- The homework is due on April 23, 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. ***Busy chef.*** Construct NFAs that recognize the following languages.

   (a) Let *Sandwich* be an automatic language.

   $$\mathit{Cut}(\mathit{Sandwich}) = \left\{ \mathit{sandwich} \,:\, \mathit{sandwich} \cdot \mathit{sandwich}^R \in \mathit{Sandwich} \right\},$$

   where $\mathit{sandwich}^R$ denote the reversal of the string *sandwich*.
   (*Also try this: no submission required.*) Let *Fish* be an automatic language.

   $$\mathit{Chop}(\mathit{Fish}) = \left\{ \mathit{body} \,:\, \begin{array}{l} \mathit{head} \cdot \mathit{body} \cdot \mathit{tail} \in \mathit{Fish} \text{ for some } \mathit{head} \text{ and } \mathit{tail}, \text{ and} \\ \text{all three } \mathit{head}, \mathit{body}, \text{ and } \mathit{tail} \text{ have the same length} \end{array} \right\}.$$

   *(b) Let *SushiRoll* be an automatic language.

   $$\mathit{Cut}(\mathit{SushiRoll}) = \left\{ \mathit{sushi} \,:\, \mathit{sushi}^n \in \mathit{SushiRoll} \text{ for some } n \geq 0 \right\},$$

   where sushi^n denote the concatenation of the string *sushi* with itself n times.

   $\bigstar$(c) Let *FudgeSquare* be an automatic language.

   $$\mathit{Cut}(\mathit{FudgeSquare}) = \left\{ \mathit{fudge} \,:\, \mathit{fudge}^{|\mathit{fudge}|} \in \mathit{FudgeSquare} \right\}.$$

2. ***When does an NFA accept everything?*** Let N be an arbitrary NFA with n states for some alphabet Σ of size at least 2. How long can the shortest word rejected by NFA N be? Put it differently, let $L(N)$ be the language recognized by the NFA N. What is the smallest $f(n)$ such that

$$\Sigma^{f(n)} \subseteq L(N) \quad \text{implies} \quad L(N) = \Sigma^*?$$

   (a) Prove that $f(n) \leq 2^n$.

   (b) Consider the alphabet set $\Sigma := \left\{\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right], \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right], \#\right\}$. Construct an NFA recognizing the language $\Sigma^* \setminus \{s_n\}$, where s_n is the following word:

$$s_n := \#\underbrace{\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\cdots\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]}_{n\text{ terms}}\#\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\cdots\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\#\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\cdots\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\cdots\#\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]\cdots\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\#.$$

   For example,

$$s_2 := \#\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\#\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\#\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\#.$$

   [Hint: You need to build a few separate gadgets to check that every chunk between two consective # have length exactly n, the two counts within a chuck are differ by one, the counts in every two adjacent chunks are consistent, etc. How many states did you use? If you use at most $C \cdot n + o(n)$ states, then you just proved $f(n) \geq \Omega(2^{n/C})$.]

★ (c) Prove or disprove that $f(n) \geq 2^{n-o(n)}$.