Problem 5. [Category: Design] For languages A and B over alphabet Σ, let the *shuffle* of A and B be the language

$$\{w \mid w = a_1 b_1 a_2 b_2 \cdots a_n b_n \text{ where } a_i, b_i \in \Sigma^* \text{ and } a_1 a_2 \cdots a_n \in A,\ b_1 b_2 \cdots b_n \in B\}$$

Show that the class of regular languages is closed under shuffle. While you need not prove the correctness of your construction, it should be clearly explained with a formal definition. **[10 Points]**

Note: In homework 4, you proved that regular languages are closed under *perfect shuffle*. In perfect shuffle, symbols from a string in A strictly alternate with symbols from a string in B. In shuffle, however, *sub-strings* (possibly empty) of a string from A alternate with *sub-strings* (possibly empty) of a string from B. For example, if $A = \{11\}$ and $B = \{00\}$ then the perfect shuffle of A and B is $\{1010\}$ while the shuffle of A and B is $\{1100, 1010, 1001, 0011, 0110, 0101\}$.

Solution: There are two possible ways to solve this problem — using closure properties, or by an explicit construction of an automaton for shuffle(A, B).

Closure Properties: Let $\widehat{\Sigma} = \{a, \bar{a} \mid a \in \Sigma\}$. So $\widehat{\Sigma}$ contains for each symbol a of Σ, the symbols itself and a "copy" (which is $\bar{a}$). For example $\widehat{\{0,1\}} = \{0, \bar{0}, 1, \bar{1}\}$. Consider homomorphism, $h_A : \widehat{\Sigma}^* \to \Sigma^*$ such that $h(a) = a$ and $h(\bar{a}) = \epsilon$ for $a \in \Sigma$. Now let us consider $L_A = h_A^{-1}(A)$. The strings in L_A are such that if we look at the "unbarred" symbols then they form a string in A. Similarly, define $h_B : \widehat{\Sigma}^* \to \Sigma^*$ such that $h(a) = \epsilon$ and $h(\bar{a}) = a$ for $a \in \Sigma$. Then, take $L_B = h_B^{-1}(B)$; so L_B contains strings such that if you look at the "barred" symbols and remove the bars then you get a string in B.

Let $L = L_A \cap L_B = \{w \mid h_A(w) \in A \text{ and } h_B(w) \in B\}$; that is, the unbarred symbols in w form a string in A, and the barred symbols form a string in B (after removing the bar). L is almost the shuffle of A and B — the only difference it has the barred symbols, which we will remove next. Consider $h : \widehat{\Sigma}^* \to \Sigma^*$ such that $h(a) = h(\bar{a}) = a$ for $a \in \Sigma$. Now it is easy to see that $h(L) = \text{shuffle}(A, B)$. Thus, shuffle of A and B is regular because we obtained it from A and B by applying regularity preserving operations.

By construction: Let $M_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ be a DFA recognizing A and $M_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$ be a DFA recognizing B. The NFA for shuffle of A and B will simulate both M_A and M_B on the input, while nondeterministically choosing which machine to run on a particular input symbol; note, unlike in the perfect shuffle case, we don't need to strictly alternate running the machines and so the construction is actually simpler. So the NFA will be obtained by a "modified" cross-product construction.

Formally, let $N = (Q, \Sigma, \delta, q_0, F)$ where

- $Q = Q_A \times Q_B$
- $q_0 = (q_A, q_B)$
- $F = F_A \times F_B$
- For $a \in \Sigma$, δ is given as

  $$\delta((p_A, p_B), a) = \{(\delta_A(p_A, a), p_B), (p_A, \delta_B(p_B, a))\}$$

  In all other cases, δ is $\emptyset$.

The correctness can be established by showing that if N on an input w reaches a state (p_A, p_B) then there is a way to break up w so that running M_A on some of the substrings reaches p_A and running M_B on the remaining substrings reaches p_B. Formally,

$$\begin{aligned}\hat{\Delta}(q_0, w) = \{(p_A, p_B) \mid \quad & \exists a_1, b_1, a_2, b_2, \ldots a_k, b_k.\ a_i, b_i \in \Sigma^*, \\ & \hat{\delta}_A(q_A, a_1 a_2 \cdots a_k) = p_A \text{ and } \hat{\delta}_B(q_B, b_1 b_2 \cdots b_k) = p_B\}\end{aligned}$$

The above observation can be proved by induction on the length of w and can be used to prove the correctness of the construction.

■