Release date: February 16, 2021 Deadline: February 22, 2020: 23:00

**Read the instructions carefully.**

This homework is primarily about proving closure properties of regular languages. To prove that regular languages are closed under some binary operation `op`, a straightforward way is to show how to construct, for any two regular languages $L_1$ and $L_2$, an NFA $N$ recognizing $\texttt{op}(L_1, L_2)$ from DFAs $D_1, D_2$ recognizing $L_1, L_2$ respectively. Remember to prove that $N$ accepts a string if and only if the string belongs to $\texttt{op}(L_1, L_2)$.

You have to decide between whether to define $N$ mathematically (eg. like we did in the proof of closure under intersection), or to describe the construction informally in a human language (eg. for closure under concatenation, "Connect every accepting state of $D_1$ to the initial state of $D_2$ by an $\varepsilon$-transition"). There is a tradeoff here. If the mathematical definition is short, clean, and intuitive, write that and avoid giving a vague informal description. If the mathematical definition is unnecessarily complicated and it hides the main idea, avoid it and describe your construction informally but clearly.

Let $\Sigma$ and $\Gamma$ be two finite alphabets. A function $f : \Sigma^* \longrightarrow \Gamma^*$ is called a *homomorphism* if for all $x, y \in \Sigma^*$, $f(x \cdot y) = f(x) \cdot f(y)$. Observe that if $f$ is a string homomorphism, then $f(\varepsilon) = \varepsilon$, and the values of $f(a)$ for all $a \in \Sigma$ completely determine $f$.

1. Let $x, y, z \in \Sigma^*$. We say that $z$ is a *shuffle* of $x$ and $y$ if the characters in $x$ and $y$ can be interleaved, while maintaining their relative order within $x$ and $y$, to get $z$. Formally, if $|x| = m$ and $|y| = n$, then $|z|$ must be $m + n$, and it should be possible to partition the set $\{1, 2, \ldots, m + n\}$ into two increasing sequences, $i_1 < i_2 < \cdots < i_m$ and $j_1 < j_2 < \cdots < j_n$, such that $z[i_k] = x[k]$ and $z[j_k] = y[k]$ for all $k$. Given two languages $L_1, L_2 \subseteq \Sigma^*$, define

$$\text{shuffle}(L_1, L_2) = \{z \in \Sigma^* \mid z \text{ is a shuffle of some } x \in L_1 \text{ and some } y \in L_2\}.$$

   Prove that the class of regular languages is closed under the shuffle operation.

2. Let $L_1$ be a regular language and $L_2$ be any language (not necessarily regular) over the same alphabet $\Sigma$. Prove that the language $L = \{x \in \Sigma^* \mid x \cdot y \in L_1 \text{ for some } y \in L_2\}$ is regular. Do this by mathematically defining a DFA for $L$ starting from a DFA for $L_1$ and the language $L_2$.

3. Prove that the class of regular languages is closed under inverse homomorphisms. That is, prove that if $L \subseteq \Gamma^*$ is a regular language and $f : \Sigma^* \longrightarrow \Gamma^*$ is a homomorphism, then $f^{-1}(L) = \{x \in \Sigma^* \mid f(x) \in L\}$ is regular. Do this by mathematically defining a DFA for $f^{-1}(L)$ starting from a DFA for $L$ and the function $f$.

4. Prove that the class of regular languages is closed under homomorphisms. That is, prove that if $L \subseteq \Sigma^*$ is a regular language, then so is $f(L) = \{f(x) \mid x \in L\}$. Here, it is advisable to informally describe how you will turn a DFA for $L$ into an NFA for $f(L)$.

5. Prove that if $L \subseteq \Sigma^*$ is a regular language then the language $L' = \{x \in \Sigma^* \mid x \cdot \text{rev}(x) \in L\}$ is also regular, where $\text{rev}(x)$ is the reverse of string $x$. Here, instead of constructing an NFA for $L'$ directly, it could be more convenient to use the already proven closure properties. For example, it might be better to write $L'$ as a union of a finite collection of languages, and then construct an NFA for each language in that collection.

6. Design an algorithm that takes as input the descriptions of two DFAs, $D_1$ and $D_2$, and determines whether they recognize the same language.