Exercise 2.6 & 2.7

For L_1, L_2 regular languages over an alphabet Σ, the *left quotient* $L_2 \diagdown L_1$ of L_1 by L_2 (note that this is different from the set difference $L_2 \setminus L_1$) is defined by

$$L_2 \diagdown L_1 := \{v \in \Sigma^* \mid \exists u \in L_2 : \ uv \in L_1\}$$

1. Use the fact that regular languages are closed under homomorphisms, inverse homomorphisms, concatenation and intersection to prove they are closed under quotienting.
2. Given finite automata $\mathcal{A}_1, \mathcal{A}_2$, construct an automaton $\mathcal{A}$ such that

$$\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_2) \diagdown \mathcal{L}(\mathcal{A}_1)$$

3. Is there any difference when taking the *right quotient* $L_1 \diagup L_2 := \{u \in \Sigma^* \mid \exists v \in L_2 : \ uv \in L_1\}$?
4. Determine the inclusion relation between the following languages:
   - L_1
   - $(L_1 \diagup L_2).L_2$
   - $(L_1.L_2) \diagup L_2$

Solution:

1. Let L_1 and L_2 be regular languages over Σ. Let us denote a barred copy of the alphabet Σ by $\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma\}$ (assuming that Σ and $\overline{\Sigma}$ are disjoint). We define a homomorphism $h : \Sigma \cup \overline{\Sigma} \to \Sigma$ as follows:

$$\begin{aligned} h(a) &= a \quad \text{for every } a \in \Sigma \\ h(\overline{a}) &= a \quad \text{for every } a \in \Sigma \end{aligned}$$

   Thus $h^{-1}(L_1)$ consists of words from L_1 with all possible combinations of letters being barred or not. (E.g. $h^{-1}(\{ab\}) = \{ab, a\overline{b}, \overline{a}b, \overline{a}\overline{b}\}$.)

   We now intersect $h^{-1}(L_1)$ with a regular language $L_2.\overline{\Sigma}^*$ in order to get all words from L_1 with prefix from L_2 but with the remaining suffix being barred.

   We can now apply homomorphism $\overline{h}$ defined by

$$\begin{aligned} \overline{h}(a) &= \varepsilon \quad \text{for every } a \in \Sigma \\ \overline{h}(\overline{a}) &= a \quad \text{for every } a \in \Sigma \end{aligned}$$

   in order to obtain the suffixes only, now being unbarred. Hence,

$$L_2 \diagdown L_1 = \overline{h}(h^{-1}(L_1) \cap L_2.\overline{\Sigma}^*)$$

   proves the regularity of the quotient.

2. In order to accept a word $v \in L_2 \diagdown L_1$, we need to guess a word $u \in L_2$ and check whether $uv \in L_1$. Therefore, we can build a parallel composition of automata accepting L_1 and L_2 using the product construction and replace all transitions by ε-transitions (we are guessing the prefix that actually is not there) and adding ε-transitions from all states corresponding to final states for L_2 to the respective state of the automaton for L_1.

   Formally, let $\mathcal{A}_i = (Q_i, \Sigma, \delta_i, q_i, F_i)$ be such that $\mathcal{L}(\mathcal{A}_i) = L_i$ for $i \in \{1, 2\}$. We construct

   $$\mathcal{A} = ((Q_1 \times Q_2) \cup Q_1, \Sigma, \delta, (q_1, q_2), F_1)$$

   so that $\mathcal{L}(\mathcal{A}) = L_2 \diagdown L_1$. We set the transition relation δ as follows:

   $$\begin{array}{lll} (p,r) \xrightarrow{\varepsilon} (p',r') & \text{for every } a \in \Sigma \text{ with } p \xrightarrow{a}_1 p' \text{ and } q \xrightarrow{a}_2 q' & \text{(guessing the prefix)} \\ (p,r) \xrightarrow{\varepsilon} p & \text{for every } r \in F_2 & \text{(prefix is in } L_2\text{)} \\ p \xrightarrow{a} p' & \text{for every } p \xrightarrow{a}_1 p' & \text{(checking the suffix)} \end{array}$$

   where $q \xrightarrow{a}_i q'$ denotes $\delta_i(q, a) \ni q'$.

3. Similarly as in (a), we have

   $$L_1 \diagup L_2 = \overline{h}(h^{-1}(L_1) \cap \overline{\Sigma}^*.L_2)$$

   The direct construction of an automaton recognizing the right quotient is not as straightforward as in the case with left quotient: we need to check the intersection of L_2 with the language recognized by the automaton $\mathcal{A}_1$ with any initial state. An easier approach is to make use of the *reverse* construction together with the construction above, since

   $$L_1 \diagup L_2 = (L_2^R \diagdown L_1^R)^R$$

4. None of the inclusions holds in general. Let

   $$\begin{array}{lll} L_1 & = & \{a, b\} \\ L_2 & = & \{b, bb\} \end{array}$$

   Then quotienting removes all words from L_1 not having a suffix in L_2 and appending L_2 may add new suffixes as follows:

   $$\begin{array}{lll} L_1 \diagup L_2 & = & \{\varepsilon\} \\ (L_1 \diagup L_2).L_2 & = & \{b, bb\} \\ L_1.L_2 & = & \{ab, abb, bb, bbb\} \\ (L_1.L_2) \diagup L_2 & = & \{a, ab, \varepsilon, b, bb\} \end{array}$$

   which disproves all inclusions except for $(L_1 \diagup L_2).L_2 \subseteq (L_1.L_2)/L_2$ and $L_1 \subseteq (L_1.L_2) \diagup L_2$. To disprove the former, let $L_1 = \{a, b\}, L_2 = \{b, ab\}$, then $(L_1 \diagup L_2).L_2 = \{b, ab\} \not\subseteq \{\varepsilon, a, b, aa, ba\} = (L_1.L_2)/L_2$. To disprove the latter, let $L_1 = \{a\}, L_2 = \emptyset$, then $(L_1.L_2) \diagup L_2 = \emptyset \diagup \emptyset = \emptyset \not\supseteq \{a\}$.

   We can at least prove the last inclusion holds for $L_1 = \emptyset$ or $L_2 \neq \emptyset$. The former case is trivial, for the latter let $v \in L_2$. If $u \in L_1$ then $uv \in L_1 L_2$ and thus $u \in (L_1.L_2) \diagup L_2$.