

- 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 information 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.

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

where sandwich^R denote the reversal of the string *sandwich*.

(Also try this: no submission required.) Let *Fish* be an automatic language.

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

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

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

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

★ (c) Let *FudgeSquare* be an automatic language.

$$\text{Cut}(\text{FudgeSquare}) = \{ \text{fudge} : \text{fudge}^{|\text{fudge}|} \in \text{FudgeSquare} \}.$$

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 , and define $\Sigma^{\leq f(n)} := \Sigma^0 \cup \dots \cup \Sigma^{f(n)}$. What is the smallest $f(n)$ such that

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

- (a) Prove that $f(n) \leq 2^n$.
- (b) Consider the alphabet set $\Sigma := \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \# \right\}$. Construct an NFA recognizing the language $\Sigma^* \setminus \{s_n\}$, where s_n is the following word:

$$s_n := \# \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \dots \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{n \text{ terms}} \# \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \dots \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \# \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \dots \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \dots \# \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \dots \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \#.$$

For example,

$$s_2 := \# \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \# \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \# \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \#.$$

[Hint: You need to build a few separate gadgets to check that every chunk between two consecutive # have length exactly n , the two counts within a chunk are differed 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 $f(n) \geq 2^{n-o(n)}$.