \cdots a_n \in A \\ \text{and } k_i \in \mathbb{Z}_{\geq 0} \text{ for all } 1 \leq i \leq n+1\}.\end{aligned}$$

Note that if $A = \{\texttt{abba}\}$, then the padding of A includes strings with an arbitrary number of whitespace characters inserted in between all the characters of `abba`, and therefore includes strings like $\sqcup\sqcup\texttt{a}\sqcup\texttt{bb}\sqcup\texttt{a}\sqcup$

Prove that the class of regular languages is closed under padding.

Problem 3. Let Σ be an alphabet. Given a string $w \in \Sigma^*$, we write w^R to denote the *reverse* of w, i.e., if

$$w = a_1 a_2 a_3 \cdots a_n$$

where $a_i \in \Sigma$ for each $1 \leq i \leq n$, then

$$w^R = a_n a_{n-1} \cdots a_1.$$

If $A \subseteq \Sigma^*$ is a language, we also define the reverse of A to be

$$A^R = \{w^R \mid w \in A\}.$$

Prove that the class of regular languages is *closed* under the reverse operator.

Problem 4. Let $B = \{0, 1\}^*$ be the language of all bit strings. Consider the following two languages:

1. $A_1 = B \circ B^R$
2. $A_2 = \{xx^R \mid x \in B\}$

One of the above languages is regular and the other is not. Describe in your own words the difference between these two languages and give some intuition about why the one is regular but the other one is not.

Bonus Problem (Extra Credit)**.** Let $\Sigma = \{\mathtt{0}, \mathtt{1}, \mathtt{2}, \mathtt{3}, \mathtt{4}, \mathtt{5}, \mathtt{6}, \mathtt{7}, \mathtt{8}, \mathtt{9}\} \cup \{+, \times, -, \div\}$ be an alphabet. (Note that this alphabet should be familiar from the previous assignment.) Let $\widehat{\Sigma} = \Sigma \cup \{\mathtt{(}, \mathtt{)}\}$ (i.e. $\widehat{\Sigma}$ contains all the elements of Σ along with the left- and right-parentheses symbols). A string $w \in \widehat{\Sigma}^*$ is called a *numeric expression* if it is in one of the following **three** forms.

1. w consists of one or more digits with no leading zeros.
2. $w = w_1 \otimes w_2$ where w_1 and w_2 are numeric expressions and $\otimes \in \{+, \times, -, \div\}$.
3. $w = \mathtt{(}x\mathtt{)}$ where x is also a numeric expression.

(Note that all simple numeric expressions from the previous assignment are trivially numeric expressions because of the first two forms). However, we can now have expressions with parentheses such as $\mathtt{(2} - \mathtt{700)} \times 8 \div \mathtt{10}$ and $\mathtt{((1} + \mathtt{2)} \times \mathtt{(3} + \mathtt{4))}$. Let $N \subseteq \widehat{\Sigma}^*$ be the set of all such numeric expressions.

Prove that N is not regular using the pumping lemma.