1. ***Busy chef.*** Construct NFAs that recognize the following languages.

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

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

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

   ⋆(b) Let *Fish* be an automatic language.

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

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

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

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

2. ***Prefix codes.*** ***Huffman code*** is an efficient lossless encoding method that achieves optimal (symbol-to-symbol) compression rate when the input probability distribution is known. The *prefix-free* property, that no codeword inside the constructed encoding is the prefix of another codeword, makes Huffman code uniquely decodable and very efficient in practice.

   Let L be an automatic language. Construct NFAs that recognize the following languages.

   (a) $\left\{ w \in \Sigma^* : \text{no proper prefix of } w \text{ is in } L \right\}$

   (b) $\left\{ w \in \Sigma^* : \text{no proper suffix of } w \text{ is in } L \right\}$

   ⋆(c) $\left\{ w \in \Sigma^* : \text{no proper } \textit{substring} \text{ of } w \text{ is in } L \right\}$