- You know the drill now: Find students around you to form a ***small group***; use ***all resources*** to help to solve the problems; ***discuss*** your idea with other group member and ***write down*** your own solutions; raise your hand and pull the ***course staffs*** to help; ***submit*** your writeup through Gradescope in *24 hours*.

Our topic for this working session is *language transformation*.

Automatic languages are closed under various transformations. We have seen several examples in class and earlier work sessions: Let L and L' be an automatic language. Then every language in the following list is an automatic language as well.

- $\mathit{union}(L, L') := \{w \in \Sigma^* : w \in L \cup L'\}$
- $\mathit{concatenation}(L, L') := \{w \in \Sigma^* : \text{there are } x \in L \text{ and } y \cup L' \text{ such that } w = xy\}$
- $\mathit{star}(L) := \{w \in \Sigma^* : w = x_1 \cdots x_k, \text{ where every } x_i \in L\}$
- $\mathit{complement}(L) := \{w \in \Sigma^* : w \notin L\}$
- $\mathit{intersection}(L, L') := \{w \in \Sigma^* : w \in L \cap L'\}$
- $\mathit{reverse}(L) := \{\mathit{rev}(w) \in \Sigma^* : w \in L\}$

Today we will work on our techniques in modifying existing DFAs/NFAs for language L so that the resulting machine recognizes certain *transformations* of L. The key is to imagine the process of machine M accepting a word w as a walk from starting state s to accepting state t (which without loss of generality we can assume there is only one, from worksheet last week), such that the labels on the edges in the walk matches the word w.

Now under a given language transformation, the word w is modified to some other words w'. If we can modify the edges and labels of M into new machine M', such that there is a walk from s' to t' in M' matching w' if and only if M accepts w, then we are done. *We emphasize that it has to be "if and only if"!* That is, if there were no walks from s to t in M matching w, in the new machine M' we shouldn't have a walk from s' to t' matching w'. Make sure to double-check.

Example. Let L be an automatic language. Prove that the following language is also automatic:

- $\mathit{supersequence}(L) := \{y \in \Sigma^* : \text{some } x \in L \text{ is a subsequence of } y\}$

Solution: Let $M = (Q, s, A, \Sigma, \delta)$ be a DFA whose language is L. We modify the DFA M into an NFA N as follows:

- Add a self-loop to every state q in Q, with label ε.

We argue that N accepts y if and only if M accepts some subseuqence $x \in L$. Without loss of generality assume both M and N have a single starting state s and accepting state t.

- If there is a walk W' from s to t in N that matches y, consider the subwalk $\hat{W}$ of W' that does not use any self-loops and goes from s to t. All the edges used by $\hat{W}$ appear in M, and thus the label on $\hat{W}$ (denoted as x) must be a word in L and a subsequence of y.
- Now for the reverse, If there is a walk W from s to t in M that matches some $x \in L$ that is a subsequence of y. We can create a walk $\hat{W}'$ in N that matches y, by making use of self-loops to match any character in y that does not appeaer in x. Thus we can go from s to t via $\hat{W}'$ that matches y, so N accepts y.

Let L be an arbitrary automatic language. Prove that the following languages are also automatic.

1. $\textit{prefix}(L) := \left\{x \in \Sigma^* : xy \in L \text{ for some } y \in \Sigma^*\right\}$
2. $\textit{cycle}(L) := \left\{yx : xy \in L \text{ for some } x, y \in \Sigma^*\right\}$
   [Hint: How do we remember the state we started from right before reading y? Rememeber, adding constant memory to DFA/NFA does not change its power.]
3. $\textit{firsthalf}(L) := \left\{x \in \Sigma^* : xy \in L \text{ for some } y \in \Sigma^* \text{ where } |x| = |y|\right\}$

To think about later: (No submissions needed)

4. Let N be a given NFA $(Q, S, A, \Sigma, \delta)$. The language accepted by N is defined as

$$L(N) := \left\{w \in \Sigma^* : \delta^*(S, w) \cap A \neq \varnothing\right\}.$$

   Prove that the following language associated to a given NFA N is also automatic.

$$L^{\subseteq}(N) := \left\{w \in \Sigma^* : \delta^*(S, w) \subseteq A\right\}$$

   (This construction is particularly useful when you try to design NFAs but really hope to set the accepting condition to be "all fingers on double-circles".)

5. Let L and L' be two arbitrary automatic languages. Prove that the following language is also automatic. $L \sqcap L' := \left\{x \sqcap y : x \in L \text{ and } y \in L' \text{ and } |x| = |y|\right\}$, where $x \sqcap y$ denotes the bitwise AND. (For example, `0011` $\sqcap$ `0101` $=$ `0001`.)

Conceptual question: Recall the complement of an automatic language still automatic. Can you repeat this argument for languages recognized by NFAs? Are lanuages recognized by NFAs closed under complement?