- The homework is due on April 30, 23:59pm. Please submit your solutions to Gradescope.
- Starting from Homework 1, all homework sets allow *group submissions* up to 2 people. Please write down the names of the members *very clearly* on the first page of your solutions.
- Answer the questions in a way that is clear, correct, convincing, and concise. The level of details to aim for is that your peers in this class should be convinced by your solutions.
- You can use any statements proved during the working sessions/lectures without proofs in your solutions.
- You might notice the difficulty of the homework problems are much higher than the worksheets. *This is by design.* These problems are meant to stretch your ability and solidify your understanding of the core concepts.
- You are expected to spend a reasonable amount of time (measured in hours) working on these problems. Remember you are allowed to utilize any resources. Make sure to cite all the people/webpages/source of infomation that helped.
- Some problems are marked with a *star*; these are more challenging (and fun) extra credit problems. They are optional and do not count toward raw grades.

---

1. ***Parsing regular expressions.*** Can the tack of parsing regular expressions itself be implemented using a regular expression? This property is known as ***universality***—the language is powerful enough to recognize its own structure.

   We can treat the collection of regular expressions itself as a language *RegExp*, containing words that are valid regular expressions over the binary alphabet $\{\mathtt{0},\mathtt{1}\}$. To write down a regular expression in *RegExp*, in addition to `0` and `1` we need a few extra symbols; so we define the alphabet set for *RegExp* to be $\Sigma' := \{\mathtt{0},\mathtt{1},\mathtt{+},{}^{*},\mathtt{(},\mathtt{)}\}$.

   Prove that the language *RegExp* itself is not regular. In other words, we cannot construct a parser for valid regular expressions using only DFAs/NFAs.

2. ***Two-way DFAs.*** We can imagine the input string of a regular DFA be written on a read-only tape, with a cursor initially on the leftmost character of the input. At each step the DFA reads a single character and move its cursor to the right. This is a rather restricted mode of input access.

   What if we allow the cursor to move freely on the tape? Define a ***two-way DFA M*** to be a regular DFA with states Q and alphabet Σ, along with a *cursor* pointing to some character of the input string. The transition function δ takes the current state q and the character at the cursor, then produce the next state q' and a *cursor movement*, either one step to the right or the left on the input tape. More formally, the transition function has the type

   $$\delta : \quad Q \times \Sigma \quad \to \quad Q \times \{\to, \leftarrow\}.$$

   We emphasize that the input tape is *read-only*; unlike a Turing machine, a two-way DFA has no ability to change the symbols on the tape.

   Prove that the class of languages recognized by two-ways are exactly the regular languages.

⋆3. ***Regular or not?*** *Prove or disprove* that each of the languages below is regular (or not). Let Σ^+ denote the set of all *nonempty* strings over alphabet Σ; in other words, $\Sigma^+ = \Sigma \cdot \Sigma^*$. Denote $n(w)$ the integer corresponding to the binary string w.

(a) $\left\{ wxw^R : w, x \in \Sigma^+ \right\}$

(b) $\left\{ ww^R x : w, x \in \Sigma^+ \right\}$

[Hint: To prove that a language L is regular, construct an NFA that recognizes L; to disprove that L is regular, construct a fooling set for L and argue that the construction is correct.]